Lightweight Methods and Models for Practical Visual Speech Recognition from Video Sequences

A Dissertation

submitted to the designated
by the Assembly
of the Department of Computer Science and Engineering
Examination Committee

by

Iason - Ioannis Panagos

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

University of Ioannina
School of Engineering
Ioannina 2025

Advisory Committee:

- **Christophoros Nikou**, Professor, Department of Computer Science & Engineering, University of Ioannina
- **Giorgos Sfikas**, Assoc. Professor, Department of Surveying and Geoinformatics Engineering, University of West Attica
- **Lisimachos P. Kondi**, Professor, Department of Computer Science & Engineering, University of Ioannina

Examining Committee:

- **Christophoros Nikou**, Professor, Department of Computer Science & Engineering, University of Ioannina
- **Giorgos Sfikas**, Assoc. Professor, Department of Surveying and Geoinformatics Engineering, University of West Attica
- **Lisimachos P. Kondi**, Professor, Department of Computer Science & Engineering, University of Ioannina
- **Aristeidis Lykas**, Professor, Department of Computer Science & Engineering, University of Ioannina
- **Konstantinos Blekas**, Professor, Department of Computer Science & Engineering, University of Ioannina
- **Anastasios Kesidis**, Professor, Department of Surveying and Geoinformatics Engineering, University of West Attica
- **Ioannis Kakogeorgiou**, Researcher C’, Institute of Informatics & Telecommunications, National Centre for Scientific Research “Demokritos”

Dedication

Dedicated to my father.

Acknowledgements

First of all, I would like to thank my supervisor Professor Christophoros Nikou for his cooperation throughout all stages of this journey. His guidance and support were of paramount importance in making this dissertation possible and I could not ask for a better supervisor. I hope our long-lasting collaboration continues beyond the scope of this work in future projects.

I would also like to thank Associate Professor Giorgos Sfikas who co-supervised this dissertation during its entirety. Our fruitful and insightful discussions helped me overcome obstacles and shift my outlooks, and has inspired me to grow as a researcher.

I would also like to thank Professors Lisimachos P. Kondi and Aristeidis Lykas for helping shape my academic knowledge and research interests, as well as the remaining examiners of the committee for their comments on improving this manuscript.

Last but not least, I would like to extend my heartfelt thanks to my family and friends for their continuous encouragement and motivation to persevere in this arduous endeavor.

Table of Contents

List of Figures

List of Tables

Abstract

Iason - Ioannis Panagos, Ph.D., Department of Computer Science and Engineering, School of Engineering, University of Ioannina, Greece, 2025.
Lightweight Methods and Models for Practical Visual Speech Recognition from Video Sequences.
Advisor: Christophoros Nikou, Professor.

Visual Speech Recognition (VSR) is a computer vision problem that aims to decode spoken words of one or more speakers from visual media without the presence of sound. Applications of VSR are found in numerous domains, with profound impacts on various aspects of everyday life. A notable application lies in the field of accessibility in medicine, where a VSR system can assist individuals with speech impairments, significantly enhancing their quality of life. Other applications include, but are not limited to, video captioning, and personal security systems, each with their own value. While recently there has been a steady increase in research interest regarding VSR, the issue of practicality has not been adequately explored. More specifically, the proposed models and methods often fail to consider the computational costs associated with their architectures, which severely limits or outright prevents their applicability in real-world scenarios.

In this dissertation, we focus on addressing this oversight by developing lightweight and efficient end-to-end models for practical Visual Speech Recognition of isolated words. To realize this objective, we explore a multitude of approaches to reduce network size and complexity using a wide variety of methods. Owing to these reduced hardware requirements, such models can be applied to a broader range of applications and cover a sizable amount of practical real-life scenarios, offering a series of benefits. The fundamental design of a VSR system follows a two-step structure that employs expensive components such as deep convolutional neural networks with large hardware overheads that are prohibitively expensive to deploy. Our goal is reducing these resource requirements while maintaining acceptable recognition rates.

To that end, we first employ techniques that exploit efficient formulations and low-cost operations to shrink model sizes without severely compromise performance. We replace the standard, resource-intensive components in existing networks with more efficient ones, achieving significant reductions in model parameter counts as well as in computational complexity. Moreover, we design a lightweight temporal block blueprint that is flexible in its design and can be adapted to the resources at hand and use it to develop highly-efficient networks with minimal hardware demands.

Next, we shift our attention to a more holistic approach, by designing a lightweight VSR model using efficient components. A systematic study is conducted evaluating multiple networks and structures for visual feature extraction as well as sequence modeling. We select the best-performing components and combine them in a unified end-to-end architecture that achieves very high recognition accuracy while being compact, outperforming all other lightweight approaches in the literature. Finally, using this model as a baseline, we explore techniques to improve its performance without raising its complexity, attempting to bridge the gap with larger models. To that end, we incorporate channel attention in its temporal blocks to enhance feature representation, while we refine its training process by introducing regularization that allows the networks to learn more descriptive features from the data. Finally, we combine these additions to achieve significant recognition uplifts without affecting the network overhead.

Εκτεταμενη Περιληψη

Ιάσων - Ιωάννης Πανάγος, Δ.Δ., Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πολυτεχνική Σχολή, Πανεπιστήμιο Ιωαννίνων, 2025.
Αποδοτικές Μέθοδοι και Μοντέλα για Εφαρμόσιμη Οπτική Αναγνώριση Ομιλίας από Ακολουθίες Βίντεο.
Επιβλέπων: Χριστόφορος Νίκου, Καθηγητής.

Η «Οπτική Αναγνώριση Ομιλίας» (ΟΑΟ) είναι ένα υπο-πρόβλημα του κλάδου της υπολογιστικής όρασης που αποσκοπεί στην αποκωδικοποίηση ομιλίας ενός ή περισσότερων ομιλητών σε οπτικά μέσα χωρίς την παρουσία ήχου. Μαζί με την «Ακουστική Αναγνώριση Ομιλίας» (ΑΑΟ) η οποία αποτελεί το ανίστοιχο υπο-πρόβλημα που αφορά μόνο ακουστικό υλικό, όπως για παράδειγμα ηχογραφήσεις, οι δύο αυτές υποκατηγορίες απαρτίζουν το πιο γενικό πρόβλημα της «Αναγνώρισης Ομιλίας» (ΑΟ). Στην ειδική περίπτωση που ένα σήμα απαρτίζεται από μια οπτική αλλά και μια ακουστική συνιστώσα, για την αναγνώριση της ομιλίας μπορεί να εφαρμοστεί μια μέθοδος (ΟΑΟ ή ΑΑΟ), αλλά και συνδυαστικές μέθοδοι που χρησιμοποιούν και τις δύο συνιστώσες. Σε αυτή την περίπτωση, οι μέθοδοι αυτές αφορούν το υπο-πρόβλημα της «Οπτικο-Ακουστικής Αναγνώρισης Ομιλίας» (ΟΑΑΟ) και εκμεταλλεύονται συμπληρωματικά τον ήχο και την εικόνα.

Κάθε υπο-πρόβλημα της ΑΟ έχει πρακτικές εφαρμογές σε πλήθος πραγματικών προβλημάτων που συναντώνται σε ένα ευρύ φάσμα τομέων της καθημερινότητας. Για παράδειγμα, συστήματα για ΑΑΟ εφαρμόζονται εκτενώς από πλατφόρμες προβολής βίντεο όπου δημιουργούν υποτίτλους από τον ήχο με αυτόματο τρόπο (χωρίς ανθρώπινη παρέμβαση). Ασθενείς που αντιμετωπίζουν δυσκολίες επικοινωνίας όπως αδυναμία ομιλίας μπορούν να επωφεληθούν σε μεγάλο βαθμό από ένα σύστημα ΟΑΟ που αναγνωρίζει ομιλία από τις κινήσεις των χειλιών χωρίς ήχο. Επιπλέον εφαρμογές που αξίζει να αναφερθούν περιλαμβάνουν τους τομείς της προσωπικής ασφάλειας (π.χ. οπτικοί κωδικοί σε φορητές συσκευές και άλλες πλατφόρμες

αλληλεπίδρασης), της κοινωνικής ασφάλειας (π.χ. έλεγχος σε δημοσίους χώρους), την ψηφιοποίηση παλαιότερου οπτικο-ακουστικού υλικού και διατήρηση της πολιτιστικής κληρονομιάς, καθώς και ποιοτικές βελτιώσεις σε θορυβώδη εργασιακά περιβάλλοντα. Και τα δύο υπο-προβλήματα συναντούν εφαρμογές στην ψυχαγωγία και τις ψηφιακές συνεδρίες για την αυτόματη παραγωγή υποτίτλων ή την καταγραφή των συζητήσεων αλλά και σε έξυπνα σπίτια (π.χ. διάφοροι αυτοματισμοί συσκευών μέσω ηχητικών εντολών). Οι συνδυαστικές μέθοδοι για ΟΑΑΟ μπορούν να λειτουργήσουν σε περιπτώσεις πολυτροπικών δεδομένων αναγνωρίζοντας την ομιλία κάνοντας εναλλαγή μεταξύ της χρήσης βίντεο ή ήχου ανάλογα με τις συνθήκες.

Η συντριπτική πλειοψηφία της έρευνας για ΑΟ εστιάζει στην Ακουστική Αναγνώριση Ομιλίας λόγω πολλαπλών παραγόντων. Αρχικά, τα δεδομένα ήχου είναι ευκολότερα στην επεξεργασία τους από τα βίντεο καθώς τα δεύτερα είναι πιο πολύπλοκα (οπτικές διαστάσεις αλλά και χρονική διάρκεια), δεδομένου ότι ένα βίντεο αποτελεί στην ουσία μια ακολουθία από εικόνες που συσχετίζονται. Παράλληλα, η Οπτική Αναγνώριση Ομιλίας είναι ένα πιο δύσκολο πρόβλημα, διότι η οπτική πληροφορία που σχετίζεται με την ομιλία είναι περιορισμένη σε σύγκρηση με τον ήχο. Για παράδειγμα, πολλές λέξεις με εντελώς διαφορετικό νόημα παράγονται από οπτικά παρόμοιες κινήσεις των χειλιών, γεγονός που είναι ιδιαίτερα εμφανές για μερικά σύμφωνα. Πολλές φορές, ο πληθυντικός μιας λέξης στην Αγγλική γλώσσα προέρχεται από την προσθήκη ενός καταλητικού «s» το οποίο δύσκολα διακρίνεται από μια ακολουθία εικόνων χωρίς ήχο. Ο ανθρώπινος παράγοντας και οι ιδιαιτερότητες του κάθε ομιλητή (π.χ., προφορά) είναι άλλη μια συνιστώσα που μπορεί να επηρεάσει τα αποτελέσματα της αναγνώρισης, το οποίο είναι ένα ζήτημα που αφορά και την ΑΑΟ. Στις παραπάνω περιπτώσεις, η ακουστική αναγνώριση επιτυγχάνει καλύτερα αποτελέσματα συγκριτικά με την οπτική, η οποία απαιτεί πολλούς διαθέσιμους υπολογιστικούς πόρους για την χρήση απαιτητικών αλγορίθμων. Η ποιότητα της εικόνας του βίντεο επίσης επηρεάζει την τελική αναγνώριση καθώς παράγοντες όπως η ανάλυση, το χρώμα, η αντίθεση ή και οπτικές ατέλειες στο βίντεο μπορούν να οδηγήσουν σε εσφαλμένα αποτελέσματα. Τέλος, δεν πρέπει να παραληφθεί το γεγονός πως τα διαθέσιμα δεδομένα είναι καθοριστικής σημασίας για την ανάπτυξη τέτοιων μεθόδων, και γενικά, σύνολα δεδομένων ήχου είναι πιο προσιτά για έρευνα, καθώς υπάρχει μεγαλύτερος διαθέσιμος όγκος, διαμοιράζονται ευκολότερα λόγω μεγέθους αλλά και μπορούν να χαρακτηριστούν με μεγαλύτερη ταχύτητα και ακρίβεια απ‘ότι τα σύνολα δεδομένων με βίντεο (χωρίς ήχο). Παρ‘όλα αυτά, η σημασία και οι εφαρμογές της ΟΑΟ δεν πρέπει να παραμεληθούν διότι αποτελεί την μοναδική επιλογή για ΑΟ σε περιπτώσεις που δεν είναι δυνατή η καταγραφή ήχου (π.χ. σε κλειστά κυκλώματα παρακολούθησης (CCTV)), όπου υπάρχει θόρυβος ή εάν για κάποιον λόγο το ηχητικό μέρος ενός βίντεο δεν καταγράφηκε.

Το πρόβλημα της ΑΟ διερευνάται για μερικές δεκαετίες, ωστόσο πρόσφατα παρατηρείται ένα ανανεομένο ερευνητικό ενδιαφέρον που σχετίζεται άμεσα με την πρόοδο της μηχανικής μάθησης. Ταυτόχρονα, η ευρεία διαθεσιμότητα μεγάλης κλίμακας βάσεων δεδομένων που περιλαμβάνουν ήχο ή/και βίντεο επέτρεψαν την ανάπτυξη και εκπαίδευση αρχιτεκτονικών για ΑΑΟ, ΟΑΟ ή και συνδυασμό τους. Αντιπροσωπευτικές γλώσσες είναι η Αγγλική και η Κινεζική καθώς πρόκεται για δύο μεγάλα πλήθη ομιλητών παγκοσμίως, αν και υπάρχουν και μικρότερα σύνολα δεδομένων με περιορισμένη χρησιμοότητα για άλλες γλώσσες. Οι παλαιότερες μέθοδοι χρησιμοποιούσαν παραδοσιακές τεχνικές υπολογιστικής όρασης και επεξεργασίας εικόνας με αποτέλεσμα οι παραγόμενοι αλγόριθμοι να είναι σχετικά απλοί και αρκετά περιορισμένοι όσον αφορά την αναγνώριση. Αυτό οφειλόταν στην υπολογιστική ικανότητα των μοντέλων αλλά και στην διαθεσιμότητα των δεδομένων, δύο συνιστώσες που πλέον δεν αποτελούν περιοριστικούς παράγοντες για την ανάπτυξη μεθόδων.

Η συνηθισμένη προσέγγιση που ακολουθείται από την βιβλιογραφία για κάθε τύπο ΑΟ χωρίζει το πρόβλημα σε δύο απλούστερα υπο-προβλήματα που είναι ευκολότερα στην επίλυση. Αρχικά εξάγονται διακριτικά χαρακτηριστικά από την είσοδο τα οποία στη συνέχεια τροφοδοτούνται σε ένα ακολουθιακό μοντέλο το οποίο μοντελοποιεί τις αλληλεπιδράσεις και αλληλοεξαρτήσεις τους στο πεδίο του χρόνου για την διάρκεια της ακολουθίας. Η προσέγγιση αυτή έχει συναντήσει μεγάλη επιτυχία στο συγκεκριμένο πρόβλημα και έχει πετύχει πολύ καλά αποτελέσματα σε όλες τις κατηγορίες ΑΟ. Σύμφωνα με την έρευνά μας στην σχετική βιβλιογραφία, όλες οι δημοσιεύσεις πλέον χρησιμοποιούν αρχιτεκτονικές βασισμένες σε βαθειά μάθηση και Συνελικτικά Νευρωνικά Δίκτυα για τις ιδιότητές τους να εξάγουν αναπαραστάσεις από την είσοδο αλλά την ικανότητά τους να εφαρμόζονται σε μεγάλο πλήθος δεδομένων. Για την επεξεργασία της ακολουθίας, μέχρι τα τελευταία 5 χρόνια επαναληπτικές δομές (LSTM, GRU) αποτελούσαν τα βασικά μοντέλα, με μια προτίμηση προς τα GRU. Αυτές οι δομές είχαν αμφιμερή σχεδιασμό, δηλαδή λάμβαναν σαν είσοδο την ακολουθία δύο φορές, κανονικά και χρονικά αντεστραμμένη. Παρ‘όλο που χρησιμοποιούνται ακόμη, έχουν σταδιακά αντικατασταθεί από αρχιτεκτονικές βασισμένες στα Χρονικά Συνελικτικά Δίκτυα, που βασίζονται την λειτουργία της συνέλιξης στο πεδίο του χρόνου και σχεδιάζονται με τρόπο που δεν επιτρέπει μελλοντική πληροφορία να επηρεάζει την τρέχουσα χρονική στιγμή σε μια ακολουθία. Τα Χρονικά Συνελικτικά Δίκτυα επιτυγχάνουν υψηλές επιδόσεις ενώ παρουσιάζουν ευκολότερη (από την άποψη βελτιστοποίησης) και γρηγορότερη διαδικασία εκπαίδευσης σε σύγκρηση με τις επαναληπτικές δομές ενώ μειώνουν την υπολογιστική πολυπλοκότητα χάρη στην απλότητα της συνέλιξης. Οι παράγοντες αυτοί έχουν συνεισφέρει σε μια συνεχώς αυξανόμενη χρήση Χρονικών Συνελικτικών Δικτύων σε δημοσιεύσεις για ΑΟ.

Όπως προαναφέρθηκε, ο σχεδιασμός ενός συστήματος για ΟΑΟ είναι μια δυσκολότερη διαδικασία απ'ότι για ΑΑΟ αφού η μορφή των δεδομένων και οι προκλήσεις του προβλήματος πρέπει να ληθφούν υπ'όψη, και απαιτώνται ισχυρές και πολύκπλοκες αρχιτεκτονικές με επαρκή υπολογιστική ισχύ. Κατά συνέπεια, τα συστήματα για ΑΑΟ είναι απλούστερα όσον αφορά την πολυπλοκότητα ενθαρρύνοντας την ταχεία πρόοδο της έρευνας, ενώ τα οι μέθοδοι για ΟΑΟ παρουσιάζουν υψηλές απαιτήσεις σε υπολογιστικούς πόρους και συχνά είναι απαγορευτικές όσον αφορά το κόστος εκπαίδευσης. Κατόπιν ενδελεχούς ανάλυσης της βιβλιογραφίας για ΟΑΟ λέξεων της Αγγλικής γλώσσας διαπιστώθηκε πως ενώ η ακρίβεια της αναγνώρισης βελτιώνεται, τα προτεινόμενα μοντέλα συνεχώς μεγαλώνουν σε μέγεθος και πολυπλοκότητα, άρα και σε απαιτήσεις υλικού. Ένα μειονέκτημα τέτοιων μοντέλων είναι ότι η εφαρμογή τους περιορίζεται σε συνθήκες εργαστηρίου με επαρκές υπολογιστικούς πόρους που δεν αντικατοπτρίζουν πραγματικά σενάρια και δεν προσφέρουν τα πλεονεκτήματά τους.

Σκοπός αυτής της διατριβής είναι ο σχεδιασμός, η ανάπτυξη και η εκτίμηση αποδοτικών μεθόδων με διαχειρίσμες απαιτήσεις σε υλικό για εφαρμόσιμη Οπτική Αναγνώριση Ομιλίας, δίνοντας έμφαση στην ακριβή αναγνώριση λέξεων σε πραγματικές συνήθκες, χωρίς την χρήση ήχου. Οι μέθοδοι που αναπτύσσονται βασίζονται σε αρχιτεκτονικές βαθειάς μάθησης και σε τεχνικές που αποσκοπούν στην μείωση του μεγέθους των δικτύων και του κόστους εκτέλεσής τους, μειώνοντας έτσι τις απαιτήσεις σε πόρους επιτρέποντας περισσότερες εφαρμογές σε πλήθος σεναρίων όπου οι διαθέσιμοι πόροι είναι περορισμένοι. Οι τεχνικές που εφαρμόζονται περιλαμβάνουν μείωση μεγέθους και πολυπλοκότητας αλλά και σχεδιασμό ενός ολοκληρωμένου μοντέλου μικρού υπολογιστικού κόστους και υψηλής ικανότητας αναγνώρισης από βίντεο. Για την βελτίωση και ανάλυση των μεθόδων πραγματοποιείται πλήθος πειραμάτων και μελετών για τον προσδιορισμό των βέλτιστων παραμέτρων χωρίς να αυξάνεται η πολυπλοκότητα των δικτύων. Οι αρχιτεκτονικές που αναπτύσσονται επιτυγχάνουν μεγάλη ακρίβεια αναγνώρισης που συναγωνίζεται πολύ μεγαλύτερα ως προς το μέγεθος και την υπολογιστική πολυπλοκότητα μοντέλα της βιβλιογραφίας.

Η πρώτη προσέγγιση που εξετάζουμε εκμεταλλεύεται το γινόμενο Kronecker για να πετύχει μείωση του μεγέθους όταν εφαρμόζεται σε επίπεδα ενός νευρωνικού δικτύου. Πιο συγκεκριμένα, τα βάρη ενός επιπέδου υπολογίζονται με χρήση ενός αθροίσματος γινομένων Kronecker μικρότερων πινάκων, μειώνοντας έτσι τις απαιτούμενες παραμέτρους χωρίς να επηρεάζονται οι διαστάσεις. Το πλήθος των όρων του αθροίσματος αποτελεί υπερ-παράμετρος (ορίζεται από τον χρήστη) και ουσιαστικά ελέγχει τη μείωση του μεγέθους του δικτύου. Σε μια ολοκληρωμένη αρχιτεκτονική αντικαθιστούμε κάθε επίπεδο πλήρους σύνδεσης και συνέλιξης με αντίστοιχα επίπεδα που χρησιμοποιούν την παραπάνω τεχνική, επιτυγχάνοντας σημαντικές μειώσεις στο μέγεθος και πραγματοποιούμε πειράματα στο μεγαλύτερο σύνολο δεδομένων για αναγνώριση λέξεων της αγγλικής γλώσσας. Παράλληλα, αναλύουμε την επίδραση σε κάθε συνιστώσα του δικτύου όσον αφορά το μέγεθος και την απόδοση, παρατηρώντας πως για περισσότερους όρους στο άθροισμα, επιτυγχάνεται μεγαλύτερη συμπίεση με ένα αντίστοιχο αντίκτυπο στην ποιότητα αναγνώρισης. Γενικεύοντας, εφαρμόζουμε την ίδια τεχνική σε ένα μοντέλο για ΑΑΟ και επιτυγχάνουμε μείωση του μεγέθους κατά 90% με μόνο 0.5% πτώση στην ακρίβεια αναγνώρισης.

Στη συνέχεια ακολουθούμε μια εναλλακτική προσέγγιση η οποία ελαφρύνει όχι μόνον το μέγεθος αλλά και το υπολογιστικό κόστος των μοντέλων. Αυτό επιτυγχάνεται σε ένα μπλοκ που λειτουργεί σε δύο στάδια, πρώτα εξάγονται κάποια διακριτικά χαρακτηριστικά με μικρότερη διάστηαση από την είσοδο και στη συνέχεια μια συνέλιξη «κατά-βάθος» τα φιλτράρει χωρικά, δημιουργώντας έτσι ένα νέο σετ. Έπειτα, τα δύο ξεχωριστά σετ συνενώνονται ώστε η έξοδος να έχει ίδιες διαστάσεις με την είσοδο όσον αφορά τα κανάλια. Ένας μηχανισμός προσοχής εισάγεται στο μπλοκ για να βελτιώσει την επίδοση καλύπτοντας μια αδυναμία αυτής της μεθόδου. Με αυτά τα μπλοκ αντικαθιστούμε τα στρώματα συνέλιξης σε μια αρχιτεκτονική μειώνοντας έτσι το συνολικό κόστος. Επιπλέον, σχεδιάζουμε μια σειρά από Χρονικά Συνελικτικά Δίκτυα βασισμένα σε ένα πολύ αποδοτικό μπλοκ το οποίο χωρίζει την είσοδο σε δύο τμήματα και εφαρμόζει ξεχωριστές λειτουργίες στο καθένα. Το νέο μπλοκ είναι ευέλικτο όσον αφορά το σχεδιασμό και την παραμετροποίηση, επιτρέποντας την ανάπτυξη πλήθους μοντέλων με χαμηλή πολυπλοκότητα και δυνατότητα για εφαρμογές σε περιπτώσεις με λίγους διαθέσιμους πόρους. Τα αποτελέσματα της μεθόδου στην αναγνώριση ομιλίας υποδεικνύουν μια μείωση στο μέγεθος και των απαιτούμενων υπολογισμών των μοντέλων που αγγίζει το 75% χωρίς να επηρεάζει αρνητικά τις επιδόσεις.

Η επόμενη προσέγγισή μας αποτελεί μια πιο σφαιρική προσέγγιση όπου ένα ενιαίο μοντέλο σχεδιάζεται εξ'αρχής με σκοπό την απόδοση και την επίδοση. Για την εξαγωγή αναπαραστάσεων από την είσοδο, δοκιμάζονται διάφορα πρακτικά και συμπαγή δίκτυα με μικρές απαιτήσεις σε πόρους από την βιβλιογραφία της υπολογιστικής όρασης, καθώς αυτά τα μοντέλα εμφανίζουν καλές δυνατότητες γενίκευσης όταν χρησιμοποιούνται για αυτό τον σκοπό. Τα μοντέλα ακολουθούν πλήθος σχεδιαστικών φιλοσοφιών: μηχανισμούς προσοχής, παράλληλες συνελίξεις, έυρεση αρχιτεκτονικής κ.α. Για την μοντελοποίηση της ακολουθίας, ως βάση χρησιμοποιούμε ένα απλό Χρονικό Συνελικτικό Δίκτυο και αναπτύσσουμε διάφορα υπολογιστικά μπλοκ με μικρό κόστος βασισμένοι σε ποικιλομορφία δομών της βιβλιογραφίας. Τα μοντέλα με την καλύτερη ακρίβεια αναγνώρισης για κάθε περίπτωση συνδυάζονται σε μια ενιαία αρχιτεκτονική που συγκρίνεται με την βιβλιογραφία και επιτυγχάνει παρόμοιες επιδόσεις στο 1/3 των παραμέτρων και 1/5 του υπολογιστικού κόστους, ενώ ταυτόχρονα ξεπερνά όλες τις αντίστοιχες (με μικρές απαιτήσεις σε πόρους) μεθόδους της βιβλιογραφίας. Επιπρόσθετα, αναλύουμε τις υπερ-παραμέτρους για κάθε συνιστώσα του μοντέλου για να αναδείξουμε την βέλτιστη αρχιτεκτονική.

Τέλος, χρησιμοποιώντας το παραπάνω ενιαίο μοντέλο, εξετάζουμε μεθόδους βελτίωσης της ακρίβειας αναγνώρισής του χωρίς να αυξήσουμε την πολυπλοκότητα. Πρώτα, εισάγουμε μηχανισμούς προσοχής σε κάθε μπλοκ του ακολουθιακού Χρονικού Συνελικτικού Δικτύου οι οποίοι εμπλουτίζουν τα εξαγόμενα ενδιάμεσα χαρακτηριστικά για κάθε καρέ στην ακολουθία, ενισχύοντας τα σημαντικότερα κανάλια και περιορίζοντας τα λιγότερο απαραίτητα. Δοκιμάζονται διαφορετικοί μηχανισμοί από την βιβλιογραφία αφού υποστούν τις κατάλληλες τροποποιήσεις για να λειτουργήσουν στο τρέχον πρόβλημα. Έπειτα, βελτιώνουμε την επεξεργαστική ισχύ του δικτύου μειώνοντας το φαινόμενο της υπο-εκπαίδευσης που παρατηρείται όταν εκπαιδεύουμε ένα δίκτυο περιορισμένης πολυπλοκότητας σε μεγάλο πλήθος δεδομένων, χρησιμοποιώντας την τεχνική κανονικοποίησης που αφαιρεί τυχαία κάποιους νευρώνες του δικτύου κατά την εκπαίδευση. Καθώς προχωρά η διαδικασία εκπαίδευσης εφαρμόζουμε μια απλή αλλά αποτελεσματική στρατηγική υπολογισμού της πιθανότητας αφαίρεσης ενός νευρώνα ανάλογα με την πρόοδο. Για τις μεθόδους προσοχής, διεξάγουμε πλήθος πειραμάτων για την εξακρίβωση των καλύτερων παραμέτρων, αλλά και της θέσης στην οποία εισάγονται στο μπλοκ, καθώς αυτή επηρεάζει σε ποια χαρακτηριστικά εφαρμόζονται, ενώ για την κανονικοποίηση, εξετάζουμε διάφορες στρατηγικές. Ο συνδασμός των δύο τεχνικών αποδεικνύεται πως είναι πιο αποτελεσματικός από την κάθε μια ξεχωριστά και βελτιώνει αρκετά τις επιδόσεις χωρίς να επηρεάζει την πολυπλοκότητα.

Η διατριβή ολοκληρώνεται με μια σύντομη περιγραφή των συμπερασμάτων που προέκυψαν από κάθε προσέγγιση, συζητώντας τα αποτελέσματα, τα οφέλη αλλά και τις αδυναμίες της κάθε μεθόδου και πιθανές εξηγήσεις και λύσεις για κάθε μια. Κλείνοντας, αναφέρονται γενικές οδηγίες για ανάπτυξη αποδοτικών μοντέλων καθώς και πιθανές κατευθύνσεις για μελλοντική έρευνα.

Chapter 1

Introduction

This thesis focuses on the problem of *speech recognition* (SR), which is the general process of extracting speech from a source signal of one or more speakers, formed by either isolated words or complete sentences. It is a widely-researched computer vision problem with rich history that has been under active research for decades and has recently received increased interest. Advances in deep learning research and progress in computer hardware are the predominant contributing factors for its resurgence as a prominent computer vision area of interest, leading to a plethora of research efforts in the last decade alone. Another contributing factor is the broader availability of large-scale corpora, either uni- (e.g., video-only) or multi-modal (audio-visual content), that have enabled training and scaling architectures based on deep-learning models.

1.1 Speech Recognition

Given a source signal that may or may not contain a speech segment, speech recognition can be categorized into two subsets depending on the modality of signal: *audio speech recognition*, or *ASR* for audio signals, and *visual speech recognition*, or *VSR* for video sources. These distinct cases involve signals that are oftentimes referred to as *uni-modal*, since only one modality is present in the source data, either audio or video. In the special case where a signal carries both an auditory and a visual component, one can apply any form of recognition procedure to decode speech, including a combination of techniques for each modality. This process is more generally known as *audio-visual speech recognition*, or *AVSR* and takes advantage of both data modalities in a complementary fashion, exploiting the distinct strengths of each technique to alleviate their weaknesses, enhancing the overall result. While some common aspects are shared between ASR and VSR, there are also some significant differences that set them apart.

Greater amounts of research effort have focused on ASR than VSR, mainly due to the simplicity of the audio modality compared to video due to the lower dimensions of the input, but also partially due to the amounts of labeled data available, since annotating audio is an easier task compared to video without sound. Another notable aspect that led to this gap in research is the resource cost in terms of computation that is required to process the data. Compared to an audio-only track, a video segment is more demanding on resources depending on its spatial dimensions (resolution). Recognizing speech from processing large video signals in an automated fashion was only made possible in the last few years with the development of powerful hardware accelerators and deep learning models that could take advantage of them.

VSR uses a video source without an audio track as its input with the aim of decoding speech exhibited by one or more targets in the video. In contrast to ASR, relying on visual cues to recognize speech is a more challenging process which involves more sophisticated and powerful architectures in order to produce accurate results. In addition, visual ambiguities between words that are produced from visually similar or identical mouth movements can cause erroneous results. For instance, some consonants share the same visual pattern while being part of words with completely different meanings, or a plural version of a word includes a suffix that is hard to distinguish using only the visual information. The human factor needs to be considered as well, in cases of individual speech patterns and variations which might complicate decoding accuracy. The visual quality of the input also plays an important role in the final output, since the resolution of video, color settings (e.g., contrast or saturation), or visual artifacts can cause mis-classified predictions or even missed words, dramatically affecting results.

Even though these factors greatly affect the overall performance of a video-only speech recognition system and need to be taken into account at the design stage, they do not apply to audio sources apart from variations in dialects and accents, since the audio cues (that are usually non-visual) greatly contribute to successful predictions. Thus, for the speech recognition task, audio signals are more suitable and are generally preferred. However, while clean audio tracks offer better results than visual media when it comes to recognizing speech, the audio stream might not always be available (e.g., in CCTV or other surveillance footage) and the presence of noise significantly affects ASR accuracy (e.g., crowded environments or background ambient noise), while the efficacy of a VSR system remains unaffected. In this case, a combined approach (AVSR) is preferred over one that utilizes a single modality due to its robustness to various occlusions that affect the recognition performance of one modality.

Applications of SR in real-world problems cover a wide span of domains with numerous benefits. A notable application with profound impact is found in medical assistance, where a SR system can be utilized to provide assistance for patients that are speech-impaired or are suffering from an inability to communicate effectively. In the same spirit, various instances of existing accessibility platforms (e.g., mobile devices and human-and-machine interaction interfaces) can be enhanced with the addition of a SR system, improving the everyday lives of many individuals.

In entertainment, SR systems have also seen widespread adoption by various video hosting platforms for automatic generation of video transcriptions. Another application in this industry is found in the archiving and digitization of older films, where SR methods have been employed to generate subtitles without relying on a human expert. More recently, software offering all forms of SR (audio and/or visual) has been used extensively to assist digital content creators and other individuals, streamlining the process of adding captions to videos, increasing user outreach and interaction.

Another domain that can greatly take advantage of SR systems is that of personal security, where they can be utilized to enhance crime prevention or mitigation operations (e.g., large crowd surveillance in public spaces such as airports or conferences). For individuals, spoken or silent passwords can add an additional layer of security when interacting with a device in a crowded or potentially unsafe environment. In forensics, VSR systems can be utilized to detect face forgeries or computer-generated content by detecting inconsistencies in speech patterns.

The emerging domains of smart connectivity and Internet-of-Things can add quality-of-life improvements by incorporating forms of SR that will enable users to control appliances in smart homes or directly affect the behavior of self-driving vehicles (e.g., by voice commands). In an increasingly digital world, where reliance on machines and interfaces for interaction is constantly increasing, applications of SR methods can be beneficial and will be even more important and wide-spread in the near future.

1.2 Research Contributions

The complexity of the dimensions associated with the spatio-temporal input of video data used in VSR (or AVSR) translates into a more challenging task, compared to ASR which processes a simpler signal, and as a result, larger models and more data are necessary to train and deploy accurate visual speech recognizers. Additionally, more information relevant to speech is carried by audio rather than video, demanding powerful models with sufficient representation capabilities to develop VSR-specific architectures in order to extract meaningful information from the data. Simultaneously, in the pursuit of performance, network size and complexity for VSR have seen a consistent upward trend as more powerful and complex architectures comprised of several modules are proposed, limiting potential applications of these models in some scenarios that might require low latency for processing and recognition. A significant majority of the use cases discussed earlier impose several requirements for effective operation that can only be satisfied by lightweight and compact architectures with reduced complexity.

The goal of this dissertation involves developing lightweight architectures with low computational overhead that efficiently and accurately tackle the problem of visual speech recognition of isolated words in video sequences, enabling application and deployment in a wide array of scenarios. This can be achieved by following several approaches, for instance, reducing the size of the model, which impacts memory requirements during loading of weights and running inference, its computational complexity, affecting latency, or both, regulating energy consumption and operational cost.

To that end, we tackle the problem of practical VSR following several approaches, and exploring a multitude of techniques that envelop different aspects regarding lightweight model design and training. Our methods are based on deep learning with a focus on reducing model size and complexity, designing practical architectures that benefit from low sizes in terms of required parameters as well as reduced computational complexity, making them ideal for various practical applications.

1.3 Thesis Structure

This dissertation is split into 7 chapters, and is structured as follows:

Chapter 2 offers a comprehensive and detailed analysis of the literature on visual speech recognition of isolated words from the deep learning perspective, covering all published works that utilize such methods, with emphasis on the most-researched language, which at the time of writing is English. Other languages are not neglected, and are given their own sub-section where datasets are presented. We perform a broad taxonomy of related works according to the method followed, outlining the evolution of VSR research.

Chapter 3 explores network compression as a means to reduce overall model size, creating compact architectures with a wide range of applications. A technique that leverages a sum of Kronecker products is employed in the main components of an end-to-end VSR system to achieve significant reductions in trainable parameters, creating models with vastly reduced size and requirements. Extensive experiments and ablative studies showcase that significant reductions in model sizes are possible with minimal penalties in performance, even at high compression rates.

An additional method that also reduces network complexity is presented in Chapter 4. We adopt an established architecture and use cost-efficient modules in place of standard layers in a customized CNN for visual feature extraction and in the Temporal Convolution Networks (TCN) used for sequence modeling. Furthermore, we propose several TCN-based models with very low computation overhead using a temporal block design that applies computations in one part of the input. Our models are several times more lightweight than other related works and can be adapted for applications in cases with severe hardware restrictions.

A more holistic approach follows at Chapter 5 where a compact yet powerful end-to-end architecture for word VSR is proposed, considering all aspects of model design for this task. For visual feature extraction, we benchmark a multitude of lightweight convolutional networks from the computer vision literature. A standard TCN architecture is employed for sequence modeling and different temporal blocks based on efficient structures are developed and evaluated. We combine both components in an end-to-end unified model that surpasses all other lightweight VSR networks in performance while being less demanding in resources, measured in model size as well as computational complexity.

Chapter 6 builds on the previous one by investigating methods to improve its recognition performance without causing noticeable increases in its overhead. Channel attention modules are added in the sequence modeling component of the architecture and their effect on improving accuracy is evaluated. Next, regularization in the form of dropout is introduced during training, resulting in a zero cost improvement in overall performance. The best-performing approach for dropout application in the compact model is discovered after exploring and comparing several strategies. A wide range of experiments shows that combining both channel attention and regularization brings the most benefit.

Finally, insights on lightweight network design for practical VSR applications, potential directions for future work and closing remarks are offered in the Conclusion.

Chapter 2

Related Work on Visual Speech Recognition

The task of visual speech recognition has been under active research for a few decades. In this chapter we provide a comprehensive analysis of related works in the literature for single word visual speech recognition (VSR) including published methods and datasets. Due to the rapid advances of machine learning research, recently-proposed works utilize a wide array of deep-learning methods exploiting their inherent advantages and increased efficacy as opposed to earlier works that relied on hand-crafted features and algorithms. Another significant factor is the emergence of freely-available large-scale datasets that enable the training of deep architectures and have led to significant performance improvements over the years, as automated methods of VSR far exceed the recognition rates of human professionals.

The predominant strategy to tackle the problem employs a sequence of steps which splits the complex task into a series of sub-tasks that are easier to manage. Speech recognition systems typically follow the above multi-step pipeline which generally begins with a face detection operation that aims to localize the area surrounding the mouth of the speaker. This area is subsequently extracted so that the following steps

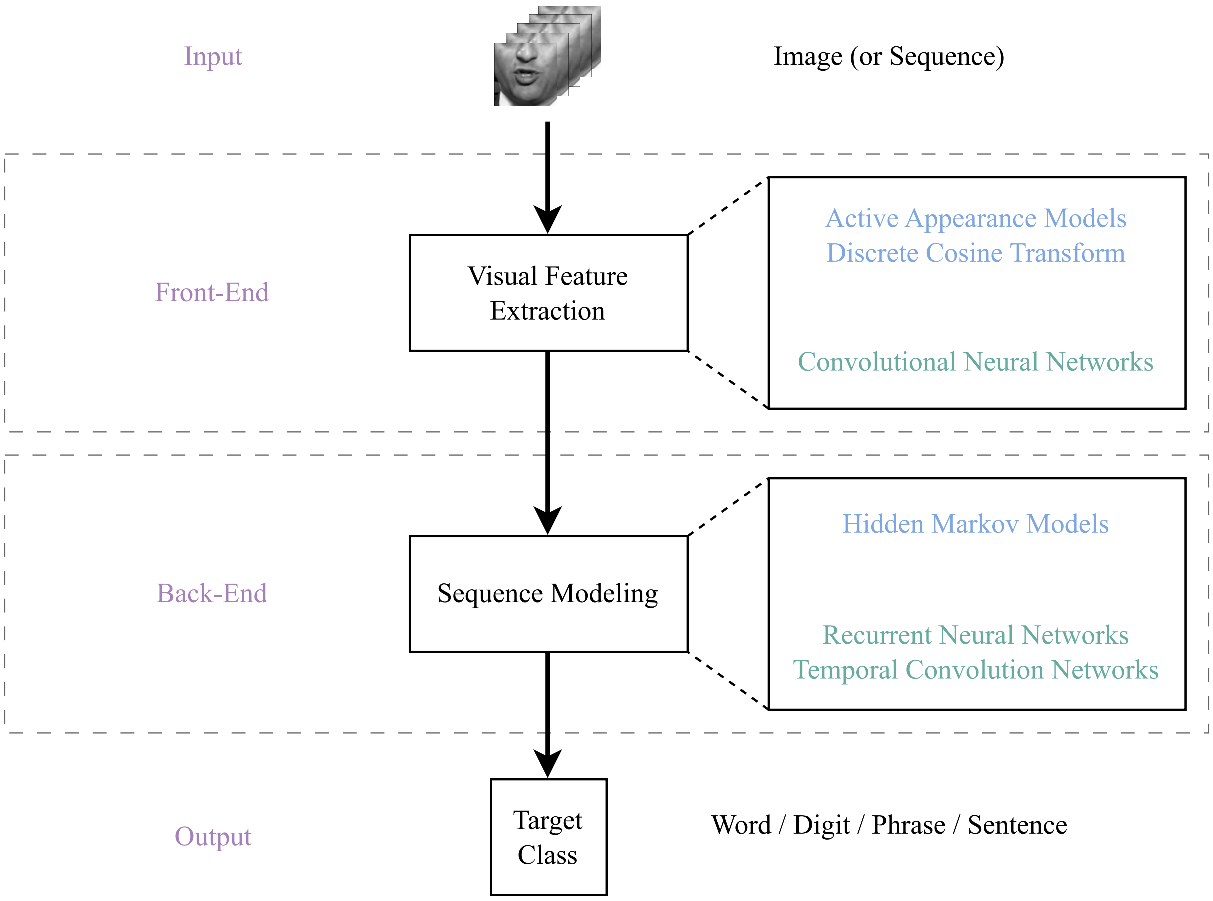
Input
Image (or Sequence)
Front-End
Visual Feature Extraction
Active Appearance Models
Discrete Cosine Transform
Convolutional Neural Networks
Back-End
Sequence Modeling
Hidden Markov Models
Recurrent Neural Networks
Temporal Convolution Networks
Output
Target Class
Word / Digit / Phrase / Sentence

Figure 2.1: General multi-step pipeline followed by speech recognition works in the literature. Commonly-used methods to handle each step are shown in the right, with deep-learning architectures colored in green and previously-used methods in blue. Image best viewed in color.

will operate on an image patch of the input which is offers two advantages: a) smaller spatial dimensions which translates to reduced computation (compared to using the entire input image), and b) it contains all relevant information about speech. After the mouth area has been located and isolated, feature extraction takes place, where meaningful higher dimensionality representations of the input are generated and processed. Finally, sequence modeling handles the temporal relations of the extracted features across time and ultimately the outcome (either a sentence, phrase, digit, or word) is generated by classification, concluding the process. A high level depiction of the general speech recognition sequence followed by works in the literature is shown in Figure 2.1.

While our focus lies on deep-learning based methods, we do not neglect to outline a brief overview of works published before the deep-learning boom.

2.1 Brief Overview of Earlier Works

Earlier approaches to the problem relied on image transform techniques such as *Active Appearance Models* (AAM) [1, 2] or *Discrete Cosine Transform* (DCT) [3] for extracting features, while *Hidden Markov Models* (HMM) were the prevailing algorithm for classification of these features into alphabets or digits in limited vocabularies [4]. Such approaches were constricted in their generalization to recognition of more complex structured speech units and as a result very few works focused solely on single word recognition [5, 6]. For a more comprehensive review of older methods applied to the task of visual speech recognition, the reader is referred to [7].

A limitation of these earlier methods was the low availability of large-scale datasets with rich vocabularies and a sufficient amount of samples to enable more applications. Another issue was the hardware limitations at that time which prevented fully taking advantage of these resource-intensive algorithms, two issues that have since been alleviated.

2.2 Deep Learning Methods and Algorithms for Visual Speech Recognition

More recently, advances in machine learning research as well as progress in hardware capabilities have led to the development of very capable architectures (e.g., convolutional neural networks) that have achieved significant results in several computer vision tasks such as image classification, multi-object tracking, etc. Coupled with the availability of public large-scale datasets for (audio-)visual speech recognition, recently proposed machine learning-based methods developed for this task typically favor such newer architectures instead of older, more constrained ones, benefiting from the improved hardware capacity and the amount of available training data.

The typical end-to-end method follows the same multi-step pipeline described previously where the task of visual speech recognition is split into a sequence of several smaller sub-tasks, each handled in a sequential fashion, by one or more submodules which nowadays employ deep learning models and techniques. Adoption of such architectures has been gradually increasing as research has progressed and performance has improved. Some of the earlier works combined neural networks for feature extraction with HMM-based classifiers, e.g., [8, 9, 10] and [11] were some of the first works that proposed to exploit the capabilities of Convolutional Neural Network architectures for feature extraction purposes while still using HMMs for classification.

Following an extensive examination of the published literature on VSR, to the best of our knowledge, CNNs are invariably used for visual feature extraction purposes and are adopted by virtually all recently published works, while the models employed for sequence modeling have been a topic of research with various architectures being used. We therefore offer a general taxonomy of published works that leverage deep learning models and techniques into four broad categories, according to the method used for sequence modeling: LSTM-based, GRU-based, TCN-based, and methods that employ other deep learning-based models.

One of the most popular designs for speech recognition is the combination of a CNN and a bidirectional RNN [12], where the CNN handles feature extraction and the recurrent architecture classifies the sequence. While these component choices are typically used for their relative simplicity with respect to their performance, other methods such as *Auto-encoders* and *Deep Belief Networks* (DBN)s have also been employed for both sub-tasks.

[8] propose to apply a CNN network instead of a traditional hand-crafted approach, taking advantage of the favorable properties of CNNs such as the utilization of GPU acceleration and the ability to learn from large amounts of available data. The proposed system uses a 7-layer CNN following the *AlexNet* architecture [13] trained on images of the speaker's mouth area and a HMM with Gaussian mixture observation model which treats the generated outputs as feature sequences. Since the dataset does not contain a sufficient amount of data to ensure generalization to data other than the six different speakers it depicts, an independent CNN is prepared for each speaker to improve the overall performance. In contrast, a single common HMM model is used for isolated word recognition.

The same authors explore audio-visual speech recognition in [9]. The proposed architecture utilizes two feature extractors processing inputs of synchronized audio signal and lip region image sequence pairs. A deep denoising auto-encoder [14] is used for audio feature extraction, by filtering out the effect of background noise from deteriorated audio features, while a CNN-derived architecture [13] recognizes phoneme labels from images of lips to generate visual features. The auto-encoder is trained to predict noise-robust clean audio features from samples that have been artificially generated by various strengths of Gaussian noise. The CNN is trained with mouth area image frames in combination with corresponding phoneme labels. For data augmentation, random shift and rotation is applied to the original images of the dataset. Finally, a multi-stream HMM with Gaussian mixture integrates the acquired audio and visual HMMs independently trained with their respective features to perform the recognition.

Takashima et al. [10, 15] opt for a different CNN architecture called *Convolutive Bottleneck Network* (CBN) [16] which combines convolution and pooling layers with a *Multi-Layer Perceptron* (MLP). While a single layer is used for convolution and pooling, the MLP consists of three fully-connected layers, where the second layer has a reduced amount of units and functions as a ”bottleneck”, aggregating the information and acting as a compact feature descriptor, akin to approaches such as PCA. The application is audio-visual word recognition, therefore two distinct CBNs are used, one applied to each modality. For the audio signals, mel-spectrogram maps are obtained, while for the visual signals, a lip image is extracted using a *Constrained Local Model* [17] and interpolated to fill the sampling discrepancy between the features. The CBNs are trained separately and the bottleneck features from both networks are used as inputs of the HMM. The audio CBN outputs phoneme labels corresponding to the input mel-maps and the label data is obtained by forced alignment using the HMM from the speech data.

The same CBN architecture is adopted in [11]. Different than the previous work [10], however, rather than using a static image as input, a dynamic feature of lip images is calculated using the current and its neighboring frames instead. The dynamic feature represents a difference image and provides a more efficient representation of the input by reducing adverse impacts of the background in the image quality, such as shaking, blurring or pose variations. Two variants of the CNN are tested, with varying sizes of feature maps and Convolution-Pooling operation blocks. For data pre-processing, grayscale images of lips are first extracted from the input images using a face alignment model [17], then up-sampled via spline interpolation to increase their spatial dimensions. The CBNs that process the dynamic features to output phoneme labels are trained on the up-sampled images. Then, the bottleneck visual features of the trained CBN, which contain the aggregated information from the input, are used to train a HMM for speech recognition. At test stage, a similar setup is followed, where the bottleneck features are used as input to the visual HMM.

2.2.1 LSTM-based Methods

One of the earliest works proposing the use of LSTMs instead of HMMs for word-level speech recognition is found in [18]. The authors propose a pipeline comprised solely by neural network structures, in a unified model trained by back-propagation, removing the need for manual feature extraction. Their approach uses a feed-forward neural model to extract features automatically, replacing earlier, non-neural network methods and LSTM layers as sequence classifiers, instead of HMM-based speech recognizers. An advantage of LSTMs compared to simple recurrent architectures is their ability to overcome issues regarding the exploding or vanishing values of the gradients during training. The proposed method is applied to the task of single word classification and evaluated against a traditional classifier which uses a Support Vector Machine (SVM) with non-neural network-based feature extraction methods ([19] and [20]). The best performing model uses one feed-forward layer followed by two recurrent LSTM layers with 128 units in each cell.

Chung and Zisserman [21, 22] design several deep network architectures for word recognition. They use the *VGG-M* model [23] as a baseline network due to its performance and running speed. A 3D convolution operator is added at the beginning of the base VGG-M architecture in order to handle the dimensions of the video input ($H \times W \times T$) and experiment with four model setups where information fusion is performed at different feature levels. In a 3D convolution operation, the convolutional and pooling filters operate and move along all three dimensions of the input. An additional architecture is designed where fusion is achieved by including two LSTM layers after the convolutional network. With the exception of the fifth architecture, all other network designs do not utilize an additional module for sequence classification.

The same authors propose a network for audio-visual speech recognition following an encoder-decoder design using deep learning components [24]. The encoder consists of a convolutional module based on the VGG-M architecture [23] generating image features for every time-step and a recurrent module (LSTM) that produces the fixed dimensional state vector. The decoder module uses an LSTM with added dual attention mechanism [25]. The entire model is trained using a progressive learning strategy whereby at the initial stages of the training single word examples are used and as the network trains the sequence length grows. The short sequences are generated from longer sentences in the dataset. This training strategy improves the rate of convergence and reduces over-fitting, raising the final performance.

An end-to-end system for word-level classification is proposed in [26] where the authors combine a residual network [27] with a bidirectional LSTM. As the first component of their end-to-end system, they design a novel spatio-temporal front-end module which is a small neural network utilizing a 3D convolution, Batch Normalization and Rectified Linear Unit activation. Finally, a 3D pooling layer (maximum pooling applied at each dimension) reduces the spatial size of the extracted 3D feature maps. The front-end applies a spatio-temporal convolution to the input frame stream, the 34-layer residual network is applied to every time step, one feature-map per time-step progressively lowering the spatial dimensions and finally, two stacks of two bi-directional LSTMs (one stack per direction) are applied and the outputs of the final LSTMs are concatenated.

The same authors propose a method that extracts deep word embeddings [28] to summarize the information that is relevant to the task of speech recognition, while discarding irregularities such as speaker variation, pose orientation and illumination conditions. Building on their previous work ([26]), they adopt the same model architecture consisting of a spatio-temporal convolution layer, a residual network and a BLSTM with a few modifications to extract the embeddings. More concretely, they replace the 34-layer network with a smaller variant using only 18 layers, reducing the overall parameters, while a pooling layer aggregates the temporal information and extracts a single embedding per video and for regularization, dropout and batch normalization are added to the BLSTM. Word boundaries are also included as an additional feature and fed to the back-end. The generalization effectiveness of the produced embeddings is evaluated on words that are unseen during the training process. For this task, the *LRW* dataset [21] is split into two disjoint sets (of 350 and 150 words), where the first set is used for training. Embeddings are obtained for the remaining 150 unseen words, which are then modeled using *Probabilistic Linear Discriminant Analysis* (PLDA) [29] enabling class estimation.

An audio-visual speech recognition method which combines residual networks equipped with spatio-temporal input layers and bidirectional LSTMs is presented in [30]. The authors perform audiovisual experiments using both intermediate and late integration, as well as several types and levels of environmental noise, and note improvements over the audio-only network, even in the case of clean speech. A detailed analysis on the utility of target word boundaries, as well as on the capacity of the network in modeling the linguistic context of the target word is also provided. As a front-end, the model utilizes a modified 18-layer residual network [31], in which the first spatial (2D) layers are replaced with their spatio-temporal (3D) equivalents and the original average pooling layer is replaced by a fully-connected layer. A two layer BLSTM designed without peepholes, an average pooling layer which aggregates information across time-steps, a fully-connected and a Softmax layer comprise the architecture's back-end.

A novel three-part deep network for automatic speech recognition was proposed in [32]. The hybrid architecture combines deep convolutional neural networks with attention-enhanced recurrent models to better model the hidden correlations of the spatio-temporal information while using the attention mechanism to weigh each key-frame according to its importance. A VGG-19 network is selected as a convolutional feature extractor for its robustness to noise and visual distortions, while an Attention-LSTM [33] models the long-term dependencies within the data sequence. The attention mechanism functions as a focusing signal, emphasizing the important information in the sequence. An experimental audio-visual dataset was constructed for the purposes of evaluating the proposed model, containing data from 6 different persons that were not native English speakers, while the audio data was used to accurately separate the video stream containing only the spoken word. The vocabulary involves the numbers from $0 - 9$ in the English language, each pronounced up to 100 times by every speaker.

Word-level speech recognition for the German language was explored in [34]. Several convolutional neural network (CNN)-derived architectures, including a spatio-temporal CNN, an LSTM and a hybrid architecture combining both in a two-stage network were trained and evaluated on a dataset containing words in German. Two experimental setups were used, where the first corresponds to seen speakers, i.e., appearing on the training and testing sets, while in the second some speakers appear only on the testing set. Each model was trained with the former setup and the best-performing one was selected for evaluation on the latter setup. Hyper-parameter tuning for all models was performed via Grid Search, controlling aspects of the convolution layer such as kernel size, as well as aspects of the training process such as learning rate and initialization functions. The final hybrid architecture consists of 3 LSTM layers with 256, 128 and 128 hidden units respectively. Methods to improve regularization such as dropout were added to increase accuracy and of the three architectures, the two-stage network achieved the highest result.

Convolutional LSTMs are first explored for visual speech recognition by Courtney and Sreenivas in [35]. The Convolutional LSTM [36] maintains the original LSTM formulation [37] that uses several gates to control the flow of information across its cell, however it substitutes the matrix multiplication operations with convolutions retaining the spatial information across the sequence length. In contrast to 3D convolutions that have access only to a limited temporal amount of the input sequence, the convolutional LSTM is not hampered by this limitation. The authors design two architectures derived from convolutional neural networks ([23] and [31]) by replacing certain layers with convolutional LSTMs in order to take advantage of temporal information at several spatial scales. Redundant LSTM layers are then replaced by typical 2D convolutions in order to reduce the overall computation costs without compromising performance. The proposed models are evaluated for word-level speech recognition and pre-training in large-scale sentence-level datasets is found to improve accuracy.

In [38], a novel lip-reading model which captures not only the nuance between words, but also styles of different speakers is presented. It uses a two-branch multi-grained spatio-temporal network to model the speaking process. First, both frame-level fine-grained features and short-term medium-grained features are extracted by the visual front-end of the network, which are then combined to obtain discriminative representations for words with similar phonemes. The fine-grained features extractor adopts a residual network architecture, while the medium-grained relies on a previously proposed model using dense connections [39]. A bidirectional convolutional LSTM [36] augmented with temporal attention aggregates the spatio-temporal information in the entire input sequence, capturing coarse-grained patterns of each word for added robustness to various conditions such as speaker identity and lighting variations. By making full use of the information from different levels of granularity in a unified framework, the model is not only able to distinguish words with similar pronunciations, but also becomes robust to appearance changes.

An application of visual speech recognition in a unconstrained driving scenario that constitutes an application in real-world conditions is proposed in [40]. The authors design a recognition pipeline to recognize words from various speakers inside a car cabin that could represent commands to an automated system that assists with or even drives the vehicle. The recognition pipeline consists of a Convolutional Neural Network (CNN) followed by a BLSTM and a Fully-Connected Network (FCN) with a plain attention mechanism [41]. After mouth detection has been performed and the area has been cropped, the CNN consisting of 18 residual layers encodes the images into visual representations which are then fed to the BLSTM to encode the entire sequence with a many-to-one mapping. The encoded sequence is finally mapped to a word category with the two-layer FCN. Two protocols evaluating the generalization capabilities of the pipeline are devised whereby in the beginning of the training process some samples of each word are not used and later all samples are used while some speakers are excluded, facilitating a realistic scenario of adaptation to new users.

The authors of [42] propose a novel model for word-level classification. A *Convolutional Auto-Encoder* architecture [43] are used for feature extraction from the frames of the video sequence followed by an LSTM where its output is converted to probabilities for word classification. As a baseline for comparisons, the authors first pre-train a convolutional neural network model with the objective of image classification on a dataset containing images that may or may not depict human lips. After pre-training, they modify the CNN for feature extraction by removing the classifier at the top and ”freezing” its weights (i.e., preventing further updates) while the LSTM is trained. Their proposed model replaces the CNN with a Convolutional Auto-Encoder (CAE) which is similarly trained in two phases. In the first phase, images containing lips are used to train only the CAE with the objective of input reconstruction. Once this process finishes, the convolutional decoder part of the CAE is discarded and the remaining part (the convolutional encoder) is paired with the LSTM. The second training phase involves using the encoder of the CAE as a feature extractor to train the LSTM.

A two-stream lipreading model is presented in [44] focusing on recognizing the word being spoken given only the video but not the audio. The presented architecture relies on a two-stream deep 3D convolutional neural network structure that expands on shallow 3D CNNs, taking advantage of grayscale video and optical flow. For the grayscale video the mouth region is extracted and the optical flow is pre-computed. Each stream is used as input to a separate deep 22-layer network using 3D convolutional layers, 3D max pooling layers and a series of 3D inception modules [45] and then the produced features are concatenated and used as input in a two-layer bidirectional LSTM. At the last layer of the latter, the features are concatenated from two directions, averaged along the temporal dimension before being passed through a linear layer to map the score for each word. The networks used are trained in two rounds, first by expanding pre-trained weights into three dimensions and then by pre-training on a large-scale video dataset [45].

An end-to-end method for speaker-independent speech recognition is proposed in [46]. In order to improve model generalization to unseen speakers for more practical applications, the authors propose decoupling content from motion information so that the learned representations more descriptive and related to speech instead of appearance. A two-stream architecture is employed where one stream ingests frames related to motion, while the other handles the content so that two distinct representations are produced independently. In order to create motion-related frames, pixel-wise subtraction of neighboring frames is applied to the input. The two-stream network consists of two identical stacks of feed-forward layers with added dropout for regularization that are then concatenated into a single joint representation and subsequently used as input in a final LSTM layer. The outputs of the LSTM are masked allowing only the last frame of the sequence to be used by a Softmax function for word-level recognition. Experiments with varying amounts of training data corresponding to source and target speakers are conducted to evaluate the speaker-agnostic model.

A simple DenseNet-based [47] architecture for Mandarin visual speech recognition is proposed in [48]. The problem is segmented into two distinct steps, the first aiming to produce a Hanyu Pinyin sequence from the input, then the final Chinese character sequence is obtained with a deep neural network model. Two models are proposed, each handling a different step. The first sub-network employs a spatio-temporal (3D) convolution and 121-layer DenseNet feature extractor, while a BLSTM with residual structure [49] utilizes the shallow features. The shortcut connections allow for information propagation across the several layers of the BLSTM. A linear and Softmax layer are added at the top to obtain character probabilities. In the subsequent step, the goal of transforming the Hanyu Pinyin sequence to Chinese characters is realized by a stack of blocks containing multi-head attention and feed-forward layers, as in [50]. The sub-networks are trained separately and later unified and evaluated on word- and sequence- level classification.

Fusion strategies for effective audio-visual word-level recognition are explored in [51]. Several architectures for fusing clean video with noisy audio are evaluated, namely, *intermediate* fusion, where two single-modality sub-networks comprised of fully-connected layers are used and their outputs are concatenated and fed to a block of fully-connected layers with an LSTM for sequence modeling and *fixed-weight* fusion, where an LSTM follows each sub-network and concatenation is performed at the logit level. Further, two variants of a new fusion method are introduced where in the first variant, the sub-networks for each modality are initially trained, then frozen and used to train another network on joint audio-visual classification, while in the second variant, the same network architectures are used with the only difference being the training process is performed end-to-end, training all networks simultaneously. The sub-networks for single-modality tasks follow [52], with two fully-connected layers and all LSTMs use 128 neurons, regardless of where they are applied (depending on the fusion strategy). The last frame of the LSTM output is used for training and back-propagation by masking the remaining frames, since the task is world-level recognition and a 51-neuron linear layer assigns probabilities to each class.

In [53], a novel method to learn speaker-agnostic visual representations is proposed. The method aims to learn disentangled and unbiased visual features from the data without associations to specific appearance characteristics. This can benefit the generalization of the model since it reduces biases towards the training dataset as well and enables learning from lower amounts of available data. Towards this goal, the training process involves disentangling the speaker identities from the lip movements by leveraging large quantities of unlabeled audio-visual data. This is achieved by using two latent vectors to represent the identities and movements, modeling the training objective as a speech prediction task, where a video encoder network [26] predicts the corresponding audio by learning only the speech-relevant features from the disentangled representations. The visual features are extracted and clustered into groups and then replaced by the closest cluster centroid, a process that removes data-specific information such as speaker identities. Meanwhile, a speaker embedding network [54] pre-trained on the task of speaker verification extracts information about the speaker identities. After the training process, the visual encoder is used for feature extraction in the task of word-level visual speech recognition on several datasets.

A visual speech recognition method tailored to medical applications in the Malayalam language using deep networks is proposed in [55]. The method involves visual feature extraction via a deep convolutional neural network (CNN) and a sequence classifier using a Bidirectional LSTM. Contrary to other works that employ several pre-processing steps in order to reduce the spatial dimensions of the datasets while retaining as much information about the mouth area as possible, the authors do not use any form of cropping or re-scaling other than normalizing the video duration using external manual software. The CNN, following the architecture of [56], is applied to the raw video data which is converted into an image sequence ranging from 30 to 90 frames. The final feature vector for all frames is obtained from the last pooling layer and fed to the BLSTM with 200 hidden unis with an additional dropout layer for regularization, before classification. The method was evaluated on the proposed dataset containing medical words in Malayalam in two settings involving speaker-dependency, as well as in an external speech recognition corpus.

A different approach to conventional recognition from frontal images of the speaker was proposed in [57]. Rather than using frontal images of the face, a wearable necklace with a built-in infra-red (IR) camera was used to capture images of the speakers' neck and face from below, presenting a realistic non-intrusive application of a recognition system that can utilize only the video stream. An end-to-end architecture combining a convolutional network with a recurrent sequence model was developed for silent word-level speech recognition. For visual feature extraction, an 18-layer residual network is employed. The extracted features are subsequently padded to an appropriate fixed size and fed to a single-layer unidirectional LSTM with 128 hidden units. The final prediction is obtained by a fully-connected layer superseding the LSTM. Dropout with a probability of 50% is added to avoid over-fitting. To increase the overall model's robustness to noise, spatial augmentations such as random affine transformations such as rotation and scaling are applied. In addition, frame splitting and randomly shifting are used to create new samples as a form of temporal augmentation to improve the network's resilience to temporal variances. The proposed method was evaluated on a user study featuring 54 English and 44 Chinese commands, respectively.

A visual speech recognition system for the Arabic language is proposed in [58]. The authors evaluate three different deep-neural-network-derived architectures to design a robust system that performs best. The first architecture involves only convolutional neural network components and consists of a stack of spatio-temporal

(3D) convolution and pooling layers, whereby the filters progressively increase as the feature resolution is lowered by the pooling operations. At the top of this network, fully-connected layers with Dropout are used to associate the features with words at the output. The second model utilizes time-distributed layers, that apply a different stack of convolution and pooling operations at each time step, with an LSTM sequence model to process the extracted features considering the order of the sequence. In contrast to the previous architecture, this model is shallow, using only a single convolution operation per frame. Finally, the third model builds upon the second one, replacing the LSTM layer with a Bidirectional variant (BLSTM) for improved performance. Evaluation is performed in two modes, using RGB and grayscale data, and the CNN architecture is found to outperform the other two. Finally, to overcome prediction difficulties of the different models, a majority voting scheme is devised where the input is pre-processed into RGB and grayscale and fed to all three architectures (i.e., 6 models in total). The chosen output is calculated by considering the predictions and confidence of each model.

Turkish visual speech recognition is explored in [59]. The authors use the common two-step deep learning network design (convolutional-derived feature extractor and recurrent sequence-to-sequence model), where several architectures are evaluated in order to find the best performing model. While in all experiments, the sequence classification part remains the same and uses a BLSTM with $2,000$ hidden units to increase its performance due to the dataset size, a wide array of convolutional neural networks are tested for feature extraction. These models include deep convolutional architectures, such as *AlexNet* [13], *VGG-16* [60], residual 18- and 50-layer variants [31], as well as more lightweight networks such as *Xception* [61], *ShuffleNet* [62] and *NASNet* [63] and their parameter values are all initialized using pre-trained weights. Several evaluation protocols are used during training and testing in order to evaluate the models' performance and generalization to unseen speakers, including randomly and manually varying which speakers' data are used.

Ivanko et al. [64] apply visual speech recognition in the real-world scenario of driver assistance. In order to effectively discard redundant data from the video segment, a two-stage data pre-processing pipeline is employed where in the first stage, audio is extracted from the source video file and a voice activity detection module is used to spot speech regions even under adverse (noisy) conditions. At the second stage, region-of-interest areas are extracted after the mouth region has been detected and the images are converted into grayscale, normalized and aligned using the histograms. The proposed end-to-end architecture consists of an 18-layer residual network with *Squeeze-and-Excitation* attention [65] blocks, followed by two BLSTM networks. During training, *mixup* [66] with a probability of 40% and a mixing coefficient between $30 - -70\%$ is used for regularization and *label smoothing* is applied for the frames' labels if mixup was not activated for those particular frames. Apart from visual speech recognition, the method is employed in a system that assists in driving by recognizing visual commands from the driver.

An end-to-end speech recognition model intended for industrial applications, such as interaction with robotic interfaces in acoustically noisy conditions, is proposed in [67]. The model follows the standard approach of a two-stage feature-extraction and sequence-modeling architecture design. Since the model is intended to be deployed in real-world conditions, an automatic pre-processing pipeline is applied where the mouth region is automatically detected and cropped [68], then converted to grayscale and finally normalized and aligned using histograms. In a similar fashion to [64], data augmentation techniques such us *mixup* [66] with variable merging ratio and *label smoothing* are also employed for increased robustness to realistic scenarios. The end-to-end model involves a residual network with channel attention [65] and a spatio-temporal initial layer, which is a commonly-followed architectural choice (e.g., [69, 70]) to extract visual features, while a 2-layer Bidirectional LSTM with 512 neurons is employed as the sequence modeling unit. Dropout is added after the residual network as well as in each BLSTM for regularization.

In [71], a multi-modal approach exploiting complementary information from both modalities for speech recognition is proposed. The input sequence is separated into the two streams (video and audio), and both modalities undergo a pre-processing step which for the visual stream splits the sequence into a series of lip region images, while log-Mel spectrogram data is extracted from the audio signal. Each stream is normalized and fed to a respective deep-learning feature extraction module for further processing. In order to improve the overall network performance, experiments to select an optimal model and fusion strategy were conducted. Several candidate deep networks were considered for each stream and the best-performing was chosen. Similarly, multiple information fusion strategies were explored, including fusing the different modalities at the prediction-, feature- and model-level. Overall, the fusion mechanism employs a fully-connected network that receives the outputs of each feature extractor, which are fused by concatenation, and then makes a word prediction. An end-to-end shared training schedule is followed to benefit the joint learning of both modalities [72].

In [73], the authors propose a method that leverages visemes, which are groups of phonetically similar lip shapes, to improve performance in word- as well as sentence-level speech recognition in the Persian language. As a sequence of visemes can correspond to several words hampering the accuracy of the model and extracting the information in those visemes can increase its robustness by producing more descriptive video features. The method is used to fine-tune the visual extraction component of a two-stream model [74] by taking advantage of the corresponding visemes. Model training uses a viseme-to-character transformation instead of the typical character prediction which can suffer from inherent language-specific biases, which is expected to be more discriminative for the task of speech recognition. First, the phoneme sequence is obtained for each video, then it is mapped to a viseme sequence and provides a stronger representation than character-level fine-tuning. The visual features are encoded by a fully-connected and BLSTM network to obtain a hidden state vector which is then passed to a decoder with two LSTMs with attention. In order to evaluate the effectiveness of the method, the fine-tuned model is tested on a different language than the one it was trained on.

A novel model for Mandarin Chinese visual speech recognition is proposed in [75]. The authors design an end-to-end deep network combining several architectures and mechanisms that follows the standard multi-part paradigm. First, for visual feature extraction, a spatio-temporal convolutional neural network is used, followed by an LSTM-derived encoder-decoder architecture [76] with 3 layers and 256 cells each, that transforms the lip features into textual representations. An attention model assigns weights to the encoded sequence, focusing on the more important features. Word embeddings are mapped into learnable vectors with an embedding model [77] and acts as a character decoder. A novel Mandarin Chinese sentence-level dataset is constructed to train and evaluate the proposed model. Finally, an additional evaluation of the method is performed on the word-level datasets *LRW* [21] and *LRW-1000* [39].

An audio-visual method for speech recognition was proposed in [78]. The method operates in with separate as well as a multi-modal inputs, adapting to various application scenarios. In the case of audio-only inputs, an acoustic model with two fully-connected layers followed by a CNN with 1-dimensional convolution, pooling and normalization layers generates acoustic features, while for video-only inputs, the model used for visual extraction utilizes two LSTM layers with 128 units each. Three fully-connected layers are added after every uni-modal feature extractor for classification. When used with multi-modal inputs, the video stream is split into two modalities and pre-processed to obtain MFCC features from the audio stream and grayscale outer lip area for the video stream which are then input to their respective models. Fusion of the two modalities is performed by concatenation of the feature maps extracted from the first fully-connected layer of the audio network and the first LSTM layer of the visual network, respectively. An additional three fully-connected layers follow, the outputs of which are concatenated as well in order to integrate the information.

A large study evaluating a series of deep convolutional neural networks for visual speech recognition of isolated words in the Greek language is conducted in [79]. The authors collect a novel corpus comprising representative commonly-spoken words and develop several deep architectures personalized for each speaker in the dataset. Various networks from the image classification literature are evaluated, including the *InceptionV3* [80], *VGG-16* [60], *ResNet-50* [31] and *MobileNetV2* [81] architectures, along with customized convolutional networks and recurrent layers (LSTMs). The training procedure involves using pre-trained CNNs with or without the LSTMs, as well as further training on the proposed dataset following a transfer learning scheme. The experimental evaluation shows that the customized architecture that combines a 3D convolutional model with an Bidirectional LSTM sequence classifier with 512 hidden unit size yields the best results. Finally, a generalized model using data from all speakers is trained on a subset of the corpus consisting of the most frequent words among speakers.

In [82], a driver assistance system using audio-visual speech recognition to recognize commands is introduced. The model operates on a bi-modal fashion, using inputs from two different streams, which are then temporally fused. Audio features are extracted from a series of spectrograms using an 18-layer residual architecture adapted to audio data (2D convolution operations substituted by 1D equivalents), while for visual feature extraction, the pre-trained model proposed in [71] is used. The latter architecture first applies a standard 18-layer residual network (with 2D convolutions) to the input, then feeds the visual representations to a Bidirectional LSTM network for sequence processing. The spatio-temporal fusion strategy involves combining the features from each separate modality at the temporal level rather than common approaches such as simple feature concatenation or cross-modal attention. This way, information about the context is preserved and the issue of synchronization between the two modalities is alleviated. Furthermore, a regulated Transformer-derived [50] network with an encoder-decoder structure uses the fused features in an iterative process to first initialize and subsequently refine its weights through fully-connected and self-attention layers. A multi-prediction strategy where predictions obtained from several model iterations for each instance of the input are considered, is employed to improve performance. The method is evaluated on two corpora, for audio-visual command recognition in a driving context on *RUSAVIC* [83] and for world-level speech recognition on *LRW* [21].

2.2.2 GRU-based Methods

Petridis et al. [84] propose an end-to-end architecture for audio-visual speech recognition that relies on residual networks and Bidirectional Gated Recurrent Units (BGRU)s and consists of two streams, one for each input modality (audio and video), extracting the features directly from the data. Each stream, apart from the convolutional feature extractor, uses two BGRUs with 1024 cells instead of LSTMs and the visual component employs the previously proposed 3D front-end and 34-layer residual network following the work of [26]. For the audio stream, a simple 18-layer residual network with 1D convolution layers is used. The outputs of each stream are fused in a stack of two additional BGRUs, incorporating sequence information from both modalities. Each stream is first trained independently and then unified in a single architecture to train the entire model in an end-to-end fashion. The method is evaluated for word-level speech recognition on the *LRW* dataset [21] with isolated modalities as inputs as well as their combination.

Speech recognition for word- and sequence-level Chinese Mandarin is explored in [85]. The authors propose a Cascade Sequence-to-Sequence Model that exploits language tonality when predicting sentences by using syntax as well as visual information to model tones, which modify the pitch of the spoken language to convey different meanings. The task is divided into three sub-tasks and thus the overall model combines three separate sub-networks which are jointly trained where each sub-network is dedicated to a sub-task. The first sub-network is used to predict a pinyin sequence, which is related to mouth movement, from the video and follows an attention-based sequence-to-sequence architecture with an encoder and decoder [86]. The next sub-network predicts tonality, which can reduce viseme ambiguities by leveraging the pinyin information produced by the first sub-network and the input video with a dual attention mechanism [24]. Finally, the last sub-network combines the video input alongside the outputs of the previous sub-nets to predict a Chinese character sequence with a triplet attention mechanism. All sub-networks use a two-layer BGRU with a cell size of 256 as encoder and a two-layer standard (unidirectional) GRU with a cell size of 512 as decoder. To improve learning, *curriculum learning* [24] and *scheduled sampling* [87] are employed.

Yang et al. [39] introduce a novel dataset for word-level speech recognition of Chinese words and perform a quantitative evaluation of three types of deep neural network models with different components used for word recognition. The evaluated models follow different architectures of previous works and include a fully-two-dimensional convolutional model with LSTMs ([22]), an architecture with 3D convolutional layers as a feature extractor and a sequence classifier ([88]), as well as a network that utilizes a combination of both designs ([26]). Following the architectural designs of [88] and [26], the authors propose combining 3D convolutions with dense connections [47] in a fully-three-dimensional residual model for feature extraction, replacing the standard 2D counterparts for image classification. For sequence modeling, the same structure of 2-layer Bidirectional GRUs as in previous works is adopted, keeping the comparison between the different architectures fair. The fully-three-dimensional dense model is evaluated on the newly-proposed dataset.

In [89], the authors present a framework for training visual speech recognition models that are unaffected by the pose of the speaker. They propose using a 3D morph-able model [90] to generate synthetic data of facial images in arbitrary poses from a source image depicting only the frontal pose. Their synthetic data augments existing datasets using only frontal views of the speaker and circumvents the costly and time-consuming process of annotating ground truth data. The network architecture ingests the mouth region of interest (ROI) image inputs consists of a 3D convolutional block performing spatio-temporal convolution, followed by a residual network [31] for automatic feature extraction and a 2-layer BGRU to model the sequence. The authors opt for the smaller (18-layer) network due to its faster training duration and equal performance to the larger 34-layer one. Their method achieves significant improvements in cases where extreme poses (over 45 degrees in yaw and pitch) are present.

Zhang et al. [91] investigate whether regions in the face other than the mouth area can be exploited in order to improve the performance of visual speech recognition systems. Citing psychological studies on human perception where it is suggested that humans recognize speech by relying on more facial information rather than just the mouth area, the authors study the performance impact of four manually-selected regions when used as inputs on several speech recognition models ([26, 30, 88]). These regions include information from areas other than the mouth (e.g., the cheeks), that is potentially beneficial to recognition, but is overlooked by the standard practice of cropping only the mouth area. They propose using *Cutout* [92] as a simple augmentation strategy that partially occludes regions in the images during training, encouraging the models to focus more on the remaining areas. Their findings demonstrate that including extra-oral regions of the face as input data during training can yield stronger features, thus boosting performance.

In [93], a method that takes into account parts of lip images for increased robustness is proposed. The method leverages parts of the lip image that correspond to different speech-related characteristics to reduce speaker-dependence. A 34-layer residual network with a spatio-temporal layer is used as the feature extractor, while a 2-layer BGRU with 512 units is utilized as the sequence model. The visual features are split into three parts according to the lip structure and each part is fed into a sequence model so as to obtain independent temporal representations of each area. Learning is modeled with a joint loss that takes into account each part and enables the end-to-end training of the entire architecture. An audio-visual framework utilizes the above network as its visual feature extractor alongside a parallel audio branch that follows [84] and two BGRUs with 1024 units each fuse the obtained features. Furthermore, the model is evaluated on a subset of *LRW* [21] with fewer classes to simulate practical applications in industrial scenarios.

The authors of [94] propose a novel pseudo-convolutional policy gradient (PCPG) based method to train visual speech recognition models. This policy aims to alleviate two problems in traditional training process of sequence-to-sequence models: a) teacher exposure bias and b) inconsistencies between the optimization target and the final evaluation metric. Character error rate is introduced as a form of training reward which optimizes the model together with the original discriminative target.

Simultaneously, a pseudo-convolutional operation is performed on the reward and loss dimension, generating a robust reward and loss for the whole optimization of the model. They base their architecture in an encoder-decoder scheme following previous designs [84]. The encoder contains a spatio-temporal network, a residual feature extractor and a bidirectional GRU, while the pseudo-convolutional decoder uses a 2-layer GRU which is regarded as an agent with the ground truth as the environment. During training, the agent takes into account the old state observed from the time steps and takes an action to output a new character or word to obtain a new state, so that both states and the environment contribute to the reward together for choosing the action. Finally, the reward is fed to the PCPG module to generate the final loss when passed to the agent.

[95] propose an audio-visual fusion method using spatio-temporal graph convolutional networks (ST-GCN)s [96]. A hybrid visual extraction network utilizing a dual-branch structure combines the properties of GCNs which can exploit the relationships among key points of the lip shape, with the traditional neural network-extracted visual representations. In that network, the graph branch, utilizing a ST-GCN and BGRU, extracts additional shape-based features related to the lip area augmenting the discriminative high-level visual features obtained by the other branch, which employs a 34-layer residual and 2-layer BGRU combination. To improve the expressiveness of the model by fusing the information from both modalities, the output of the visual network is mixed with the features extracted from an audio stream with the use of a novel attention-based bidirectional fusion module that consists of a synchronization block and a 2-layer BGRU. The synchronization block uses a series of convolutions and attempts to align the features from each modality reducing the temporal asynchrony between them that hampers recognition performance.

In [97] the authors aim to improve representation learning by introducing mutual information constraints during training. Constraints are enforced on both the local feature level enhancing fine-grained movements at each time step and on the global sequence level encouraging the model to learn latent patterns. Combining these constraints aims to enhance the relations of the features with the speech content improving the effectiveness of the representation and the robustness to noise produced from the change of pose, lighting conditions, speaker appearance and speaking speed. The model used in this work follows the same feature extraction network as in previous works (e.g.[30]), while for the back-end a 3-layer BGRU to capture and classify the latent patterns of the sequence. The local mutual information maximization acts on patches of the feature maps produced by the convolutional extractor, while the global mutual information maximization acts on the representations obtained by the BGRU.

Xiao et al. [98] propose a two-stream network trained in a self-supervised manner to learn the deformation flow between adjacent frames in a sequence, which directly captures the motion information within the lip region. The lip movements during speaking are modeled as a sequence of apparent deformations in the lip region. The learned deformation flow is generated by feeding the source and target images in a encoder-decoder module (using a residual network) and is then combined with the original grayscale frames in the two-stream network to perform speech recognition. The architecture uses two branches, both following [26] with gated recurrent units instead of LSTMs, however the first branch employs 2D convolutions for the deformation stream instead of 3D ones, which are used in the second branch. The two-stream network is trained using a novel bidirectional *knowledge distillation* [99] loss, enabling the simultaneous exchange of information between the two branches which are trained jointly during the learning process.

[100] propose an audio-enhanced multi-modality speech recognition model with the goal of enhancing the audio quality in noisy environment scenarios where the impact of audio information in the overall recognition quality is reduced. Their model uses two sub-networks, each fulfilling a different role; one sub-network enhances the visual representation by learning to separate a speaker's voice from the background noise, while the other performs multi-modal speech recognition. Similar to their other work ([101]), their visual feature extraction network uses a pseudo-three-dimensional residual network [102] architecture and for recognition, the element-wise attention GRU [103] is chosen for its effectiveness over long sequences. In the visual speech enhancement sub-network, visual and audio features are extracted from the mouth region of interest (ROI) and magnitude spectrograms of the audio, respectively. The extracted features are fused by concatenation and fed to the GRU and the resulting enhancing attention mask is multiplied with the original noisy spectrograms.

The authors of [104] propose a method to alleviate issues related to the synchronization of the audio and video data in multi-modal speech recognition systems. Rather than following the standard approach that modifies the sampling rate of one modality to match the other one which can miss inter-modal relations, they propose a method that aligns the features to generate new aligned audio-visual sequences which can be used as input features by the sequential models (e.g., BGRUs). The method applies an iterative procedure of mutual attention steps, where features from one modality guide the alignment process of the other one and vice versa, without being mixed in the process. The method relies on a modified *Transformer* encoder [50] using a stack of multi-head attention and fully-connected feed-forward layers with added sinusoidal positional embeddings that encode information about the positions of the features within the sequence. The whole model involves one branch for each modality, following commonly-used architectures ([31]) and BGRUs for sequence modeling. After features for each modality have been obtained, mutual attention aligns them and two BGRUs fuse the aligned features before final classification into words.

In [105], a hybrid fusion method is proposed to increase robustness in audio-visual speech recognition by exploiting the complementary nature of both modalities. The hybrid fusion method takes advantage of *feature fusion*, which utilizes the variations between frames of different modalities for a more discriminative representation and *decision fusion*, which determines how to optimally combine the different modalities under noisy conditions. First, audio and video features are extracted from audio waveforms and mouth region of interest (ROI) sequences by two separate deep networks. Then an audio-visual fusion network using a lightweight residual architecture projects audio and video features into a common latent space, discovering the internal correlations between the different modalities and an audio-visual BGRU models the joint sequence. Finally, a decision fusion module fully utilizes those correlations by examining the outputs produced by the audio, video and fusion encoders. A two-step training procedure is followed, where each sub-network is first trained separately, then the unified model is fine-tuned in an end-to-end process with a joint loss function that optimizes the robustness of learning joint representations across both modalities.

Similarly, a feature fusion network for audio-visual speech recognition that extends single-stage approaches is presented in [106]. An additional early feature fusion stage is added, allowing the network to benefit from fusion at different stages. The model utilizes three separate streams and a late audio-visual feature fusion part, integrating the complementary features of each stream and increasing the model's robustness to noisy scenarios. The video stream ingests frames and utilizes a previous architecture [84] with an added spatio-temporal non-local block [107] at the beginning which captures long-range features of lip frames, while the audio stream receives audio corrupted by random babble noise. The features obtained by the residual networks in each uni-modal stream are concatenated and fed to the third stream for early-stage fusion with a two-layer BGRU. Finally, the fused features along with each uni-modal stream's BGRU outputs are concatenated and combined with two additional BGRUs for late-stage fusion. A three-phase training process is employed, where in the first stage only the three distinct streams are trained independently, then their weights are frozen while the late-stage feature fusion part is trained separately and finally the unified network is fine-tuned in an end-to-end manner.

In [108], a method to improve the feature extraction capabilities of the spatio-temporal (3D) layer commonly adopted by speech recognition works is proposed. The authors introduce a *Temporal Shift Module* (TSM) [109], that extracts spatio-temporal features by shifting the feature map channels along the time dimension, to two different deep convolutional neural network architectures. The first architecture consists of an 18-layer residual network using only 2D convolutions, without the spatio-temporal first layer, while the second architecture uses a combination of 3D and 2D convolutions, which is a common design principle in the literature (e.g., [110, 91]). For both setups, the sequence modeling network is a 2-layer BGRU with 1024 hidden units. Since the TSM could potentially lower the spatial feature extraction capability of the 2D CNN due to the amount of channel shifting, the authors experiment with several variables that control the amount of shifting and the modules are inserted into each residual block, to preserve the representation strength of the networks. Moreover, the authors study the influence of the amount of shifting and data sampling interval, which controls the temporal density of the frames used as the input to the model.

In a similar fashion, Li et al. [111] also propose enhancing the convolutional extractor with additional temporal capabilities without introducing much computational overhead via the *Temporal Shift Module* (TSM) [109]. The TSM shifts parts of the feature channels forwards and backwards in the temporal dimensions, mixing the information of the current frame with that of its neighboring frames and capturing contextual information about the speech pattern. The authors use the popular setup of an 18-layer residual network with a spatio-temporal first layer to keep running costs low and add this module in each residual block to enable the convolution operations to extract spatial features between sequential frames. A possible limitation whereby some correlations and relationships between these sequential frames could be ignored by the temporal module is alleviated by the introduction of a channel attention mechanism [65] that re-calibrates the channel activations at the end of the residual block to filter the useful features. The sequential model that follows the enhanced feature extractor is a simple 3-layer BGRU with 1024 units.

Self-supervised learning with a proxy task is proposed in [112]. This task guides the learning of semantic visual features from unlabeled data. The purpose of the task is to seamlessly exploit the semantic correlations between the audio and video modalities. In order to jointly learn temporal and cross-modal semantic correspondences from the unlabeled data, a multi-task framework is presented where a novel proxy task guides two simultaneous predictive coding sub-tasks, each involving predicting target latent features from the given input by learning its context vectors. For the intra-modal predictive coding task, the model learns to predict future samples from the past ones and vice versa, while in the cross-modal sub-task the objective is to predict latent features for one modality using the other. A pair of residual encoders generates latent features for audio and video, using 18 and 34 layer architectures, respectively, while the context vectors are learned from the input using three different GRUs (audio-only, video-only and audio-visual). Fully-connected layers are finally used to decode the feature embeddings from the outputs of the GRUs, following [113]. As the focus lies in visual representation learning, after the self-supervised training stage the audio encoder is discarded and the method is evaluated on the visual speech recognition downstream task.

The authors of [69] propose a pipeline for visual speech recognition by incorporating various training strategies to gain as much improvement as possible, without resorting to significant architectural changes of the model. Several factors are considered, including the scheduling scheme of the learning rate during training, how the dataset is pre-processed and the choices for different sub-networks used in the final model. Variations of convolutional and recurrent neural networks are considered for feature extraction and sequence classification, respectively. For data augmentation, *mixup* [66] is employed to reduce over-fitting. Word boundaries [30] are also included which improve performance. Their final model which performs best consists of a 3D convolutional layer, followed by a residual network with four *Squeeze-and-Excitation* [65] modules for the front end and a 3-layer BGRU for the back end.

[114] explore large-scale visual speech recognition in the Russian language. The authors propose a novel dataset consisting of 235 words and use it to evaluate sev-

eral deep-learning-based published works from the literature [39, 98, 110, 69, 115]. Among those, most are evaluated on the *LRW* [21] dataset, while [115] is evaluated on the related task of action recognition and adapted to the current task of VSR following principles from [26] and [116]. The evaluation involves using pre-trained weights as well as training the models from scratch on the newly-proposed dataset and [39] is found to outperform the other methods. Selecting this model, an ablation analysis is presented where several training tricks, such as *label smoothing* [80] and data augmentation techniques such as *CutOut* [92] are examined to increase performance. Furthermore, a novel architecture derived from [69] and involving split-attention [117] is proposed and evaluated on *LRW*.

Another application to the medical domain is presented by Sen et al. [118], where the authors design a personalized network for a patient suffering from a neuro-degenerative disease that results in the progressive loss of motor neurons that control voluntary muscles and ultimately the loss of voice. Since collecting and annotating enough data to train a personalized deep neural network is a time-consuming process, a pipeline for creating personalized datasets for such applications using a set of unique words in a one-shot scenario is proposed. First, a set of unique real examples is gathered from the patient as well as three additional speakers that include sign language and deaf users that also mouth the words. A lip synchronization network [119] generates speaker-specific synthetic examples and text-to-speech techniques are employed to augment the one-shot data. For speech recognition, an established backbone network [69] is used and trained on a combination of synthetic and real data using a novel *Variational Encoder* to bridge the distribution gap between the different forms of data.

Distillation is proposed in [120] as an effective method to improve recognition performance by taking advantage of audio data to alleviate some ambiguities of visually confusing phonemes which are produced by similar mouth movements. In order to transfer more diverse features from the audio teacher to the video network, the proposed method incorporates sequence- and frame-level distillation considering partial as well as overall information about the entire sequence. Since the features from the different modalities correspond to unequal sequence lengths, a Gaussian-shaped weighted average is used to learn a mapping between them using a normal distribution that does not hamper efficiency, compared to more expensive mechanisms like attention. Audio samples are extracted from the dataset and augmented to create more training examples and used to train the audio model, which consists of three residual blocks, a fully-connected and a BGRU layer. After the teacher model has been pre-trained, it provides an additional training signal via distillation to the visual-only student model, which follows the architecture of [69].

In [121] a multi-modal framework that can utilize both audio and video inputs with the goal of augmenting the lip visual information by recalling corresponding speech audio data from its saved memory is proposed. The authors adopt a multi-modal *memory network* [122] that stores source-key and target-value memories, each containing pairs of encoded audio-visual features. In order to allow the framework to recall the saved modality (target) by utilizing the stored memory of the other (source), they propose a method to associate between the memory buffers, thus allowing the recalled (multi-modal) information to complement the input (uni-modal) at inference time. The proposed framework maintains separate latent-space representations for each modality and is able to use a multi-modal representation for the input even if one modality is not available. A baseline architecture [84] is combined with the proposed memory network and trained in an end-to-end manner, while evaluation is carried out on two uni-modal downstream speech-related vision tasks, namely isolated word recognition and silent-video speech reconstruction.

In [123] a robust algorithm for visual speech recognition is designed. The proposed pipeline involves a series of steps pertaining to the utilization of the data which includes the processes of acquisition, processing and classification. In the pre-processing stage, the mouth region is first extracted from each frame and color information is removed to keep the number of features low, followed by a lip detection operation carried out by a *hybrid active contour* approach [124, 125] utilizing a novel edge detection function optimized for lip images. Due to the flexibility of the mouth feature, a robust parametric model [126] composed of cubic curves is selected to represent a precise contour of the mouth area. The pre-processed data is fed to a four-stage neural network architecture combining spatio-temporal convolutional neural networks with a stack of two Bidirectional Gated Recurrent Units of decreasing hidden unit size, global average pooling and classifier. Apart from visual speech recognition, the architecture is evaluated on the downstream task of lip segmentation.

The authors of [70] introduce a novel dataset for daily Mandarin conversations as well as a system for robust visual speech recognition with the goal being the application in a medical environment. The dataset contains 1000 videos from 10 distinct speakers and each video is recorded in a controlled environment where the speaker is facing the camera. The video quality is high, featuring a frame rate of 30 frames per second with a spatial resolution of 1920×1080 pixels. To increase the size of the dataset, face landmarks are extracted and various spatial transformations are applied resulting in a total of 10.000 samples, segmented in a $7:2:1$ split corresponding to training, testing and validation sets, respectively. The proposed recognition system includes a spatio-temporal convolution layer, an 18-layer residual network augmented with *Squeeze-and-Excitation* blocks [65] and a global average pooling layer to extract features which are then fed to a module with a 3-layer Bidirectional GRU followed by a series of 1D convolution layers.

In [127], *Slow-Fast* networks [128] are employed for speech recognition. The authors note some key difficulties when applying Slow-Fast networks on the domain of speech recognition and propose methods to alleviate these issues. A dual-stream network is proposed that captures subtle lip motion features by exploiting a Slow-Fast structure where the two streams extract dynamic as well as static features. In order to better exploit the information in lip motion sequences, two sampling methods are devised that complement each stream, improving its feature extraction capabilities. The dual-stream network follows the Slow-Fast Net architecture, implemented with an 18-layer residual network with an added *Temporal Shift Module* [109] in each residual block, while for sequence modeling a typical 3-layer BGRU setup is used. When combining the streams, two fusion strategies, early and late, are considered where in the former the feature vectors are concatenated before being input to the BGRU. In contrast, for late fusion, each stream is considered independently and the features are fused after sequence modeling.

In [129], the authors explore an alternative approach to automatic visual speech recognition by using data provided by event cameras [130]. An event camera produces data at a higher temporal resolution compared to a traditional (RGB)-type camera. A novel multi-grained network with two branches, low-rate and high-rate, operating on data of different frame rates is proposed. The branches enable the network to perceive features of different temporal resolutions taking advantage of the properties of both. While architecturally the branches consist of a spatio-temporal layer and several residual blocks, the high-rate branch uses fewer convolution filters to keep the amount of computations manageable. Due to the different feature granularities of the two branches, a message flow module is proposed to allow feature communication between them, fusing the information with an attention map. The message flow module is added after each pair of residual blocks, fusing the information at several spatio-temporal scales. A Bidirectional GRU receives the fused output of the last message flow module to model the sequence. The method is evaluated on test samples of a novel event-driven dataset, according to three protocols: 25 frequently mis-classified word pairs from *LRW* [21], 50 randomly selected words from *LRW* and on the combination of both sets.

A novel method using 3D convolutional vision transformers is proposed in [131]. In a *Convolutional Vision Transformer* (CVT) [132], convolution projection operations are added to the original vision transformer [133] in order to model the local spatial context of the input without considering the temporal dimension information. The authors combine a 3D convolution neural network with the CVT backbone aiming to extract representations from local features (adjacent frames) as well as from global ones (long-distance frames). Spatio-temporal representations are obtained from the input by the 3D CNN which are then fed to a CVT backbone extracting both global and fine-grained feature information utilizing the self-attention mechanism and convolution operations respectively. A Bidirectional Gated Recurrent Unit (BGRU) with 1024 hidden unit size models the video sequence.

In [134], the authors present an application of an audio-visual speech recognition system in a virtual aquarium setting. A wireless head microphone records user audio that is then fed to a speech recognition API transforming the list of spoken words into vectorized representations. Simultaneously, a frontal camera captures the mouth movements and extracts a frame sequence that is used as input to a deep network architecture to obtain visual features. For the extractor model, the authors propose a novel architecture that combines a typical 3D spatio-temporal network as its first layer with a 3D dense block architecture [47], where each convolution layer is replaced by a 3D equivalent operation and a final multi-scale 3D network. The latter utilizes a multi-branch design, where three parallel branches receive the same input and their outputs are concatenated in order to extract features at different scales containing information of several granularities. Each branch combines a stack of *convolution—normalization—non-linear activation* layers with a spatial attention module [135] that enhances the more important channels for improved performance. *Dropout* [136] and its spatial derivative [137] are included, each in one branch to prevent overfitting. A sequence processing module with two Bidirectional GRUs follows after the feature extractor and finally a fully-connected layer receives the merged features from both the audio and visual representations. The method is trained and evaluated on a held-out dataset containing a set of 54 words including control commands, numbers, emotions as well as nouns related to aquatic animals.

In [138], deformation flows driven by the audio signal are explored. The proposed method utilizes the audio modality to guide the flow generation process, focusing more on the speech-relevant information rather than the visual elements. A three-component pipeline trained by a three-step process is proposed where the deformation flow guided by the audio is first generated by a network, then fed to a flow-based visual recognition model and finally to another network that recognizes the speech. The first network to be trained is the flow generator, which uses a bimodal design where two convolutional encoders extract features from each modality and a deconvolution decoder fuses the information to produce the deformation flow which contains speech-related lip movement. Then, the generated flows are used to train the flow-based visual model and finally, this model functions as a teacher network in a *knowledge distillation* framework transferring the information from the audio to the student model which performs the speech recognition. After the training phase, the flow-related networks (generator and recognizer) are discarded and the distilled student is evaluated.

The authors of [139] propose a visual speech recognition model to assist religious practitioners and researchers of the Arabic language. They design a uni-modal network for word-level classification relying only on the input video stream. A two-stage model architecture is adopted that uses a 18-layer residual network with a spatio-temporal first layer for feature extraction, combined with a Bidirectional GRU [140] using 3 layers. In addition, they introduce a novel dataset containing digits as well as words in order to evaluate their proposed model via a transfer learning scheme where the model's weights are initialized from pre-trained values on the *LRW* [21] dataset and trained further. Data augmentation in the form of horizontal flipping and affine transformations are applied to increase the dataset size. During training, scheduling the learning rate according to a cosine schedule [69] is also employed. Apart from the word-level task, the model is also evaluated on single or disjoint letter recognition.

The authors of [141] introduce a novel framework for silent speech interactions in mobile devices relying on visual speech recognition. The framework combines contrastive and few-shot learning strategies in order to learn robust representations via self-supervision and to seamlessly adapt to unseen examples. A pre-trained encoder-decoder architecture following [69] is employed as the model and fine-tuned with a contrastive learning objective [142] in order to learn an embedding space between the samples. To evaluate the model, a customized corpus was designed containing words as well as phrases related to mobile phone interaction commands. Furthermore, a mobile application was developed enabling command customization via a keyword spotting method that detects new keywords from the lip movements. An incremental learning scheme was introduced enabling the continuous improvement of classification accuracy as more samples are provided. The overall framework was evaluated on a user study involving pre-defined commands in English as well as in an unrestricted language scenario.

Sun et al. [143] proposed a mobile system for silent speech recognition following a different approach. The system leverages a built-in microphone and speaker from an earphone to detect the ear canal deformations that occur during speaking by analyzing ultra-sound signals. The in-ear speaker transmits audio in the ear canal which is then reflected backwards and captured by the microphone and subsequently synchronized to obtain clear reflection signals. A segmentation algorithm detects the deformation events according to the short-term energy variation and ear canal deformation features are extracted with a combination of a *Transfer Function* (TF) and *Continuous Wavelet Transform* (CWT) containing complementary aspects of different granularities. A two-channel hierarchical neural network aggregates the two distinct features by employing a stack of GRUs [144] for the former and a lightweight convolutional architecture derived from [145] for the latter, which are then concatenated to fuse the information. In order to avoid the time-consuming process of gathering and annotating large amount of ground truth data required to train deep networks, as well as to increase the model's robustness to noise, the authors apply several augmentation techniques for the different features (TF and CWT). While jittering is used to add noise to the original data for the TF features, time warping, masking and time-frequency masking are employed for the CWT. The method is evaluated on two datasets covering letter- and word-level recognition tasks.

In [146], a bimodal noise-resistant system combining several input methods is introduced. An automated interface recognizes word lists from an audio stream which are then converted into vectorized representations with a pre-trained word embedding deep model [77]. For recognition utilizing only the video stream, a convolutional network with two branches receives an image and its corresponding spectrogram for robustness to noisy environments. This network utilizes a dense spatio-temporal CNN with spatial attention [135] and a 2-layer Bidirectional GRU. The first component extracts discriminative visual features from the input by leveraging the spatial attention module to focus on the information with the strongest spatial correlations while the BGRU facilitates the fusion of information between the different modalities. A previously-published dataset [134] was used to evaluate the proposed method in controlled conditions. In addition, experiments where noise was added into the audio input simulating realistic applications were performed to evaluate the method's robustness to noise.

Similar to some previous works (e.g., [108, 111, 127]), *Temporal Shift Modules* (TSM) [109] for visual speech recognition are revisited in [147]. The authors expand on the original module by introducing separate channel shifting ratios, allowing for a more refined information exchange between channels within the same residual block, which in turn enables efficient temporal interactions between neighboring frames. The model proposed in [148] that uses a spatio-temporal 18-layer residual network with a DC-TCN is employed as the baseline and the authors add the improved TSM at the beginning of each residual block to extract temporal information. In the improved TSM, a shifting ratio that varies according to the residual block stage is chosen that causes the least amount of feature loss. Moreover, the global average pooling layer that comes after all residual blocks in the visual feature extraction network is replaced by a 3D convolution layer that expands the effective receptive field of the architecture, harmonically integrating feature information from different temporal dimensions. Another architectural change compared to the baseline is replacing the DC-TCN for sequence modeling with a 3-layer Bidirectional GRU and adding a channel attention mechanism [65].

A dual-stream speech recognition approach combining several architectural innovations is proposed in [149]. The model adopts a two similarly-structured networks comprised of a 3D convolution, residual network and *Temporal Shift Modules* [109] in a dual pathway SlowFast [128] architecture. The *slow* path adds a second 3D convolution after the residual network, while the *fast* one utilizes a pooling operation instead. The residual blocks in each CNN are augmented with attention modules in the form of Temporal Shift Modules with varying shifting ratios following [147] as well as *Squeeze-and-Excitation* [65]. In order to effectively combine the spatial information of different granularities obtained by the two pathways, two fusion methods are investigated. The first method fuses features by concatenating each pathway across the channel dimension and then using a Bidirectional GRU with three layers of 1024 hidden units, while the second method adds two BGRU models, one for each path. The end-to-end model is evaluated for word recognition in two languages, English [21] and Chinese [39].

A novel architecture following the typical three-step design for VSR is introduced in [150]. For visual feature extraction, the authors employ a standard 34-layer residual network, where the first layer is a 3D convolution, followed by normalization, non-linear activation and pooling layers. The residual blocks are augmented with a novel spatial attention method that uses parallel execution paths, adopting design elements from [65] and [135]. For sequence modeling, a Bidirectional GRU is combined with a *Mamba* [151] model that employs selective attention mechanisms for improved information extraction and both networks receive the visual features. Contrary to the standard approach in the VSR literature that converts color images to grayscale to be used as inputs, this method operates on the original (RGB) images instead and uses the conversion to grayscale as an additional augmentation technique during training. The network is trained with *focal loss* [152] rather than the commonly-used Cross-Entropy to handle imbalances between the dataset classes.

2.2.3 TCN-based Methods

The authors of [153] explore a self-supervision strategy to learn audio-visual embeddings without labeled data since collection and annotation of training datasets is an arduous and time-consuming task. To learn meaningful embeddings, the task is modeled as a cross-modal retrieval problem, where input in one domain guides a process of finding the most relevant sample in another domain. A novel self-supervised training method is proposed where the network learns cross-modal embeddings by being trained for unlabeled multi-way matching by leveraging similarity-based methods and multi-class loss functions, rather than the typical pair-wise ones. A two-stream network is used where a visual stream component is paired with an audio counterpart that extracts multiple features to facilitate the one-to-many feature matching task. The network learns to minimize the embedding distance between the audio and video, encouraging a low distance for a matching pair to be far from all negative (non-matching) pairs. The method is evaluated on the audio-visual synchronization task which requires time-aligned modalities, on cross-modal biometrics (e.g. [154]), as well as on the downstream task of word-level visual speech recognition.

A visual speech recognition model aiming to improve on previous works is presented n [110]. The authors address the limitations of models using bidirectional gated recurrent units (e.g., [84]) by replacing those units with *Temporal Convolutional Networks* (TCN) [155] which are sequential models making use of 1D temporal convolutions receiving a time-indexed sequence of feature vectors as input and mapping it into another such sequence, preserving the initial sequence length. The authors propose a variant of the vanilla TCN network, called Multi-Scale TCN (MS-TCN) utilizing three parallel branches in each block, where each branch uses a different kernel size providing the network with several receptive fields. Each branch splits the input and operates on a subset and their outputs are concatenated, effectively fusing the information from multiple temporal scales, allowing the network to better model sequences, compared to the original architecture. They also simplify the training procedure by adopting a cosine scheduler [156] to anneal the learning rate allowing the model to be trained from scratch in one stage. The whole architecture consists of a standard residual network where the first layer is substituted by a 3D convolution (instead of the standard 2D), followed by their proposed Multi-Scale TCN and finally a Softmax layer. To reduce the training times, they pre-train on a subset of the 10% hardest words of a dataset, which they claim allows for faster training and even yields a small performance improvement while adding a minimal training overhead.

An application of visual speech recognition in the medical domain is presented in [157], where the goal is to recognize spoken words from patients. A novel end-to-end two-phase deep learning model is proposed utilizing convolutional and Temporal Convolution Networks (TCN) for increased performance compared to RNNs. An 18-layer residual network is employed as the feature extractor and a fully-connected layer is used to lower the feature dimension before the TCN. The architecture of the latter consists of 15 layers, each utilizing convolutions with kernel size equal to 9 with a dilation rate of 2 for a large receptive field compared to the typical approach that uses kernel sizes of 3 or 5, allowing the network to exploit more features from the sequence. The residual network's weights are initialized by pre-trained values and are fine-tuned on a subset of randomly selected words from the *LRW* dataset

[21] before training on the Greek Words Medical dataset. An additional comparison between the TCN and the LSTM model for sequence classification is performed and the TCN is found to achieve higher accuracy.

Chen et al. [158] propose a novel deep learning architecture using *hierarchical pyramidal convolutions* [159], replacing the standard layers. These modules apply kernels of different increasing spatial sizes (e.g., 3, 5, 7, 9) encouraging multi-scale processing during the feature extraction as each kernel extracts feature maps of a different context leading to improvements over the model's ability to discover fine-grained lip movements. Their novelty includes local and global feature maps which are utilized with a hierarchical connection, where the local feature map is used as a part of the output as well as an input for the global feature extraction. A consensus method using self-attention [50] is also employed (in place of average pooling) to merge information from all time steps within a sequence focusing on frames that are more relevant to the annotated word than those that are not. Their architecture follows a previous work [110] that uses pyramidal convolutions in the residual network and the self-attention after the MS-TCN, taking advantage of the combination of both proposed novelties to improve its classification capabilities.

A novel end-to-end network module is designed in [160] where spatio-temporal (3D) and spatial (2D) convolutions are alternated to learn effective features from the data. Since the module consists of a sequence of both spatio-temporal and spatial convolutions, it takes advantage of the properties of each type learning the data relations in the temporal dimension. Conversion components from each convolution type (3D to 2D and vice versa) are added in order to enable the 2D convolution to operate on the feature maps and to restore their temporal relations. The module produces a sequence-to-sequence mapping with the original length preserved, by first applying a 3D convolution, a conversion from 3 to 2 dimensions, followed by a series of 2D convolutions, an inverse (2 to 3 dimensions) conversion and a final 3D convolution. The module can be inserted into existing architectures and the authors experiment with several front-end setups including using the module exclusively or combining it with a residual architecture, while for sequence modeling they employ a Multi-Scale TCN [110].

Audio-visual complementary data is exploited in a self-supervised manner by Sheng et al. [161]. The authors propose a framework combining contrastive with adversarial training that takes advantage of information from both modalities of the input data producing more discriminative visual representations for speech-related downstream tasks such as visual speech recognition. For contrastive learning they design a novel loss where the objective is to discern real from noise samples during training, while for adversarial learning they propose two pretext tasks to encourage the disentanglement of the representations from information related to identity and modality. As for the network architectures, a simple 34-layer residual model with the standard 3D spatio-temporal block is used for encoding the visual information and the VGG-M [23] is used as an audio encoder. Multi-Scale TCNs [110] are adopted (one per network encoder) to aggregate speech information from the extracted representations. The effectiveness of the method is evaluated on the downstream tasks of word and sentence-level visual speech recognition.

Dense connections [47] were added to the temporal blocks used in the *MS-TCN* model in [162] with the aim of overcoming drawbacks of the previous architecture [110]. These connections allow a convolution layer to receive inputs from all its previous layers within the same block. Additionally, the new dense temporal blocks incorporate convolutions of several kernel sizes with a varying dilation rate per convolution in each block, as opposed to the previous multi-scale design [110] that uses the same hyperparameters for the convolutions within the block. Channel attention in the form of *Squeeze-and-Excitation* (SE) blocks [65] is added at the start of each block for increased performance. This model, named Densely Connected TCN (DC-TCN) utilizes these temporal dense blocks for enhanced expressive capability since each layer has access to the receptive sizes of all previous temporal convolutions and is able to use more information from different effective receptive fields.

Spatio-temporal Graph Convolutional Networks are introduced in [163] to explicitly model the mouth contour deformations that contain semantic information about speech. A two-stream sub-network is designed to produce representations from the input via two separate streams of different granularities. The first sub-network utilizes a typical spatio-temporal convolutional neural network with a 3D convolution and a residual architecture to extract global motion features from the input, while the other sub-network leverages computed landmark key points from the face to exploit local information from around the lip area. A novel adaptive graph convolution network building on [164] takes into account the landmarks along with encoded features related to local motion and coordinates to model their semantic spatio-temporal relationships. The extracted representations from both streams are added and fed to a sequence modeling network that takes the form of two distinct structures, depending on the task at hand and maps the fused features to natural language. The method is evaluated on two visual speech recognition problems, at sentence and word-level, where a *Transformer* [50] architecture and a Multi-Scale TCN [110] network are used, respectively.

The capabilities of TCN networks for VSR are extended in [165]. A drawback of the original design related to the limited receptive field of the temporal convolution layers in the early stages of the network is noted and a series of architectural modifications are proposed to address it. Due to the small kernel size used by the TCN, its ability to extract time-relevant information over long intervals is hampered and a multi-dilation design scheme is proposed to effectively combine data from several time scales. Building upon the TCN architecture of [110], for the various branches with different kernel sizes for the convolution operation, two sub-branches with different dilation rates are used in each temporal block. This formulation is applied only to the early stages of the network, while the original dilation rate is kept in the later stages, which have adequate receptive fields. Self-attention is added after each temporal convolution block to better utilize the implicit inter-relations of the lip movements and positions in each sequence. To accelerate the training process, a mini dataset comprising the 50 hardest words of *LRW* [21] was created to evaluate the models.

In [166], the authors propose a novel model to tackle two inherent challenges of the task of visual-only speech recognition due to the insufficient information related to lip movement and the produced speech. The proposed architecture involves a double-stream network augmented with a novel multi-head visual-audio memory module that saves cross-modal information in order to model the relationships between the audio and video representations. For the visual front-end, the common residual spatio-temporal and BLSTM combination is used, while the audio front-end is designed with two convolution layers and one residual block. The back-end follows the *MS-TCN* [110] and is augmented with the proposed visual memory module applied at four different levels to extract the relevant context at several temporal scales. The network is trained with audio-visual datasets in an end-to-end fashion, while during inference only visual input is used and the model recalls the inter-modality relations from the memory module and extracts the relevant information, using audio to complement the visual representations.

A novel two-branch network for Cantonese visual speech recognition is proposed by Xiao et al. [167]. The global branch extracts coarse information from the whole lip area, while the local branch captures subtle fine-grained details regarding motion and deformations. The proposed model extracts features using a standard spatio-temporal layer followed by an 18-layer residual CNN, which are then fed to the double-branch sequential network. Each separate branch receives the feature map produced by the residual network and while the local branch splits the features into three parts according to the real space (i.e., center area, left and right of the lip), the global branch leaves them unchanged. *MS-TCNs* [110] are used in both branches with the difference being the local branch using three, one for each area of the lip, to the global branch's one, in order to capture temporal variations in the appearance of each lip area that occur during the sequence. A bidirectional *knowledge distillation* loss, akin to [98], is employed to jointly train both branches and increase performance.

A collaborative learning approach, where two network branches work in combination to complement each other's weaknesses is proposed in [168]. A two-branch model is designed to take advantage of the information in a lip image from several spatial dimensions. In contrast to using the whole image of the mouth area which contains global, coarse-grained information, splitting the image in small-sized segments can convey local, fine-grained information that is typically ignored. A commonly-used spatio-temporal residual network is first applied to the image input to extract a compact feature map, which is then fed to the two different branches. The extracted features are used without any pre-processing by one branch to model the whole lip area encapsulating global spatial information while for the other branch, they are split in three chunks guided by the corners of the mouth area in order to model more localized information. A fusion module is added in the part-branch to adaptively weigh the partial features according to their affinity to those of the global branch [169]. The two branches are trained jointly with collaborative learning (e.g., [170]), where each branch provides an additional supervised signal to the other, enhancing its own representation ability in a similar manner to *knowledge distillation* but without the need for a pre-trained teacher model.

Akin to [91], Zhang et al. [171] propose a network that leverages multiple views (spatio-temporal and spatial) of the input to produce more powerful visual representations. To that end, a two-stream network is used to integrate the multi-view features of the input that contain representative information about the lip appearance and shape. A spatial-only (2D) and a spatio-temporal feature map are extracted by the the spatial view and the spatio-temporal view streams, respectively, which follow residual architectures. In addition, local heat-maps are predicted using a regression model [172] and utilized along the global 2D features by a spatial graph model in order to learn the lip topology and position-specific relationships between the lip landmarks. Two sequential models using the MS-TCN architecture are added after the spatio-temporal branch and the output of the graph convolution, modeling the long-time context of the sequence and integrating the shape-related information of the lip area, respectively. A decision-level fusion method integrates both streams by weighing the two branches and the optimal weight is learned during training on out-of-dataset samples.

In [173], the authors propose an audio-visual representation learning approach that exploits the compounding information in multi-modal data. A fusion module based on a stack of a single hidden-layer *Perceptron* and twelve multi-head attention *Transformer* blocks models the long-term context dependencies of the data. The multi-modal alignment information is preserved by concatenating the audio and visual embeddings in the feature dimension. The embeddings are extracted from two modality-specific networks, following the architectures from [174] and [110], respectively, then down-sampled to the same length and masking [175] is applied before fusion to encourage cross-modal learning. At test time, the modality that is not relevant to the task at hand can be masked out in the fusion module, allowing the model to be applied to problems of a single modality. The method is evaluated on speech recognition using one or two modalities after pre-training is completed on audio-visual data.

Yang et al. [176] introduce architectural changes in a sequence modeling network commonly-used in previous works to better handle noise present in the input data. The initial architecture follows the standard spatio-temporal convolution and 18-layer residual network and the Multi-Scale Temporal Convolution (MS-TCN) model [110] with a *Multi-Head Visual Memory* module [166]. A Temporal Shrinkage unit designed to filter noise channels according to a learnable hyper-parameter threshold combines a Residual Shrinkage Building Unit [177] with 1D dilated convolutions to extract temporal features. A residual network is also added to the unit before the pooling operation to retain spatio-temporal relationships between the data. To alleviate the potential information loss that occurs when average pooling is used to fuse features across the temporal dimension, after the MS-TCN, the authors employ a *NetVLAD*

[178] layer that clusters and weighs the extracted features according to the distance between them and the cluster center. The entire network is evaluated on *LRW* [21] as well as a custom dataset of Chinese words, without and with various levels of noise.

A novel network utilizing deformable convolutions and temporal attention for speech recognition is proposed in [179]. The authors incorporate deformable 3D convolutions in a residual block structure aiming to extract more accurate spatial information from the input lip area. In a deformable convolution [180], each sampling point in the kernel has a variable location which corresponds to the input, as opposed to the standard design where the convolution locations are fixed. Positional offsets representing these variable locations are first learned using a standard convolution network and then used to generate new sampling grids from the original ones which comprise the deformable kernel. Furthermore, in order to model correlations between the spatial and temporal aspects of a sequence focusing on the more important information, a channel-temporal attention block is proposed. This block uses two components, a *Squeeze-and-Excitation* [65] network which weighs the spatial information along the channel dimension and a query-key-value formulation which attends to the sequence in the temporal dimension. Two designs of the block are proposed, where in the first, both components are applied to the input in parallel, while in the second the temporal attention is performed sequentially after the spatial. The deformable 3D convolution block is added in an 18-layer residual convolutional neural network, while the channel-temporal attention block is added in the Multi-Scale TCN architecture [110]. The method is evaluated on speech recognition in English [21] and Mandarin Chinese [39].

Training strategies that improve the achieved performance of speech recognition models are investigated in [148]. The authors conduct an analysis evaluating the impact of several data augmentation techniques, spatial as well as temporal, that are commonly used by previous speech recognition works (e.g. [110, 69, 162]) on accuracy. They conclude that the most optimal settings for data augmentation include *mixup* [66], where new training examples are generated by a linear combination of two input samples and *Time Masking* [181], where in each training sequence an amount of frames is masked (in this case replaced) by the sequence mean. These techniques are then utilized to train a spatio-temporal residual and ensely-connected TCN (*DC-TCN*) [162] network backbone. Finally, word boundary indicators [30], *self-distillation* [182] and pre-training on larger audio-visual datasets (either self- or fully-supervised) are found to provide an additional benefit, further improving recognition performance.

A framework for improving cross-language representations for speech recognition is proposed in [183]. The authors redesign the spatio-temporal layer that is commonly found in the speech recognition literature in order to facilitate learning of more descriptive spatio-temporal representations, by changing its formulation with a 3D reconstructed kernel [116]. Two design choices for this network component are explored in order to reduce the overall computation cost of this module. As a way to enhance the sequence modeling capabilities of the network by including more short-term spatio-temporal information from the input, a block consisting of multiple 3D reconstruction kernels is added to the feature extraction component of a previous architecture ([110]), while the network used for sequence modeling remains unchanged. A curated dataset containing a balanced amount of samples from *LRW* [21] and *LRW-1000* [39] is created for evaluating the cross-language learning capabilities of the proposed architecture.

A method for speech recognition of mandarin is proposed in [184]. Using a residual network as a computationally intense baseline for feature extraction, the authors replace it with an alternative model that is more lightweight [185]. This architecture leverages grouped convolutions (where the operations are applied to parts of the input) to reduce complexity and channel shuffling as a means to mix information from the different groups without using costly operations. This network greatly reduces the computational cost of the overall architecture as it is combined with an already lightweight temporal convolution network that follows the standard architecture of [155]. To improve the feature extraction performance, an attention module performing spatial as well as channel attention [135] is added to the network. The module first computes weights for the channel dimension, uses them to recalibrate the input and subsequently computes spatial attention weights which are used to scale the overall output.

Similar to [166], Yeo et al. [186] present lip-audio memory in multiple temporal scales to enhance lip-related movement information. The memory module leverages audio signals (both short- and long-term) to generate features from multiple temporal scales, while simultaneously storing an alignment between the visual and audio features. To assist in the feature generation, a temporal audio model is proposed to capture contextual information from a sequence of audio features. By sharing the temporal information as the visual feature, a time-alignment is possible between the different modalities, producing an accurate mapping in the memory module. The multi-temporal lip-audio memory network follows [166] enhancing the architecture with an additional audio temporal model [187] and treats the mapping between visual-to-audio as a one-to-many alignment where one lip movement corresponds to several audio representations. The memory network is added at different stages of the ensely-connected TCN [162] which is employed as the baseline.

In [188], the authors investigate visual speech recognition on subjects with an occluded face (wearing a mask). To that end, they collect a novel dataset including images of masked speakers and to facilitate accurate learning, since the most relevant information is occluded they augment it with images of unmasked speakers. In order to crop the relevant area in the input image that contains the localized lip region of interest (ROI) several landmark detection models are evaluated, including taking into account only the eyes, adding other regions of the face, or using 68-point model that considers the whole face. The extracted ROIs are fed to a visual speech recognition architecture that follows the design of [110] with a 3D convolution and residual feature extractor and Multi-Scale TCN sequence classifier. The model weights were initialized in the *LRW* [21] corpus and fine-tuned on the proposed dataset according to several setups where either masked, unmasked or an aggregation of all images were used for training as well as testing. For each setup, a speaker-independent cross-validation scheme was followed, where images corresponding to one of the 20 speakers in the dataset was held out and used to test the remaining 19.

Chen et al [189] introduce several forms of attention to a Temporal Convolution Network. Since the task of visual speech recognition is highly sensitive to both the spatial and the temporal aspects of lip movements, a novel module is designed to leverage attention mechanisms across the temporal, channel and spatial dimensions of the data, compared to a more standard approach that applies a single form of attention. The module enables the network to focus more on the spatial attributes of the lip shape as well as the temporal relations of the visual features across the length of the sequence through these three supplementary attention mechanisms. Several architectural designs related to the positions of each type of attention modules within the network are considered and the best-performing is found to be the one where channel and spatial attention supersede the temporal attention. The overall architecture follows the typical architectural design of two main components, where the feature extractor is a spatio-temporal layer and an 18-layer residual network equipped with the proposed attention mechanism to extract more discriminative features, while for the sequence modeling back-end the ensely-connected TCN [162] is used. The model is evaluated in word-level speech recognition in both English and Chinese [39].

Class-incremental learning, where the goal is to enrich a previously-trained network with knowledge about new classes without losing any stored information, is explored in [190]. A novel two-stage deep neural network is proposed that simultaneously incorporates new knowledge while retaining already-learned class information. While in the first stage a commonly-used spatio-temporal residual network with 18 layers is employed, for the second stage a dual sequence modeling back-end is added to facilitate the incremental learning process. The multi-step training process of [191] is followed, where the source dataset is split into disjoint subsets which are used to train the model in a sequential manner. After completing each subset, the weights of both back-ends are aggregated and stored in one back-end in order to retain the learned temporal information. A novel *knowledge distillation* [99] variation is proposed to enhance the transfer of stored knowledge from the back-end with aggregated information into the one that is currently trained on the new data. The proposed method is evaluated on an incremental learning scenario that resembles real-world applications using training and testing samples from *LRW* [21] and *LRW-1000* [39] on a subset of the total categories of the datasets.

Huang et al. [192] propose a method to improve the performance of the standard 18-layer residual network that is used for visual feature extraction. The method involves using a two-branch temporal module [193], where one branch collects long-term temporal feature dependencies by operating on the channel-level, while the other focuses on location-sensitive information by enhancing features. A convolution operation aggregates the temporal information obtained from both branches, allowing the network to leverage features of different temporal scales, increasing its robustness. The module is placed into each residual block between the convolution layers to recalibrate and enhance the intermediate feature maps that are calculated at each spatial scale as the architecture gradually re-scales the input. The improved residual network is combined with a Densely-Connected TCN [162] and a classification head for VSR of isolated words.

The authors of [194] offer an approach that combines key frames with long range data dependencies during temporal modeling of the sequence with the goal of reducing incorrect classifications of similar words. To that end, they introduce two modules that form the core of a building block which is used to construct an architecture based on a TCN with dense connections [162]. The first module extracts multi-scale features with different temporal granularities by utilizing two TCN blocks, each using a different set of convolutions, utilizing smaller kernel sizes for more local information and larger ones for a broader receptive field. The extracted features are subsequently fed to an adaptive feature fusion module where attention weights are calculated and used to dynamically re-weigh the importance of each type of feature. Finally, all features are fused together via a summation operation with the input, mixing all information. A sequence of these two modules consists of a single block and the authors retain the dense connectivity of the original architecture, where each dense block consists of four such sub-blocks.

More recently, [195] proposed a method to adapt speech recognition to unknown speakers for more practical applications. Compared to traditional adaptation techniques where costly fine-tuning operations are required, the proposed method does not introduce significant computational overheads and can be incorporated into existing architectures in a plug-and-play manner. In order to better adapt to the new speakers, the method utilizes the appearance as well as the temporal aspect of the lip movement and involves adding a series of components in a baseline two-stage architecture. In the first stage where the visual features are extracted by a deep neural network, decomposition matrices are introduced to the convolution layer to adapt to the new speaker's visual characteristics. After the pooling operation of the network and before the sequential model, a parameter-efficient adapter module [196] enhances the spatio-temporal learning of the features. Finally, a novel adaptive weight module is added to the output of the sequential model for temporal adaptation of the speaker's unique talking habits.

Chen et al. [197] propose frameworks that exploit viseme[1] sub-words for enhanced speech recognition generalization. Their approach involves breaking down each word into a sequence of viseme sub-words and modeling the associations between corresponding frames and visemes, weighing the impact of every specific frame on the final decoded word. The decomposition from word labels to a sequence of sub-word

[1] A viseme is defined as a speech unit with identical appearance during pronunciation and comprises one or more phonemes (their acoustic equivalent), e.g., ”p”, ”b” and ”m” share the same lip movements and constitute a unique viseme [198, 199]. Several viseme-to-phoneme mappings exist in the literature, examples of a such can be found in [200] and [201].

visemes is achieved by a Gaussian mixture model and a Hidden Markov Model system using lip embeddings obtained by a pre-trained deep architecture ([158]). A hybrid framework utilizing multi-task learning is introduced where a standard end-to-end method is combined with the proposed sub-word approach in a unified two-branch model. The hybrid framework takes advantage of the complementary nature of both aspects during joint training to improve recognition performance in cases of speaker head movements. In addition, a collaborative framework utilizes a temporal mask module to capture word and state interactions in order to model the hierarchical relationships of words and their decomposed sub-words. The module operates between the two branches and filters irrelevant visual representations that correspond to noise or silence according to a balancing threshold. The proposed frameworks are evaluated on the *LRW* [21] and *LRW-1000* [39] using several architectures following [108] and [162] for visual feature extraction and sequence modeling respectively.

The authors of [202] introduce a model to alleviate difficulties with feature extraction from the input by considering global features that might affect recognition output. Two apparent shortcomings of typical architectures are highlighted: the oversight of global interactions by the residual network due to a limited receptive field and a design limitation of TCN variants that overlooks continuous information at the local level due to the use of dilated convolutions. With these in mind, they propose a new framework for word VSR that aims to address both issues. First, a global context [203] block is inserted at the residual architecture incorporating global cues into the block and is utilized as an attention mechanism. Then, the authors redesign the popular Multi-Scale TCN block [110] that uses several different kernel sizes with a double branch design where one branch uses dilated convolutions while the other uses regular convolutions. The inclusion of non-dilated operations allows the capturing of temporally sequential information, increasing the representational capabilities of the network. Several experiments regarding the fusion of the two branches are made and an early fusion approach is found to perform best.

[204] introduce an end-to-end cross-modal framework for visual speech recognition that utilizes audio to supervise the visual component during training. A double learning objective is proposed to enhance the representation capabilities of the network. The standard recognition loss is paired with an audio reconstruction loss which aligns each video frame with a number of quantized audio tokens. This combination allows for a more accurate detection of distinct homophenes, which are visually

similar lip movements that correspond to different phonemes and are typically hard to distinguish. An encoder generates a sequence of quantized audio tokens in an auto-regressive manner, while a linear projection layer is inserted after the visual representation module. The latter adopts an architecture consisting of a 3D convolution layer, an 18-layer residual convolutional network and a *Transformer* encoder, similar to previous works [166, 186]. Several models are trained with the proposed method and evaluated on the tasks of word- as well as sentence-level speech recognition.

Gu and Jiang [205] propose a model that exploits complementary information from the asymmetric nature of the mouth lip area, that regards an image as two distinct parts separated vertically at the middle. They use a shared-weight double-stream network which has shared weights, allowing it to learn identical information from the two halves. After extracting short-term features with a 3D CNN network, they are then divided in two parts and each half is ingested by one network stream. The core component of this network is a modified residual block where an attention mechanism [65] is added at the end to comprise the redundancy-aware operation, filtering excess information with the help of a soft threshold function. In addition, the authors propose a module to recombine information from both feature halves, inserting this module after each pair of residual blocks. This module utilizes a cross-attention calculation of each feature half, facilitating interactions and information sharing between them. For temporal processing of the extracted features, they are merged with a concatenation operation and inserted to a typical Multi-Scale TCN [110] followed by an average pooling operation, a fully-connected layer and a Softmax activation function.

The authors of [206] propose a five-step method for efficient fusing of complementary features. First, the mouth region of a video frame is cropped and used to compute landmarks corresponding to the lip area. The model consists of two input streams, one utilizing a 3D convolution and residual network architecture to extract visual features from the image, while the other makes use of two graph attention network layers [207] and a transformer encoder to produce geometric features according to the landmarks. To fuse the extracted heterogenous features, a multi-head cross-attention mechanism aggregates the outputs of the two streams, combining the complementary information of pixels and landmarks. For the final computation step, a Multi-Scale TCN [110] is added at the end of the architecture for sequence decoding.

The method is evaluated on the task of isolated word VSR using an English and an Arabic dataset.

A novel framework leveraging synthetic data is proposed in [208]. The method performs data augmentation by combining audio samples with facial images using a generative model to create a set of videos that contain realistic natural variations. Each image is paired with five different audio clips and used to generate animated frames that are used as additional training data. During training, viseme classification is employed as an additional learning task to improve the model's capability to distinguish between different mouth shapes. The model architecture follows [148] and adds an extra execution branch after the temporal convolution network to facilitate training of the new task. This branch uses an attention mechanism for the embedding and is combined with the standard branch for the final classification features.

A spatio-temporal feature fusion network aimed at modeling short-term temporal dependencies between proximate frames is proposed in [209]. Building on the core of the 18-layer residual network, the refined architecture reduces the overall depth and incorporates spatial as well as temporal features, the former extracted with the original residual block and the latter with two novel modules. The first temporal module captures the connections of close frames and utilizes several depth-wise 3D convolutions with dilation, extracting short-term features. The second captures more global features by enhancing channel information with a 3D *Squeeze-and-Excitation* [65] operation. Temporal fusion is achieved by multiplying the extracted features from the temporal modules with the residual of the input, while for spatio-temporal feature combination all features are added together with the input. Attention pooling is added to the Densely-Connected temporal convolution network of [148] which is used as a sequence model, in order to re-weigh the contribution of time-steps to the overall result.

Zhang et al. [210] propose two methods to improve performance and generalization of visual speech recognition applications. First, a novel data augmentation strategy is proposed where the original audio clip is split into sub-sequences and then time masking [148] is applied to each sub-sequence with randomized properties. Time masking replaces a randomly selected number of consecutive frames of a sequence with a mean frame, which is calculated from the sequence itself. The authors then propose a novel end-to-end model that allows for easier capturing of long-range temporal dependencies. It retains the 3D convolution sub-network at the beginning but removes the standard 18-layer residual network, replacing it with a customized block that combines a *shifted window (Swin) transformer* [211] with dense layers [47], connected laterally. The Swin transformer computes self-attention within local windows and alternates their arrangements in consecutive blocks to introduce connections between adjacent windows. Each dense layer contains three convolution operations. The extracted features are then processed further by a Densely-Connected TCN [148] and the proposed model is evaluated for both English and Chinese word visual speech recognition.

Bai et al. [212] develop a performance enhancing module to improve the feature extraction capabilities of an existing VSR network. A 3D convolution sub-network using two depth-wise convolutions with different spatial and temporal scales is constructed. The output of each convolution is fed to a spatio-temporal attention mechanism that utilizes pooling operations applied in parallel, then concatenates the results and applies convolutions with sigmoid activation functions. The attention-weighed features from each depth-wise convolution are then further pooled, concatenated and fed through another convolution and sigmoid activation for further refinement to obtain the final attention weights. The module is integrated in a standard residual network at each stage and an ablation study is performed to determine the best-performing configuration showcasing that 5 modules bring the most benefit for performance with a small increase in parameters and network overhead. A Densely-Connected Temporal Convolution Network [162] is employed for temporal sequence modeling and the architecture is completed with a classifier.

A similar approach [213], introduces a novel network for improving the performance of a temporal convolution baseline used for sequence modeling. The densely-connected multi-dilation CNN proposed in [214] is used as the baseline and the authors remove the stem layer while adding temporal convolutions to its architecture, in order to adapt the network to the task of visual speech recognition. Observing that naively combining dense connections with convolutions of a fixed dilation is an issue, the authors design a multi-dilation network aggregating information from several receptive fields without overlaps or gaps during feature calculation. The architecture is organized in blocks comprised of an amount of sub-blocks, where each sub-block contains a number of densely-connected temporal convolution layers with varying dilation rates, while the blocks also utilize dense connections and aggregate the information. A 3D convolution and 18-layer residual network are used to extract features from the input.

2.2.4 Other Approaches

Visual speech recognition for the Czech language is explored in [215], where contrary to other works, depth sensor data is also taken into account. Several feature extraction techniques are evaluated, including traditional methods such as Active Appearance Models, Discrete Cosine Transform, Spatio-temporal Local Binary Patterns [216], Spatio-temporal Histogram of Oriented Gradients [217] as well as a Deep Learning model with stacked spatio-temporal convolution, max pooling and non-linear activation layers. All traditional methods are applied on a mouth region of interest which is optimally selected according to landmark configurations that perform best, while the deep learning model operates on square images of the input that cover the mouth area. For the deep learning model, two training configurations are compared: pre-training on an external dataset and fine-tuning on the Czech dataset for transfer learning. A depth-based spatio-temporal model is also developed and trained from scratch only on the Czech dataset. While the focus of the work is sentence-level recognition, the models are also evaluated on the word-level task to optimize the hyper-parameters.

A Convolutional Neural Network-based model without any recurrent architectures for sequence modeling is introduced in [218]. The authors design a twelve-layer CNN comprised of stacks of convolution, activation and pooling layers for end-to-end feature extraction. An additional two layers of batch normalization are added to stabilize the training procedure by reducing speaker variances related to speech (e.g., accent, tonality) and to the image quality (e.g., lighting, resolution). In each video sequence, the lip regions from every frame are extracted using a face detection model and then concatenated to form a single image resembling a mosaic of lips from the entire sequence and the model is trained on this image. Since this input formulation does not constitute a sequence (only the spatial information is retained), the output classes for speech recognition are obtained with two fully-connected layers with dropout that are added after the CNN. The method is evaluated on phrase as well as word recognition.

Zhang et al. [219] propose a system that aims to solve two drawbacks of other approaches: neglecting the short-range temporal dependencies which are critical when producing a mapping from lip images to visemes and discarding local spatial information due to a global average pooling operation. A novel convolutional block called *Temporal Focal block* to describe short-range dependencies is proposed and used as a building component in a larger model and follows a simple design with a single branch of two convolutions, layer normalization [220] and rectified linear unit activation. In order to fuse features at multiple scales, the authors experiment with several branches of convolutions with different kernel sizes and shortcut connections and local self-attention is adopted to capture long-range temporal dependencies. A spatio-temporal fusion module that aims to maintain the local spatial information while simultaneously reducing the feature dimensions is also proposed. This module uses high-dimensional spatio-temporal features into low-dimensional temporal ones without discarding important local spatial information. The temporal fusion model applies a spatial pooling operation across the entire spatial dimension which extracts a small feature map from each spatial feature and then reshapes them. The features are then fed into a stack of temporal convolutions to enhance communication between time steps and to control the number of output channels. The authors employ their newly-proposed fusion module in their approach to replace the global average pooling operation that is typically used by other works in the literature. Their model uses lip image sequences as input and outputs sequences of characters. To extract visual features, a convolutional feature extractor model consisting of two 3D convolution layers and a 18-layer residual network is used. Each 3D convolution layer is followed by a max-pooling layer while some of the stride operations in the residual network are removed in order to significantly reduce the overall training time.

SpotFast networks are proposed in [221] as a novel deep learning architecture for word-level speech recognition based on a network that utilizes a temporal window as a ”spot pathway” and all frames as a ”fast pathway”, based on the *SlowFast* networks [128] that are used for video recognition with modifications for this task. Word boundary information is used by the former pathway (spot) which is a temporal window centered at the keyword-spotted frame, while context before or after the keyword is implicitly modeled by the latter (fast). Both pathways are fused via lateral connections, using a convolution fusion with additional adaptive average pooling to temporally reshape the features from all frames into a fixed-length temporal window. The convolution fusion consists of a dual two-layer 1D temporal convolution aggregating the temporal information on each pathway. A 6-layer transformer encoder is placed on top of each pathway to further learn features for classification. Lateral connections are then added to all layers of both transformer encoders from the all frame pathway to the temporal window pathway. Each transformer encoder is memory-augmented at the penultimate layer [222] to increase the capacity and stabilize training.

A two-stage multi-modality audio-visual speech recognition model is presented in [101]. In the first stage, the target voice is separated from background noises with help from the corresponding visual information of lip movements, making the model 'listen' clearly. At the second stage, the different modalities (audio and video) are combined to further improve the recognition rate of the model. The multi-modality network consists of two sub-networks: an audio enhancement network that receives image frames and audio signals as inputs and outputs the enhanced magnitude spectrograms while filtering the noisy ones and a two-stream speech recognition network. The former uses temporal convolutions [155] and an element-wise attention GRU [103], while for the latter the authors build a fully 3D CNN network [102] instead of the common 3D module and 2D residual network combination used in previous works.

Cross-modal self-supervision is proposed in [223] where the goal is to learn representative speech and speaker features in the individual modalities without having access to any form (manual or automatic) of annotated data. A novel training strategy is proposed where the objective is to learn embeddings that are discriminative for both the primary cross-modal task and for secondary uni-modal downstream tasks. A two-stream network with two sub-networks is used where the audio stream model receives mel-filterbank features of speech segments and the video stream model ingests a video depicting a cropped face. Both sub-networks follow the *VGG-M* [23] architecture with modifications depending on the task at hand. The training function optimizes cross-modality metrics as well as intra-modality class separation by encouraging the relative distance between corresponding pairs of audio and video to be closer than the non-corresponding pairs, while also penalizing the distance between same-modality inputs which are close to each other, helping the network to learn more discriminative embeddings. The method is trained on the tasks of audio-visual synchronization and cross-modal biometric matching which serve as proxies and is evaluated on the downstream tasks of visual speech recognition and speaker verification.

Multi-lingual visual speech recognition is explored in [224] where the authors note the similarities between movement patterns of the mouth in human spoken languages despite the obvious differences in their rules and grammar. A multi-lingual learning framework is proposed where phonemes related to the lip movements rather than the alphabet are introduced as modeling units that guide the learning process of different languages and then a novel model learns the rules for each language from the data. The model architecture follows an encoder-decoder design, where a spatio-temporal CNN and stacked self-attention [50] blocks encode the input sequence, while for the decoder a two-branch transformer architecture is used. Since similar phonemes lead to similar visual patterns regardless of language, the objective of the model is to learn the composition rules of each language by utilizing its bidirectional context. An extra task of predicting the language identity is introduced in the learning process which improves language-specific capabilities.

A novel network utilizing *Transformers* for word-level speech recognition is proposed by Huang et al. [225]. The model leverages a deep convolutional network with a self-attention Transformer encoder-decoder structure [50] for word-level speech recognition without relying on any audio information. The Transformer architecture is selected over RNN-based derivatives (e.g., BGRUs) due to its low simplicity which facilitates a faster training process. For feature extraction, the overall structure uses a *VGG*-16 model [60], where in order to save time from training the model from scratch, its parameters are initialized from pre-trained weights. A dimensionality reduction operation down-samples the spatial dimensions of the extracted features before feeding them to the Transformer model for training. For sequence encoding and decoding, a standard Transformer architecture with an equal number of attention heads for both components is used.

Multi-lingual speech recognition is explored in [226]. Noting the lack of sufficient training resources for languages other than a select few, the authors aim to exploit existing knowledge from a transfer learning standpoint in order to build a system that is language-independent. To that end, they generate a small-scale dataset comprised of samples from larger datasets in three different languages, which serves as a benchmark for knowledge transferring. Its vocabulary size is normalized by selecting an equal subset of words from the three source datasets that are classified as easy or hard, keeping the difficulty of the samples balanced. A ratio of $8 : 1 : 1$ is set for the training, validation and testing subsets, respectively. Two deep learning visual speech recognition architectures are evaluated: a convolutional model with dense connections [26] and an auto-encoder with soft-attention [50], both making use of a 34-layer residual network. For sequence recognition, while the first architecture uses a bidirectional RNN, the second one relies on the attention mechanism and two fully-connected layers. Two training setups are devised, where in the first one an English dataset [21] is used to pre-train networks that are subsequently fine-tuned in other language datasets ([227] and [39]) and in for the second setup the multi-lingual dataset is used instead.

Ren et al. [228] explore a cross-modal distillation scheme where the audio signal is used to pre-train a network acting as a teacher which then transfers its knowledge to a student (video-only) network in the distillation process [99], since audio recognition tends to perform better than its vision counterpart when the goal is speech recognition. Due to the inherent domain gap in the modalities of audio and video which can potentially hamper training by applying distillation naively, the authors employ a network that utilizes signals of both modalities as well as their combination to act as the teacher providing a more complete training signal for the student. Furthermore, during the training process, the teacher network is regularized with feedback from the student, guided by two pre-trained modality-specific ”tutor” networks. A curriculum learning strategy [229], where the dataset difficulty gradually ramps from easier to harder samples up as training progresses, is employed to improve convergence.

In a similar fashion, [230] combine cross-modal distillation with a novel unsupervised domain adaptation [231] method using out-of-class data. Since audio contains more relevant information about speech than video without sound, an audio-based recognition network is used as the teacher for the unlabeled adaptation data as it contains more descriptive representations and can provide a better supervisory signal during training. The adaptation data is an additional training set used to adapt a pre-trained model of a different domain to a new one, especially in cases where collecting and annotating a sufficient amount of new data is not feasible. To allow the use of unknown class data (out-of-vocabulary words), the intermediate output of the teacher network, which contains implicit representations about the data (e.g., sub-class), is used in the distillation process. Both models share a two-part structure consisting of a stack of convolution layers that encode the input data and a classifier layer to obtain the probabilities of words. A three-step training adaptation procedure is performed where the audio model is trained in advance on the task of word-level recognition using audio data. With the fixed parameters of the audio model as the teacher network, the visual model is trained with an additional distillation loss and finally, the visual model is adapted to the unknown class data.

A memory network that can augment a speech recognition model with rich representations from audio data is proposed in [232]. The purpose of the memory network is to memorize corresponding audio features and complementary speech-related information during the training stage by leveraging the audio-visual features of lip movements. A key-value formulation is used, where audio representations related to a specific video feature are extracted from the audio track and saved in the *audio-value* memory, while the location of those representations within the module is stored in the *video-key* memory. To train the model, a synchronized (identical temporal receptive fields) double-stream network is used, where a feature extractor corresponds to each modality (audio and video) and both outputs are fed to the visual-audio memory module which has a double objective to memorize as well as match the pair of inputs. The saved memory feature is then concatenated with the visual features and fused via a fully-connected layer before being fed to a sequence modeling network to make predictions. At the inference stage, the audio-only stream is not used and by using uni-modal (video-only) inputs, the corresponding audio features are recalled to enrich the visual representations by accessing the stored video-key memory.

An alternative approach is introduced in [233], where the goal is to improve the generalization of baseline word recognition models by applying principles of information theory. The authors propose a novel training-time-only temporal masking module [234] that learns a latent encoding from the original input to the target which maximizes the compression of the former and the expressiveness of the latter. The *temporal mask module* is introduced after the feature extraction stage and before sequence classification, operating on the concatenation of the feature maps from different temporal receptive fields which holds information of several temporal scales. This module acts as an information bottleneck and is trained alongside the other components of the architecture, compressing the extracted visual features by selectively filtering those of a low importance score that is learned from the data and keeping the more salient ones. At test time this module is removed, thus it does not incur an additional computational overhead to the overall model. The rest of the architecture is comprised of a spatio-temporal residual network with channel attention [69] with one of several sequence-to-sequence models including a 3-layer BGRU, a MS-TCN or a Transformer.

In [235], a novel audio-visual fusion approach is introduced. In order to effectively fuse the two different modalities, incorporating aspects of both, the authors propose a method that exploits a *Siamese neural network* [236] with shared weights along with feature masking and polynomial sampling. Two encoders are utilized and their outputs which have the same number of feature maps are fed to the Siamese network which is composed of two fully-connected layers with shared weights and enables the seamless integration of the information from the audio modality to the video by learning the correlations between the heterogenous modalities without introducing noise. Two masking matrices of equal dimensions as the feature maps are initialized with standard normal distribution values and are multiplied value-wise with the feature matrices to obtain the masked results that are then applied as the input for the subsequent polynomial sampling. As opposed to simple concatenation which would double the channel dimension, random binomial sampling of the features is performed and retains the original feature map dimension functioning as an additional regularized during training.

Pan et al. [237] propose leveraging non-labeled data of a single modality to improve performance. The authors use models that are pre-trained in uni-modal data through self-supervision. The overall architecture involves two sub-networks, one for each modality, with different front-ends relevant to the task, while the back-ends have the same design. More specifically, for the audio modality, the front-end follows [174] which is commonly used for speech recognition tasks, whereas for the video modality, a contrastive learning model [238] pre-trained on images is employed after substituting its first layer with a spatio-temporal network. Both modalities use the same architecture for their back-end components, which consists of 1D convolutional layers applied to the time dimension combined with Transformer encoder layers for temporal modeling of each single modality. A fusion module concatenates the features from each modality after normalization and a similar 1D convolution and Transformer encoder layer network is used to fuse the features together. Two decoders are trained simultaneously based on the output of this fusion module. The overall model training is conducted in a multi-stage fashion where each uni-modal front-end is pre-trained through self-supervised learning and then the audio network is trained in an audio-only setting, while the video network is trained at word-level video clips. Finally, the audio-visual model is be trained after the modality-specific models have converged.

A cross-modal framework is proposed in [239] where the authors aim to exploit the language-related information present in multi-modal data to improve audio-visual speech recognition performance. This framework leverages feature disentanglement learning strategies via a novel linguistic module that extracts and transfers knowledge across modalities through cross-modal mutual learning. Pairs of encoder-decoders are employed for each distinct modality (i.e., video and audio) and feature type (i.e., identity, linguistic) and the cross-modal linguistic module utilizes a modality-invariant code-book [240] and a speech recognizer producing the linguistic representations and speech recognition, respectively. The module is able to extract linguistic and identity information from cross-modal input data into modality-agnostic representations regardless of the source modality. Apart from speech recognition, this information can be utilized in a way that affects the audio-visual data output depending on the subject's identity as well as on the linguistic context and for this reason the proposed framework is evaluated speech recognition as well as speech synthesis.

Similarly, Akman et al. [241] propose a dilated convolutional neural network model for word recognition inspired by the Temporal Convolutional Network [155] architecture. Following design principles introduced by the TCN, the proposed CNN-derived model contains 5 residual blocks with two convolution layers and distinct dilation rates that affect the receptive field of each operation. In the first and last block of the model, a dilation rate of 1 for both convolutions is used to preserve the low- and high- level features respectively. Identity shortcuts are added at every block in order to facilitate an easier training process for deep networks. With the exception of the first block where the input is passed through a convolution layer with a dilation rate of 1 before being added to the residual pathway of the block, a simple summation is performed for all other blocks, similar to [31]. Spatial dropout is added at every block after each convolution operation to avoid over-fitting. A novel dataset is constructed from YouTube videos containing balanced samples of daily words and phrases and used to train and evaluate the proposed architecture.

Yu et al. [242] approach audio-visual recognition using Liquid State Machines (LSM) [243]. The LSM is a three-layer architecture that is well-suited to sequential problems of a spatio-temporal nature. The first layer functions as the input layer and is sparsely connected to the intermediate layer which is called the liquid layer and is the central component to the LSM. This layer acts a filter that transforms the input to non-linear patterns of higher dimension producing a response for each input that constitutes a *liquid state*. The final *readout* layer consists of *spiking neurons* [244] and transforms this response into a feature vector. A novel bi-modal spiking neural network architecture is proposed where the input is first transformed into a sequence of events in order to be processed by the model. One LSM per modality is used to extract features from the input and a soft fusion method [245] combines the information by re-weighting each modality with an attention mechanism [246]. Apart from recognition using clean data, the method's robustness is validated in experiments where extra noise is added to the raw audio data.

A novel method to improve model generalization to unseen examples is proposed in [247]. Since adapting to new speakers that are not present on the training dataset is particularly difficult for visual speech recognition models due to a sensitivity to appearance and particular movements of the lip, the method involves introducing non-zero padding values into the convolution operations of the feature extractors in existing architectures. Instead of the conventional approach which inserts zeros, the added padding represents an additional input that interacts with the kernel filters during the encoding of the visual features without having to add extra parameters or to modify the existing weights. The model can then be adapted to the new data by using the pre-trained weights and fine-tuning on a new dataset. Moreover, the learned padding values can be inserted into other architectures without the need to re-train the networks. To showcase the method, the *LRW* [21] dataset is used to create two non-overlapping splits with added speaker information.

The authors of [248] propose a customized architecture for speaker independent speech recognition applications. The proposed system follows a lightweight design inspired by 3D convolutional neural network designs used in action recognition [249], that consists of stacked spatio-temporal convolution, normalization and non-linear activation layers, where despite the convolution filters gradually increasing, the kernel sizes remain static with a stride of 1. A pooling layer is added to retain relevant information from the feature maps while reducing the spatial dimensions and the amount of computation required. In contrast to previous works that utilize recurrent or temporal networks for sequence modeling (e.g., [106, 162]) after the visual feature extraction process, the authors opt for a more compact design, adopting a simple fully-connected layer with 32 units and dropout instead. The proposed recognition system is trained and evaluated on a novel dataset of high quality recordings of speakers uttering words. After training, the model's weights are exported to a portable format for evaluation by a mobile application.

Different than previous approaches, [250] explore quantum machine learning in a privacy protection setting with applications in visual speech recognition. A multi-step pre-processing procedure involves extracting filterbank features and video grayscale data from each modality and subsequently splitting them into patches that are encoded into initial quantum states using a gate. Further transformations are performed by a privacy-enhancing randomly-generated quantum circuit that divides the input into patches to generate quantum-privatized data. A novel metric that measures inter- and intra-class similarities between data in order to evaluate the privacy-preserving capabilities of different methods is proposed. The metric is used to measure the robustness of the proposed method against two types of privacy attacks on the downstream task of visual speech recognition. Noise is added to corrupt the original audio stream, while the video stream is not degraded. An experimental evaluation showcases that some information loss is inevitable, however, the quantum-based privacy-preserving method retains relevant visual feature information for downstream tasks such as speech recognition while being more resistant to privacy attacks.

Extending their previous work ([247]), the authors explore several *prompting* strategies, which involve input-level modifications of the data, in [251] as an effective method to adapt a pre-trained model for a different task or data distribution. The first strategy consists of adding speaker-specific perturbations in the form of a matrix of same spatial dimensionality as the source frame to all frames in the sequence that is input to a convolutional neural network (CNN), in order to increase the network's adaptability to unseen speakers. The second strategy applies different padding values to the convolution operation within the CNN's intermediate layers to enable adapting networks of large architectures where input prompting might not succeed. The final strategy concatenates the encoded visual features that are produced by the CNN with a prompt of the same shape, in the temporal dimension before proceeding with the sequence model. The three strategies can be utilized in combination for visual speech recognition purposes and do not require additional layers, adaptation networks or extra fine-tuning of the source weights, as only the prompts are tuned when adapting the model to unseen speakers.

An approach using graph convolutional networks (GCN) is proposed in [252]. Different from previous works using GCNs (e.g., [95] and [163]), this method utilizes point clouds instead of standard images as inputs. Compared to using whole images in a GCN framework, point clouds avoid redundant computations related to pre-processing and inference, while offering robustness to dataset biases such as speaker appearance distribution. In addition, GCNs can benefit from point cloud data by learning correlations between points other than the mouth area. First, landmark detection is used to extract point cloud data which is subsequently normalized and aligned with a fixed reference point. Input points are selected either by representative regions of interest corresponding to different areas of the face, or by selecting a subset of the entire point cloud. The adaptive graph convolutional network [253] is used to encode the point cloud information spatially by learning a set of adjacency matrices and TCN layers with additional residual connections [254] encode the temporal evolution of each node in the graph.

A cross-modal language modeling framework for sequence- and word- level speech recognition that includes two components is proposed in [255]. The visual component comprises a deep convolutional (CNN) model for feature extraction and a single decoder architecture for generating texts without requiring an encoder. The CNN model extracts multiple lip representation sub-spaces corresponding to frame features originating from different convolution layers within. It combines a stack of residual blocks with local attention modules that re-calibrate the generated representations at each block to adjust its final output, taking into account the diverse information. Weighted averaging determines the contribution of each local attention module, inspired by [50], to the final output. For decoding, a standard *Transformer* decoder is employed [50] to generate the transcripts. The training strategy resembles multi-task learning and pre-trains the visual components separately. First the decoder is pre-trained for character generation and then it contributes in the pre-training of the visual feature extractor. After both modules have been trained, cross-modal language modeling or *cold fusion* [256] can be applied to fine-tune the model for speech recognition tasks in an end-to-end fashion.

Akin to [226], cross-language speech recognition is explored in [257]. The authors propose a method where generalized language-agnostic knowledge that is learned during training is exploited to improve performance on datasets with fewer samples, e.g., for under-represented languages. Two deep architectures utilizing a common spatio-temporal residual network extractor, with Bidirectional GRUs and Multi-Scale TCNs, following [69] and [110] respectively, are employed as baselines for their experiments. In order to facilitate cross-language knowledge transfer, an adversarial domain adaptation framework is adopted. More specifically, source and target image pairs corresponding to two different languages are used as inputs for feature extraction and the visual representations obtained by the residual network are fed to a compact TCN-derived language discriminator model. The latter aims to encourage the network to align feature-space representations corresponding to similar words from the two languages. Since adapting to an under-represented language could be hampered by over-fitting due to a smaller size dataset, several regularization techniques are applied.

An approach utilizing *Spiking Neural Networks* (SNNs) for event-based speech recognition is proposed in [258]. Event cameras record the variations in each pixel's brightness values focusing on the foreground only, while their fine-grained temporal resolution allows capturing a substantial amount of data that can contain implicit lip movements, both properties that favor these cameras for the task. The operation of SNNs resembles that of biological neuron activations [259] and the processing of spatio-temporal data of fits with the event-based cameras' method of function. In order to overcome some of the challenges involved in incorporating SNNs in a speech recognition framework that involve filtering the data due to their amount and distinguishing between visually similar words, the authors propose a novel spatio-temporal attention block to focus on features relevant to the lip movement. The block is comprised of two distinct branches, for spatial and temporal attention, where each branch undertakes a different role, the former localizes the important spatial patterns, while the latter evaluates their importance with regard to movement. The outputs of both branches are then fused to form a complete representation that combines attributes from both attention mechanisms. The SNN-derived architecture is trained with a triplet loss [260] which leverages a *triplet group* that consists of three items, an anchor, a positive and a negative item, encouraging an increased distance between positive and negative pairs, which enhances the model's robustness to visually similar samples. During training, three parallel architectures with shared weights are trained, while at inference only one is utilized. The proposed method is evaluated in a speech recognition dataset captured using an event camera [129].

Extending their previous work ([250]), the authors of [261] propose an approach that combines classical neural networks with quantum machine learning. A hybrid speech recognition network preserves user privacy by adopting the *Differentially-*

Private Stochastic Gradient Descent [262] algorithm for training, which adds Gaussian noise into the computed gradients for each mini-batch. While for feature extraction, the typical 18-layer residual network with the 3D convolution is used, a *Variational Quantum Circuit* [263] handles the temporal sequence modeling, replacing other architectures (e.g., TCN, RNN). This circuit uses *projection-valued encoding and measurement* [264] to convert the data into a quantum representation allowing for improved recognition performance and scalability to large numbers of quantum bits.

2.2.5 Works most Related to this Thesis

We dedicate this subsection to recent works in the literature that propose methods developed with goals that closely align with the objectives of this dissertation, i.e., lightweight and practical visual speech recognition of isolated words. The aim of these works can be broadly defined as improving efficiency, which can be achieved through several approaches such as reducing the computation resources required by the employed models, shrinking the network size in terms of parameters, lowering the memory access cost of the network components and generally producing architectures that achieve faster inference speeds for VSR. Lightweight models exhibit desirable properties such as improved training or inference running times and offer a broader range of applications as more devices are able to utilize them due to the reduced hardware costs. Moreover, a favorable indirect outcome of developing, training and deploying such models is the reduction of energy expenditure and carbon emissions which are topics of ever-increasing importance in a modern industrial world. By including these publications in their own subsection, we present the status of the literature on this specific domain, identifying oversights and potential areas for improvement.

One of the earlier works on this topic can be found in [265], where the authors propose taking advantage of *Hahn moments* to reduce computation costs. To that end, they insert a Hanh-based layer in the beginning of a CNN architecture to extract a mixture of Hahn moments which are subsequently used in the CNN, which also functions as a dimensionality reduction mechanism. Hahn moments are a set of orthogonal moments based on discrete Hahn polynomials defined over the image coordinate space and have the ability to hold and extract salient information from the image without redundant computations as they do not require any form numerical approximation. An additional benefit lies in their flexibility to capture global or local characteristics of the image, improving the quality of the extracted features. A simple CNN-only architecture is designed, without using any type of sequential network keeping the overall complexity and computation overhead low. Since the first layer computes the Hahn moments, the input to the CNN is a matrix of these moments instead of an image. The architecture is evaluated in word and digit recognition.

In [266], the proposed models are designed with mobile devices in mind. The authors first use *depth-wise separable* [61] convolutions in the 3D convolution module to reduce the parameter count and computational complexity without significantly affecting accuracy. In order to reduce memory access costs and keep model sizes low, the authors also design a mobile-friendly convolutional module using the low-parameter depth-wise convolution with residual connections and propose an architecture using stacks of this module that can be scaled with respect to its depth according to user demand. For temporal modeling, the authors note that the components based on recurrent architectures which are used by previous works (e.g., BLSTM [30], or BGRU [97]) are rather memory-intensive and performance unfriendly. For these reasons, they forego using these components in their method, opting for a simpler temporal convolution-based architecture instead.

A web application for automatic visual speech recognition for the Japanese language at the word-level is proposed in [267]. The application is tailored to smart devices, i.e., computers, portable tablets and mobile phones and it circumvents the need for powerful hardware since it offloads the processing to the server. Affine transformations for scale and rotation are applied to normalize the facial feature points reducing differences in the distance between the camera and face, the camera shake and the head movement, based on the detected facial feature points of both eyes. For feature extraction, the authors follow an approach that uses only the motion-based features obtained by subtraction between the current frame and the next one at each of 20 feature points near the lip area after normalization, as that approach is fast to calculate and does not compromise accuracy. For the recognition process, a GRU [140] is employed since in their experiments it outperforms the LSTM.

Several techniques for practical speech recognition models are proposed by Ma et al. [268]. In order to significantly reduce the computation costs of their previous work(s) ([110, 84]), the authors propose replacing the standard convolutions in the temporal convolution layer of the TCN backbone with *depth-wise separable* [61] convolutions that are more efficient, producing a lightweight temporal convolution architecture called *Depthwise Separable Temporal Convolutional Network* (DS-TCN). In addition, they replace the standard residual network used for feature extraction with the much more lightweight *ShuffleNetV2* architecture [185] which reduces the overall computation costs by a significant factor. A variation of *knowledge distillation* [99] where a larger model, called the *teacher model* adds an additional supervisory signal besides the dataset in the process of training a smaller model, called the *student model* is employed for model training. In this variation [182], both networks have the same architecture and in the next training iteration, the student model becomes the teacher and is used to train a new network (the new student).

Voutos et al. [269] propose a uni-modal speech recognition network for medical patients deployed in a mobile application. Personalized recordings are first collected from a patient in a controlled environment, then annotated and adjusted to remove imperfections such as noise. The pronounced words were selected so as to cover all phonemes in the Greek language. A data pre-processing step extracts the relevant video frames and the corresponding isolated area of the lip from each frame using a pre-defined mask. The architecture involves a sequence-to-sequence recurrent network that is trained on a subset of the extracted frames for the task of word prediction and evaluated on a disjoint testing set. The overall model is comprised of three LSTM layers starting with 256 hidden units for the first layer and halved for each subsequent one and two fully-connected layers with a final Softmax activation function to obtain the word class probabilities. Post-training, the architecture was exported to a mobile format for integration in practical applications.

In a similar fashion to [266], Wisesa et al. [270] develop a novel model for visual speech recognition with low complexity for deployment in limited resource devices. Since the goal is application in low complexity or constrained hardware, the authors focus on developing lightweight models with minimal computation costs. Drawing inspiration from [69], two end-to-end variants are designed, both using a *ShuffleNetV2* [185] network without the last convolution block to lower complexity and while the first variant uses a Bidirectional Gated Recurrent Unit as the sequence encoder, the second replaces it with a Transformer module that performs slightly better at a smaller parameter size. The models are trained following the settings from [69] and the standard pre-processing steps of previous works, which includes conversion to grayscale to reduce complexity as well as mouth area region of interest (ROI) cropping to lower the input spatial dimensions while simultaneously discarding redundant information from the frames of the sequence.

Bulzomi et al. [271] propose an efficient end-to-end neuromorphic model for word-level speech recognition. Different from previous approaches using convolutional deep learning architectures, their method utilizes *Spiking Neural Networks* (SNNs) [272] due to their energy efficiency and low latency, both favorable factors for real-time applications. SNNs simulate biological neuron activations by emitting voltage ”spikes” and operate on data captured from event-based cameras that record events such as pixel-level changes in brightness which may occur at irregular time intervals, allowing for lower power consumption compared to traditional video cameras which record whole frames at a fixed rate. In practice, due to the non-differentiable spiking operation of the SNN, it is approximated through surrogate activation functions that allow training of the network during gradient back-propagation. Their model follows the design of [129] with a residual network backbone which is subsequently modified with a spiking architecture [273], while the Gated Recurrent Unit (GRU) that extracts temporal information in the original architecture is removed. Several surrogate activation functions are tested as well as layer substitutions for the removed GRU and all models are evaluated on the *DVS-Lip* [129] dataset which contains event-based speech data.

An efficient model for deployment in devices with limited storage capacity and computational resources was developed in [274]. More specifically, the authors develop an efficient channel attention module which is then inserted in several bottleneck blocks of a lightweight network baseline [275] achieving parameter reduction without compromising accuracy. Following the design of [276], the module consists of a pooling layer, 1D convolution operation with a channel-adaptive kernel size and a non-linear activation function and the generated attention weights are multiplied with the input feature map. A Gated Recurrent Unit (GRU) is selected as the sequence modeling network over the LSTM and BGRU variants for its comparable performance and simpler structure. For the purposes of evaluating the newly-proposed model, a dataset containing digit utterances from 10 speakers was created. In order to increase the robustness of the model and to simulate applications in real-world conditions, speaker recordings were obtained by cameras set up in multiple angles. The proposed efficient network is evaluated and compared to other lightweight architectures ([275, 81]) on the task of single word speech recognition.

More recently, a large study of deep learning architectures was conducted [277] where the authors explored a wide range of models across several representative datasets for word-level visual speech recognition. The authors evaluate the performance of various convolutional and transformer-based networks used in both components of the system, i.e., feature extraction and sequence modeling. For the former, the authors experiment with the typical 3D convolution sub-network commonly found in other speech recognition works (e.g., [69, 162, 268]) combined with the standard [31] and wide [278] residual networks, as well as *EfficientNet* [279], while for the latter, MS-TCN [110] and Transformers [50] are used. Besides the convolutional architectures, Vision [133] and Video Vision Transformers [280] are also explored. All models are trained with the same data augmentation settings that include *RandAugment* [281] which selects a data augmentation method from a pool of possible options randomly and *mixup* [66], which generates random image pairs from the training data with a fixed weight that controls the amount of mixing.

The authors of [282] extend their previous work [131] by proposing a more lightweight approach. Following the same transformer-based architecture, a more compact model is obtained after a series of steps that are applied to the original model. First, weight transformation [283] and parameter sharing are introduced to the transformer components. More specifically, linear layers are inserted in the Multi-Head Self-Attention block, while a depth-wise convolution is added in a Multi-Layer Perceptron. In addition, parameter sharing [283] is applied across the transformer blocks excluding the newly-added and normalization layers. Next, training with a distillation objective is performed in order to transfer knowledge from the larger pre-trained models to the final, smaller ones. A triplet loss utilizing the sum of three distinct distillation methods is used, taking into account predictions as well as the transformer self-attention and hidden states. The first method leverages the typical distillation loss [99], while the others leverage the various components to improve performance, since both networks utilize transformer blocks [283]. Furthermore, data augmentation and other techniques are applied during training to enhance accuracy. The compact models are evaluated in English as well as Chinese word recognition tasks.

[284] propose a lightweight visual feature extraction network utilizing the *Swin Transformer* [211] rather than a residual network. A Swin Transformer block applies self-attention in local windows that are shifted across the image pixels to facilitate

information exchange. The authors adopt the hierarchical model of [211] and perform a few modifications to adapt it to the task of speech recognition. To capture more global information regarding the lip movements, a large initial patch size is used. Also, to avoid any potential drops in accuracy, the last stage is removed and replaced by a 1D convolutional attention module which contains feed-forward layers, a multi-head self-attention mechanism and a 1D convolution followed by a normalization layer in order to enhance feature expression along the temporal axis. The modified four-stage architecture is employed as a visual feature extraction unit and combined with sequential models and the overall model is evaluated on two datasets for word-level VSR and one dataset for sentence-level VSR. In the first two cases, for processing of the extracted features, the TCN with dense connections [148] and a bi-directional GRU is used, while in the last case, the authors employ a *Conformer* encoder [285]. A variant intended for real-world applications where future frames are not available for computations is created by removing the self-attention layer in the last stage, yielding a slightly more lightweight model with a small degradation in performance.

2.3 Datasets

In this Section, we overview datasets published exclusively for the task of single word visual speech recognition over the years. This work focuses on the English language, mainly due to the amount of the available data, since in order to train powerful deep models, a significant amount of data is required, especially as the size of the model increases. An additional factor is the popularity of the language, which is worldwide and as a result most published methods for word-level VSR use English datasets, allowing for a fair comparison. As the focus of this dissertation is recognition of spoken English words, we devote the first subsection to datasets specifically in English. For the sake of completeness, datasets intended solely for word recognition purposes for other languages are also included and grouped in a separate subsection. As a special case, multi-lingual datasets (i.e., containing more than one language) are included in a separate subsection.

2.3.1 Datasets for the English Language

Lip Reading in the Wild (LRW) dataset

Prior to the release of the *Lip Reading in the Wild* (LRW) corpus [21], the available datasets were rather limited in terms of samples and vocabulary. As of this writing, LRW remains the largest dataset for single word speech recognition of the English language. This dataset contains a rich vocabulary of more than 500 distinct words, spoken by more than $1,000$ speakers. Variants (e.g., singular and plural) of the same word are also included in this vocabulary adding an additional layer of difficulty to the dataset since they lead to ambiguities in predictions for some words. This is due to the fact that compared to audio, the video information does not assist in distinguishing the word variant and mis-classifications can occur as one word can be mistaken for another.

Owing to the fact that the video sequences were recorded from television shows, a significant amount of visual variation in both the speakers depicted as well as the scene backgrounds is noticeable. For the same reason, the video quality is high and the lighting conditions are adequate without dark spots or visual occlusion of the speakers' faces or mouth area. The scene backgrounds vary widely depending on the program that was recorded and there are multiple head pose angles for the different speakers. Having several views of the head pose and angle leads to an increased training robustness as opposed to datasets that depict the person from a single angle (typically from the front). Examples of images from the dataset are shown in Figure 2.2.

The LRW dataset is split into three smaller, unequal subsets: the train set, the validation set and the test set, shown in Table 2.1. Each particular subset contains an amount of non-overlapping video segments and each segment depicts a single speaker for a duration of 29 frames at a fixed frame rate of 25FPS featuring a spatial resolution of 256×256. For each video, one word utterance occurs in the middle of the sequence. The majority of the samples in the dataset are used for training purposes with the training subset containing a total of 488.766 samples numbering between 800 to 1.000 sequences for each spoken word. The remaining two subsets, used for validation and testing respectively, contain equally 25.000 samples each, with 40 to 50 sequences for every word. All video segments (train, validation and testing) amount to a total length of 173 hours, of which $157,5$ correspond to the training set.

Figure 2.2: Sample images from the LRW dataset.

All works presented in later chapters of this dissertation involve this dataset, using its aforementioned splits for training, validation and testing.

DVS-Lip

DVS-Lip was introduced in [129]. To the best of our knowledge, this is the only dataset for word-level recognition using event-based data. An event camera that simultaneously outputs an event stream alongside intensity images is used to collect the data. The vocabulary is sourced from *LRW* [21] by selecting the 25 most frequently mis-classified word pairs (e.g., `million–billion`) as well as a random selection of 50 other words, resulting in a combined total of 100 different words. The videos feature 40 distinct speakers in an indoor scene reading sequences of words that are then subsequently split accordingly to the audio data so as to produce samples containing

Table 2.1: Dataset split details for the LRW dataset.

Split	**Samples**	**Sequences/word**	**Hours**
Train	488.766	$800 - 1000$	$157,50$
Validation	25.000	50	$8,05$
Test	25.000	50	$8,05$

only a single word. The final dataset contains a total of 19.871 valid word samples, split into two non-overlapping splits with 14.896 samples from 30 speakers for training and 4.975 from the remaining 10 for testing, respectively. The dataset can also be divided into two parts depending on the word source, where the former contains 7.441 training and 2.493 testing samples of mis-classified pairs and the second contains 7.455 training and 2.493 testing samples of the random words.

AusTalk

The *AusTalk* corpus [286] is a large audio-visual dataset covering several recognition tasks. Its content is diverse, comprising words, digits as well as sentences pronounced by Australian speakers. Recordings were performed in several locations across Australia, featuring a wide variety of up to 1.000 geographically diverse speakers, that are not necessarily native. The total length of the collected audio-visual data is 3.000 hours. As annotation is an expensive and time-consuming process, considering the size of the dataset, only a subset of the total data was annotated manually.

MODALITY

The *MODALITY* database was introduced by Czyzewski et al. [287] to assist with the development of audio-visual speech recognition systems. Recordings of 35 speakers uttering sentences or isolated words are included. In order to evaluate visual speech recognition performance under noisy conditions, for some recordings, acoustic background noise, such as traffic sound was simulated. The videos were captured at 100 FPS frame rate with a resolution of 1920×1080 using a setup with two stereo *RGB* (color) cameras. Each speaker was recorded using the RGB cameras from a slight angle providing multi-view data that can also be used to retrieve 3D depth data. Furthermore, for some speakers an additional depth camera was used, providing 3D depth data that can also be utilized in other tasks. The language selection simulates voice control scenarios where a command would be spoken to a device, thus it includes 182 unique words. The vocabulary consists of numbers, names of months and days as well as a set of verbs and nouns mostly related to controlling computer devices. In order to facilitate word- as well as sentence- level recognition, apart from the isolated words, a set of 42 sentences containing all words in the vocabulary was also uttered by the speakers. Overall, 31 hours of recordings were obtained and for every utterance manual annotation was applied.

MIRACL-VC

The *MIRACL-VC* [288] corpus contains depth as well as color images that can be used for other tasks apart from speech recognition, such as face detection, biometric estimation, etc. Another aspect of this corpus is that its vocabulary covers two tasks: word as well as phrase recognition. For each task of the dataset, 15 speakers utter a set of either 10 words or phrases for 10 iterations, resulting in total 3.000 word and phrase data, that is split equally. In addition, 2D images and depth maps were acquired with a Kinect sensor at a frame rate of 15. As it contains both word and phrase data, MIRACL-VC has been used to evaluate methods focusing on either word- or sentence-level speech recognition.

LIPSFUS

Different than other datasets, *LIPSFUS* [289] is an audio-visual neuromorphic corpus collected using a set of *Address-Event-Representation* sensors and tools. The data is collected from *Neuromorphic Sensors* [290, 291] that generate spike information encoded and maintain the synchronization of the different modalities. Recordings take place in a quiet, noise-isolated environment and a noisy one, where background noise is present with similar lighting conditions for both environments. The speakers consist of 22 persons of 5 different nationalities, in the ages between 6 and 61 years. As for the vocabulary of the corpus, it includes spoken digits, robotic commands, as well as words sourced from online challenges and other datasets ([21]) and each person pronounces each word in an isolated way (i.e., not as part of a sentence that is later cropped).

LIPAR

The *LIPAR* corpus was created to evaluate the proposed system of [248]. The authors used the MIRACL-VC [288] dataset as a reference, following its data collection principles, however, the repetitions per word were reduced and depth images were omitted (not collected). High definition recordings depicting the frontal view of 35 speakers each uttering 10 words were recorded in a controlled environment with adequate lighting conditions. Depending on the speaker's talking speed, each recorded sequence spans $50 - 69$ frames corresponding to an average duration of $1,2$ seconds. The videos were split to frames and the regions of interest covering the lip area are cropped according to landmarks relevant to the lips that are automatically calculated after a face detection step. The cropped images are then converted into grayscale for reducing the amount of required parameters by the model and normalized using the global contrast normalization technique. Finally, additional frames are inserted into sequences to match a set length.

CRSS-4ENGLISH-14

The *CRSS-4ENGLISH-14* audio-visual corpus was introduced in [292]. Recordings were conducted in a controlled environment (sound booth) with sufficient illumination conditions using several microphones and cameras. Background uniformity was achieved with a green screen that was placed behind each speaker. The corpus contains several modes of speech, including continuous sentences, short phrases, numbers, as well as single words. An additional, smaller, audio-only subset was collected where an audio device was used for background noise playback during each recording, simulating real-world conditions such as crowded spaces. The amount of speakers is 442 with several English accents and around 30 minutes of data correspond to each speaker. Transcription was performed manually. A subset of the corpus containing recordings of 105 speakers was used to evaluate the methods proposed in [292].

Other

A novel dataset for the purpose of evaluating a multi-modal recognition system in a virtual aquarium environment was constructed in [134]. It has a vocabulary of 54 words that includes 20 control commands, 15 words related to aquatic organisms, 11 numbers from zero to ten and 8 words conveying an emotional response. The data was gathered from 40 participants with an average age of 29 years using a head-mounted camera. Each word utterance was repeated 100 times in a sequence and the final video was split into frames corresponding to each word. A total of 216.000 video clips with audio were collected at a frame rate 30 with a resolution of 640 × 480. A ratio of $7:3$ was used to divide the dataset into training and validation sets and the samples for each set were selected manually.

In order to evaluate their proposed silent speech interaction framework, the authors of [141] introduced an in-the-wild dataset that simulates interactions between a mobile phone and a human user. The vocabulary covers 25 voice commands as well as casual expressions in English ranging from simple words to small phrases. Several recordings were performed with mobile phone cameras in various environments such as indoor and outdoor featuring a wide range of lighting conditions, scene backgrounds as well as speaker posture and orientation relative to the camera. A difference with other datasets is that some samples were recorded while the speaker was walking, resembling a more realistic scenario. The videos were collected from 11 speakers over 7 recording sessions so as to include the aforementioned conditions and each utterance was repeated 5 times. In total, 9.625 samples at 30 FPS frame rate with a resolution of 1920 × 1080 comprise the dataset.

[78] proposed a small-scale dataset containing a vocabulary of 9 randomly-selected words to evaluate their method in three modality settings that include the isolated modalities as well as the combination of both. The dataset was recorded in a controlled environment using a video recorder with a resolution of 1920 × 1080. The videos depict variable light conditions and average a duration of $1,6$ seconds per utterance at a frame rate of 60 FPS, corresponding to $80-100$ frames per video. Overall, 80 sequences were captured from 16 different speakers between the ages of 18 and 30 and each utterance was repeated 5 times. A pre-processing step temporally cropped the videos to 1 second duration and halved the frame rate (to 30). The corpus was partitioned in training and validation sets according to a $3:1$ ratio and the validation set was used to evaluate the proposed method.

2.3.2 Datasets in Other Languages

Greek

Kastaniotis et al. [40] published a dataset containing image sequences of Greek words in an unconstrained driving scenario in order to facilitate a practical application of speech recognition in a real-world task. The sequences are recorded using different mobile phones mounted inside several cars and depicts either the driver or the passenger over multiple views and lighting conditions, which represents a possible use-case by an automatic system. The vocabulary size is 50 words related to everyday driving, such as commands or locations, uttered by 10 persons and each word is repeated 5 times by each speaker, resulting in an average sequence length of 44 frames at a 25 frame-per-second rate. This dataset, to our knowledge, is one of the first publicly available datasets for word-level speech recognition in the Greek language targeting a real-world application.

Another word-level dataset in Greek with a practical application is introduced in [157] in the context of the medical domain. In this dataset, the frontal cameras of mobile phones are used to record sequences of patients uttering a pre-defined set of words without a particular order. No occlusions of the speaking face are present and other constraints, such as lighting or pose are not enforced in order to keep the dataset naturally varied (i.e., in-the-wild), simulating real conditions. Adding to this, the spatial video resolution varies depending on the phone camera characteristics and the frame rate stands between 25 and 29. The vocabulary size is 10, since the selected words are related to the medical domain, with $3 - 10$ repetitions per word and the speakers are also 10.

More recently, the *MobLip* [79] dataset for visual speech recognition in the Greek language was released. A unique one-minute utterance containing the most prevalent words in the Greek language was spoken by 30 speakers and recorded at 30 frames-per-second. The annotations were created manually by linguistic experts that highlighted the starting and ending times of each word within the sequence. These boundaries were then converted to frame indices and as certain words (e.g., pronouns) corresponded to fewer frames than others, the sampling rate was modified accordingly. 55.275 images were gathered, encompassing a total of 3.685 words. Since each uttered sequence contains more than one instances of certain words, the corresponding frames are used as a validation subset consisting of 4.425 samples. The

MobLip dataset was used to evaluate a wide array of deep convolutional architectures for speech recognition.

Chinese (Mandarin)

The *LRW-1000* [39] is a large-scale benchmark dataset for isolated word speech recognition in-the-wild scenarios containing 1.000 classes, where each class corresponds to the syllables of a Mandarin word composed of one or several Chinese characters. The dataset takes into consideration the practical applications of this task in real-world scenarios and includes a diversity of speech modes and visual conditions, including resolution variations, changes in lighting and individual characteristics related to pose, age, gender and appearance, since the samples are collected from more than 2.000 distinct speakers. Similar to *LRW* [21], the samples are collected from TV programs, featuring a wide variety of scenery, however, in contrast to *LRW*, the video resolution is not fixed, providing more natural conditions. After data curation, the total number of samples is 718.018, with over 1 million Chinese character instances including 286 Chinese syllables. The above factors contribute to the challenging nature of the dataset as it corresponds to challenges encountered in practical conditions.

In [176], the *MCLR-100* dataset for Mandarin Chinese was introduced. This corpus contains a vocabulary of 100 Chinese words and is developed with mobile devices in mind. The videos are captured by a head-mounted camera from 50 different speakers spanning a wide age range 20 to 60. The spoken words are instructions and each is uttered within a 2 second time span. The resulting video length is 60 frames and the lip area is localized in a 480×640 pixel region. In total, 50.000 RGB audio-visual samples are collected and each class contains 500 examples.

[184] released the *Databox* mandarin-only word dataset for the study of visual speech recognition of the Chinese language. The authors avoid choosing mandarin characters with very similar lip movements and collect simple utterances from everyday scenarios. The words are uttered by 80 volunteers and consist of a limited vocabulary of 20 simple words. Each volunteer repeats the words ten times, producing an overall of 16.000 video clips.

Chinese (Cantonese)

The *CLRW* [167] dataset is a corpus containing words in Cantonese, which, although is a Chinese dialect spoken primarily in south-eastern China, differs grammatically and phonetically from Mandarin. Data is collected from various video sources such as websites and TV, at 25 FPS with several resolution scales and subsequently filtered, keeping only single-speaker clips. Audio is synchronized with the video using a deep network [293] and annotated using an audio transcription service before being manually verified for correctness. After these steps, landmarks of the face are extracted and are used to normalize that area before obtaining the regions of the mouth. Due to the video source diversity, the dataset covers a wide gamut of speakers, pose, backgrounds and image conditions at about 65 hours of data. The 800 most frequent word classes are kept, averaging 500 samples each.

German

The *DLIP* word-level dataset in German was proposed in [34]. 10 utterances were repeated by 15 different speakers in a controlled environment without background noise. The word selection was influenced by the popularity of each word and covers nouns, verbs and adjectives. In total, 1.800 short-length videos ($0,56 - 2,37$ seconds) were recorded at a frame rate of 30 FPS with a resolution of 640×480. A pre-processing step includes detection of the face and the mouth area followed by a cropping operation and a conversion to grayscale. From the resulting data, two subsets for seen and unseen speakers were created. In the former subset, every speaker appears in all splits (training, validation and testing), while in the latter, all videos from 11 speakers are used for training and the remaining videos (corresponding to 4 speakers) are chosen for validation and testing, evenly split by speaker. In order to increase the size of the dataset, augmentation methods such as horizontal flipping, blurring, equalizing and adding noise were applied to every frame in each sequence.

More recently, the *GLips* (German Lips) dataset was presented by Schwiebert et al. [294] as a large-scale corpus for the German language. Over 1.000 videos are collected from an online source containing clips of parliamentary sessions along with their respective subtitle files. The data is pre-processed with a two-step pipeline, where the first step involves extracting the separate modalities of the clips (video and audio) with the appropriate transcription, as well as manually inspecting the text files and removing additional words not present in the video. A phonological transcript is created and used to synchronize the audio file with its corresponding text transcription with a web service. The aligned audio file is then used to guide the extraction of the related video from the original sequence. Face detection is applied only to a segment of the entire frame as the camera capturing the speaker is static and centered. Although the audio and video files are synchronized, they are stored in separate files to facilitate several applications of speech recognition (e.g., video-only). The total number of videos is 250.000 depicting approximately 100 distinct speakers with a vocabulary of 500 words and 500 instances of each. The format of the dataset follows *LRW*, i.e., a video resolution of 256 × 256 at 25 FPS, allowing for seamless transfer learning applications.

Russian

LRWR was released by Egorov et al. [114] for in-the-wild word-level speech recognition for the Russian language. Samples were collected from various Russian-speaking YouTube channels boasting a wide selection of speakers with different characteristics, appearance and speaking speeds covering multiple topics. Moreover, various environments with diverse conditions and camera angles are depicted. A filtering process discards videos where more than one speaker appears on the frame and a 1.500-word vocabulary of the most frequent words is prepared according to an automated transcription of 50 hours of collected samples. After the filtering process, face detection is performed and extracted landmarks normalize the video before cropping the lip region. An additional refinement step balances the resulting dataset by excluding multiples of the same word that appear in the transcription. Overall, the final dataset contains 235 words from 135 speakers with over 117.500 samples and 500 examples per word, which are split in 450 for training and 50 for testing purposes.

Czech

The *TULAVD* dataset [215] is a moderately-sized Czech language dataset featuring data from 54 different speakers, covering an age range from 20 to 70. The dataset covers the word- as well as sentence- level speech recognition tasks with 50 isolated words and 100 sentences corresponding to each task, respectively. The phonetic balance of the selected samples was taken into account and the sentences were divided into two equal groups (50 each) where the first group contains common sentences (spoken by all speakers), while the second group contains speaker-specific ones. The recordings were performed in an office environment, so lighting conditions are adequate. In contrast to other datasets, TULAVD includes RGB-D data from a depth sensor (Microsoft Kinect), which is fully synchronized with the video stream.

Bengali

The *BenAV* [295] dataset is, to our knowledge, the first audio-visual dataset with sequences of words in the Bengali language. It is collected via a multi-step procedure comprised of six steps. Random but common short everyday words described in a 50-word vocabulary of $3-4$-Bengali-character words are selected and them grouped into five 10-word batches and assigned to 5 individuals for recording. The batches are distributed in a balanced manner among speakers aged between 21 and 35 that utter their assigned words 40 times in a home environment. Then, videos with insufficient lighting, resolution, occlusions or unsteady recordings are discarded and the remaining videos are processed into a fixed resolution, bit-rate and frame rate. Segments are cropped from the processed videos and finally, after visual inspection, the cleanest examples comprise the dataset, which spans $7,3$ hours, containing 128 speakers with $350-600$ utterances per word for a total of 26.300.

Another dataset for the Bengali language was published in [296]. Similar to *BenAV*, the dataset is aimed at the task of word-level recognition. Its vocabulary numbers 12 words from three distinct categories: words that are commonly-spoken, words that have phonetically similar beginning and ending, and words that are rarely-spoken. 4.800 videos were captured using web cameras, with lengths varying from $1,6$ to $3,5$ seconds. A three-stage pre-processing method is used to produce the final dataset that includes video conversion to images, cropping of the mouth area to reduce the dimensions and keep sizes low and frame-rate normalization. The resulting dataset contains about 76.800 images.

Turkish

A dataset for Turkish was published in [59]. Similar to works for other languages, the word selection involves short videos of commonly-spoken words. The source videos are captured by a mobile phone camera in a controlled environment, meaning that all clips feature the same ambient and lighting characteristics. To correct deviations in the camera angle during recording, after cropping the mouth area from the obtained frames the cropped images were rotated accordingly, creating additional samples that enrich the dataset with a form of augmentation for added robustness. The vocabulary numbers 111 words spoken 15 times each by 24 different speakers and the video resolution is set to a fixed 1920×1080 at a frame rate of 30. For the training split, videos from 18 speakers are selected and the remaining are used for testing.

For the purposes of evaluating a novel deep convolutional neural network architecture for the task of visual speech recognition in Turkish, [241] collected a corpus containing words as well as phrases. Three classes are used for each category, covering expressions used in daily communication. Short clips containing the target words or phrases in the vocabulary were cropped by videos from the YouTube platform and the corresponding frames where the target occurs were extracted. The data is collected in such a way as to keep the data distribution of the classes balanced while depicting various speaker appearances. Pre-processing involves in multiple steps, starting with conversion to grayscale, followed by face detection and lip area isolation using landmark points and concludes with scaling to an appropriate size that represents enough visual information to keep computation costs manageable without compromising accuracy. The total number of samples is 1.390, split into subsets for training, validation and testing with a $4.67 : 1 : 1$ ratio.

More recently, Berkol et al. [297] introduced another dataset for the Turkish language, sourced from YouTube videos. Compared to [59], this dataset offers a large variety of backgrounds and speakers, since the videos are obtained from several YouTube channels with different topics. The vocabulary contains common words and phrases, each appearing in over 200 instances from several speakers with distinct characteristics related to age and appearance. Initially, 2.335 instances of videos were captured using screen recording software and split into frames. The resulting frames were manually inspected and in cases where more than one face appeared on the video it was cropped out, leaving only the speaking person while trying to preserve as much of the background as possible.

Arabic

The *AVAS* dataset [298] is, to our knowledge, the first published audio-visual dataset for speech recognition applications in Arabic. The dataset contains a total of 36 words and 13 phrases that are commonly-spoken and cover all phonemes of the language. Videos as well as static images of 50 speakers aged $18 - 60$ were captured using multiple fixed angles, resulting in various poses of the speakers' heads, as well as complex backgrounds and illumination variations. The recordings were completed in two sessions, allowing for changes in speaker appearance and illumination conditions via spotlights. In addition to the visual data, which sports a 640×480 spatial resolution at 30 FPS, corresponding audio files are also provided separately and digitally enhanced.

Another dataset for Arabic called *AVSD* was released in [299]. It contains 1.100 videos with a vocabulary of 10 isolated words used for daily communication. The data was recorded in an indoor environment using mobile phone cameras and the speaker selection includes males and females aged between 11 and 24. Each word utterance was repeated 5 times by every speaker in order to obtain different variations of each word. The videos were recorded in a 1920×1080 resolution at a fixed frame rate of 30 and depict each speaker from the frontal view. Post-recording, the obtained data was inspected and imperfections were manually rectified. Ground truth annotation and cropping of the square mouth region was also manually performed, resulting in a final resolution of 32×32.

[58] introduced a word-level dataset of their own for the Arabic language. 1.051 videos without sound are collected from 73 native Arabic speakers aged $18 - 21$, uttering common Arabic words. Each word of the 10-word vocabulary was uttered by each speaker at least once, considering the differences of each person regarding appearance characteristics, talking speed, lip deformations and cadence. In addition, a high variability in scene backgrounds, lighting conditions, camera distance and hardware properties are present in the videos, improving the generalization of the dataset. While the frame rate of every video is fixed at 30, there is no fixed resolution for all videos.

The *AQAND* [139] corpus was introduced to assist in Arabic language research applications. It consists of around 16 hours of RGB videos in 1920×1080 resolution recorded at a frame rate of 30, numbering 10.490 samples. Videos of 22 speakers whose ages ranged from 20 to 59 were collected in an indoor environment using digital cameras from three different angles, providing multiple views of the speakers. The vocabulary contains 10 isolated words, as well as 43 letters including single and disjoint ones and each utterance is thrice repeated. Variations between the speakers as well as the utterances are also present, reducing the overall bias of the dataset. Post-recording verification was performed manually to ensure the target word or letter is present, while the video length varies between 2 to 10 seconds, standardized to a precise number of frames equal to 60, 80, or 300, depending on its length. The final dataset contains videos of the cropped lip regions, which are extracted after being localized through a sequence of mouth-face-lip detection steps.

Another dataset for the Arabic language was published in [300]. The vocabulary contains 20 isolated words (digits, weekdays and other common words) in modern classical Arabic, which is used daily. 40 participants were sat frontally across a camera and uttered 20 words each and all recordings were made in the same laboratory environment using a static camera. One video per speaker at a source resolution of 1920×1080 with a frame rate of 25 per second was collected and subsequently split into sub-segments with a duration of one second per word. The produced dataset was used to evaluate a method for visual speech recognition as well as another for viseme prediction.

Daou et al. introduced a large-scale in-the-wild corpus for Arabic in [206]. Several stages were involved in the collection and preparation of the dataset, beginning with gathering a diverse set of videos of Arabic speakers from the YouTube platform. Then, a HOG-based detection algorithm identified changes in scene and used as a guide to split the videos. Face detection and tracking was applied to each frame and those containing multiple speakers were filtered. After a manual inspection for further data cleaning, the Vosk[2] speech recognition system was employed to create annotations. The videos were divided into short clips, each containing a single word utterance and resized to a size of 256×256 followed by cropping of the speaker's face. The resulting dataset contains a vocabulary of 100 words from a total of 36 distinct speakers, with different articulation speeds, tonalities, face pose variation and backgrounds. 200 repetitions of each word are present, distributed to 20.000 videos, each with a $1,2$ second duration and 25 frame-per-second rate. A split of $80-10-10$ is set for training, validation and testing.

[2] https://alphacephei.com/vosk/

Persian

A dataset for recognizing Persian words was proposed in [301]. The authors used a streaming website of Persian videos to collect samples depicting various lighting conditions and speaker poses from several sources such as TV shows, movies and interviews. 205 hours of videos are gathered using a multi-step process regarding video source selection, where interviews are preferred since they often clearly depict the person speaking. After face detection and tracking have been applied, the audio stream is used to determine the active speakers and to filter videos where voices of multiple speakers overlap. Then, an automated speech recognizer creates an approximation of the transcription and the 500 most frequent words are used to split the videos, preserving the appropriate clips and maintaining a varied vocabulary. Finally, face identities are automatically extracted and manually inspected to ensure that the dataset is speaker-independent. The resulting dataset contains 30 hours of data, at a frames-per-second rate of 25, with a spatial resolution of 224 × 224. The overall amount of speakers is 1.800 with 244.000 videos of which 233.000 are used for training and validation and 11.000 for testing.

Malayalam

A medically-oriented audio-visual dataset of words in the Malayalam language, which is spoken in southern India, is published in [55]. The samples are collected from 2 speakers (male and female) each uttering a word related to the medical domain (e.g., `fever, allergy`) for 100 repetitions resulting in a total of 2.000 videos. Emphasis is given in the facial characteristics of the speaker during the word utterance as the face can convey additional information about the speech and can assist in linguistic analysis applications. The video resolution is fixed at 1280 × 720 with a frame rate of 29.9 FPS.

Romanian

[227] presented the *LRRo* corpus for the Romanian language, with two distinct subsets, *Wild LRRo*, an in-the-wild variant designed for more practical applications and *Lab LRRo*, recorded in a lab environment with more accurate data. For the former, raw videos were downloaded from YouTube and depict recordings of TV shows using natural speech, while for the latter, the data was captured in a controlled environment.

The video segments were split into images and the annotation process was completed manually as well as with the assistance of automatic tools for data processing and filtering, ensuring that a balanced selection of speakers of multiple appearances is present. For the *Wild LRRo* subset, due to the diversity of the data sources, large variations of lighting conditions and pose occur while the vocabulary numbers 21 words uttered from 35 different speakers, for a total of 1.100 instances over 21 hours of source data. In contrast, for the Lab LRRo subset, only pose variations of the speaker are present, with a larger vocabulary of 48, 19 speakers and 6.400 instances.

Tibetan

A speech recognition corpus called *TLRW-50* for the Tibetan language is proposed in [302]. It features frontal views of 20 different speakers with a word vocabulary of 50 classes, selected from the most commonly-spoken words. Recordings were performed by a mobile phone in a controlled environment with adequate illumination and each word pronunciation was repeated three times. The isolated words were segmented from the video stream and each individual sample was saved for further processing, resulting in 6.000 video samples. A final 100 × 50-pixel image sequence is generated after detecting the speaker's face and cropping the mouth area. Data augmentation techniques such as mirroring, random rotation, cropping and noise, among others, are employed to increase the sample size, which totals 720.000 lip images overall.

Japanese

In [188], the authors investigate visual speech recognition on subjects wearing a mask (i.e., with an occluded face). The applied task is word-level recognition and for that purpose, the authors collect a novel dataset that includes images of masked speakers. The pre-processing procedure mirrors that of non-occluded (i.e., without wearing a mask) speakers. More concretely, the first step is face detection, where several methods such as HoG, Haar-Like features or a deep neural network are employed and the best-performing one is chosen. Then, landmarks of the face are automatically calculated and utilized to extract the lip region of interest. 15 common Japanese words were selected as the vocabulary and uttered by 20 speakers while wearing a variety of masks that included different shapes, fabrics and colors. In total, 5.400 images of masked faces were gathered under variable background and lighting conditions and augmented with an additional 4.500 unmasked samples from a different dataset.

Indonesian

The authors of [303] presented a new dataset in Indonesian for visual speech recognition applications, called *IndoLR*. The vocabulary of words consists of commonly-used daily words in Indonesian. Five women and three men, all with different lip shapes spoke four phrases and ten words in front of a camera with a *480p* resolution (640×480). 30 samples of the same word are collected per person, while for the phrases 50 samples were gathered, totaling 2.400 and 1.600 respectively and collectively they are called the *IndoLR* (Indonesian Lip-Reading) dataset. The lip region was detected using the *MediaPipe* framework[3]. The dataset was used to evaluate two proposed deep learning methods for visual speech recognition of words and phrases.

Another Indonesian dataset was introduced in [150]. The authors utilize a self-developed automated dataset generating tool that, given a target language and dictionary, automatically obtains and annotates videos from the YouTube platform that conform to the Creative Commons license. Videos depicting scenes of news or discussions were collected using the tool and pre-processed to detect the active speaker in cases of multiple ones, in order to create the in-the-wild dataset called *IDLRW*. The vocabulary consists of 100 distinct words and over 48.000 video samples, each containing an utterance of a single word, typically lasting below 1 second in duration with the majority being $3-5$ character words. The resolution is set to 224×224 and the dataset splits follow a $7:2:1$ ratio for training, validation and testing subsets.

2.3.3 Multi-Lingual datasets

The *vVISWa* [304] corpus was collected for multi-view isolated and continuous word speech recognition in three languages. Although the context is Indian, the recordings contained cover the English, Hindi and Marathi languages. It contains recordings of 58 different speakers captured from the front, side and 45 deg angles, a process that took place in a controlled environment resulting in clear images with sufficient illumination without reflections and other occlusions. The vocabulary of the corpus is varied, containing names of cities, colors, months, numerals, fruits as well ass daily communication words for all three languages. A single continuous recording

[3] https://github.com/google-ai-edge/mediapipe

in 720×576 resolution at a frame rate of 25 FPS was performed for each speaker, where the target words were uttered for 10 repetitions. In total, 278.360 samples were collected from native speakers with an additional 9.000 samples obtained from non-Native speakers uttering a set of numerals in all three languages. Furthermore, a subset containing recordings where the speakers applied red fluorescent color to their lips was collected in order to facilitate applications in tasks other than speech recognition, e.g., in tracking or mouth deformations.

A multi-language dataset combining samples from *LRW-1000* [39] and *LRW* [21] was introduced in [183] aiming to facilitate cross-language learning for speech recognition tasks. A curated selection of samples from both source datasets was selected according to scene consistency, which encourages learning of speech-specific features instead of appearance characteristics. Pre-processing in the form of mouth area cropping guided by landmarks of the face is applied only to *LRW*, as the cropped regions are already contained in *LRW-1000*. Since the goal is applying models trained in this corpus to multi-lingual scenarios, for the two language categories offered (i.e., English and Chinese), 100 classes are used for each, with over $80,000$ total samples, balanced between the languages. The dataset is divided into three temporally-disjoint subsets following a $9.5 : 1 : 1$ split for training, validation, test, respectively.

The *LRRo* [227] corpus was combined with *LRW* [21] and *LRW-1000* [39] to form a multi-lingual subset called *LRM* [226] for the purpose of evaluating transfer learning in multiple languages. Composed of the union of each respective subset of the source languages, it contains 75.191 utterances of 141 unique words. Experimental results using this dataset in a multi-lingual training strategy showed that it can improve the performance of models in each source language.

Contrary to standard datasets containing images of the frontal views of the speakers' faces, Zhang et al. [57] constructed a different corpus for the purposes of evaluating a smart necklace in realistic scenarios. The novel dataset contains images of the neck and face captured by a built-in infra-red (IR) camera positioned under the chin area that records deformations of the neck and face shapes during speech. The vocabulary contains commands and digits in the English and Chinese (Mandarin) languages, numbering 54 and 44, respectively, uttered by 20 speakers wearing the sensor. The captured images are pre-processed to remove visual artifacts such as clothing reflections from background lights by producing differential images of adjacent frames as well as color and brightness masks to separate foreground from background. Affine transformations are applied to correct changes in rotation and orientation that occur due to sensor movement and finally a 192×144 rectangle is used to crop the image which is centered at the chin position.

Chapter 3

Reducing the Size of Existing Architectures

As mentioned in the literature overview (see Chapter 2), most published works for visual speech recognition focus on improving final word accuracy without taking into account practical aspects such as model latency, severely limiting their applications in time-critical scenarios. As a result, the proposed models tend to be sizable and cumbersome, requiring a significant amount of parameters to operate at high accuracy, which in turn demands powerful hardware for training and evaluation, restricting their applicability in controlled environments where hardware is not a concern, e.g. computation clusters.

Research towards improving the efficiency of existing methods or designing lightweight and practical models for visual speech recognition is scarce in comparison. One approach to developing lightweight architectures for speech recognition involves techniques that reduce the size of existing network models, thus enabling their deployment in a broader range of hardware devices with different computational capabilities and allows for a wider adoption in various real-life conditions, as opposed to limiting their application in controlled environments.

Neural network compression has been studied extensively by the literature and several strategies have been proposed, including *quantization*, where the network's weights and/or activations are converted from 32-bit floating point into representations of fewer bit-depths (e.g., 16-bit or 8-bit integers), *low-rank adaptation*, where trainable decomposition matrices of lower ranks are inserted in the network and trained instead, or *pruning*, where parameters are removed from the model according to a criterion.

Another similar approach to compression is *knowledge distillation* [99] where a larger network is used as an additional supervisory signal during training of a smaller one, effectively transferring its "knowledge" to a more compact architecture. The end goal of distillation is the same as network compression, although it is achieved by a different approach, since during distillation two networks and therefore more hardware resources are required.

To the best of our knowledge, apart from a few approaches that adopt distillation techniques [268, 282] with the specific goal of shrinking network sizes, no other technique for network compression has been explored in the context of visual speech recognition. In this Chapter, we present an approach that adopts Parameterized Hypercomplex Multiplication (PHM) layers to compress existing architectures that are powerful but large, effectively reducing their hardware demands measured in computational complexity and storage space. In order to obtain a better understanding and insight on how the PHM layers affect several attributes of the models, including performance and compression, we conduct an extensive array of ablative experiments on various components of the architectures. Our results showcase that significant compression is achievable for a minor accuracy penalty, enabling a broader range of applications in hardware-constrained scenarios.

3.1 Background – Temporal Convolution Networks

In this Section, background on *Temporal Convolution Networks* [187, 155] is provided since they form the core sequential processing module of all methods presented in this dissertation.

The TCN is a *fully-convolutional network*, meaning it only uses convolutional and pooling layers as its building blocks, allowing for a more streamlined training process compared to recurrent architectures, such as the LSTM [37]. Following the architectural design of Time-Delay Neural Networks [305], it is a sequence-to-sequence model that takes advantage of 1-dimensional (1D) temporal causal convolutions applied in the temporal dimension of its input. In a causal convolution, the output at time-step t is convolved only with elements from time-step t and earlier, preventing information leakage. For visual speech recognition the TCN uses causal convolutions, but it can function in a non-causal manner for other tasks that do not impose such limitations. A drawback when dealing with sequences of very high length is the limited effective receptive field of the convolution operations, which is handled by progressively increasing the dilation rate for the deeper layers, allowing them to include a wider "view" of the input in each calculation without raising the kernel size to prohibitive numbers that would introduce complexity and implementation issues. Despite the simplicity of its design, the TCN has been shown to perform exceptionally well in the task of VSR and a multitude of variants based on the basic TCN structure have been proposed and widely adopted, effectively replacing recurrent neural networks for sequence processing, obtaining state-of-the-art results, while being computationally more efficient. The TCN is illustrated in Figure 3.1.

The *Multi-Scale TCN* (MS-TCN) is a variant proposed in [110] that aims to enhance its effectiveness by taking advantage of short and long term information within a sequence. In this variant, rather than using one convolution as in the original design ([155]), each block is split into branches that employ convolutions with different kernel sizes, providing the block with several receptive fields (see Figure 3.2(b)). Each convolution in a branch uses C/b kernels, where C refers to number of channels and b is the amount of branches. Their outputs are concatenated, effectively fusing information from multiple temporal scales and retaining the original input channel dimensions. Thus, the multi-scale architecture of the blocks allows the MS-TCN to better model sequences, compared to the vanilla TCN architecture.

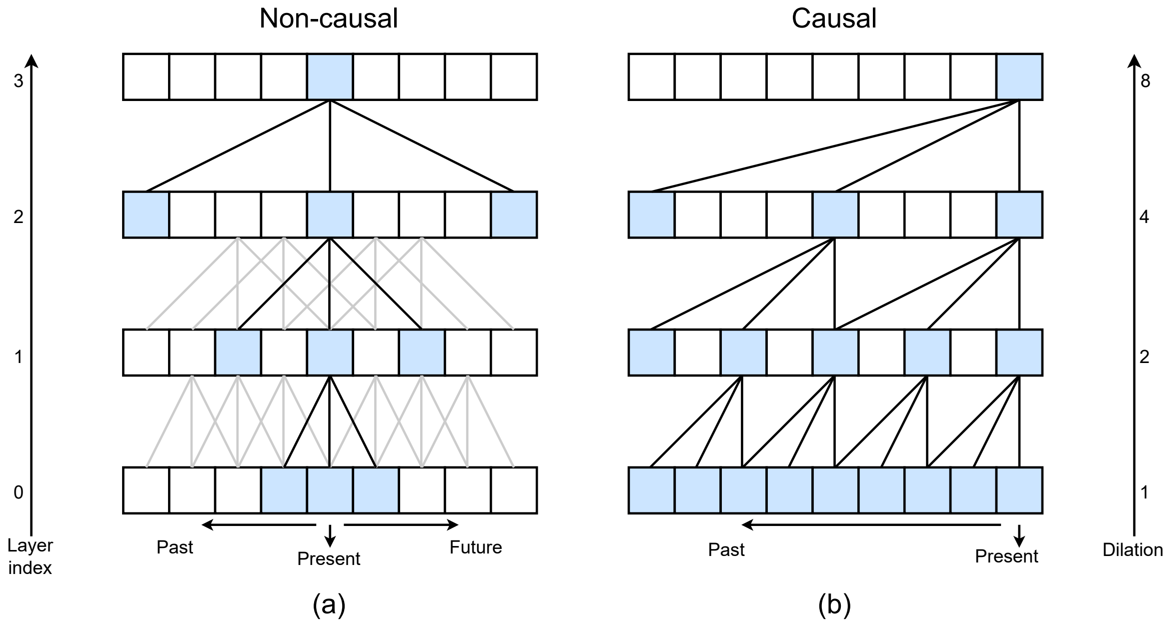

Figure 3.1: Illustration of a Temporal Convolution Network (TCN) that uses 4 computation stages and 1D convolutions with a kernel size of 3. The dilation factor doubles at each stage, starting from 1, allowing the later stages to process information from more distant time steps. Left - non-causal TCN. Right - causal TCN.

Dense connections [47] were added to the MS-TCN architecture in [162] with the aim of overcoming drawbacks of the previous design. These connections allow a convolution layer to receive inputs from all preceding layers within the same block. This architecture retains the existing multiple-kernels-per-block design paradigm and incorporates manually set dilation rates in each block, as opposed to the previous design that uses the same convolution hyper-parameters (kernel size, stride and dilation) for all convolution layers within each block. Channel attention in the form of *Squeeze-and-Excitation* (SE) modules [65] is added before the first convolution of each block to improve performance. This model, named *Densely Connected TCN* (DC-TCN) utilizes these dense blocks for enhanced expressive power since each layer has access to a wider amount of information obtained by the effective receptive sizes of all previous temporal convolutions.

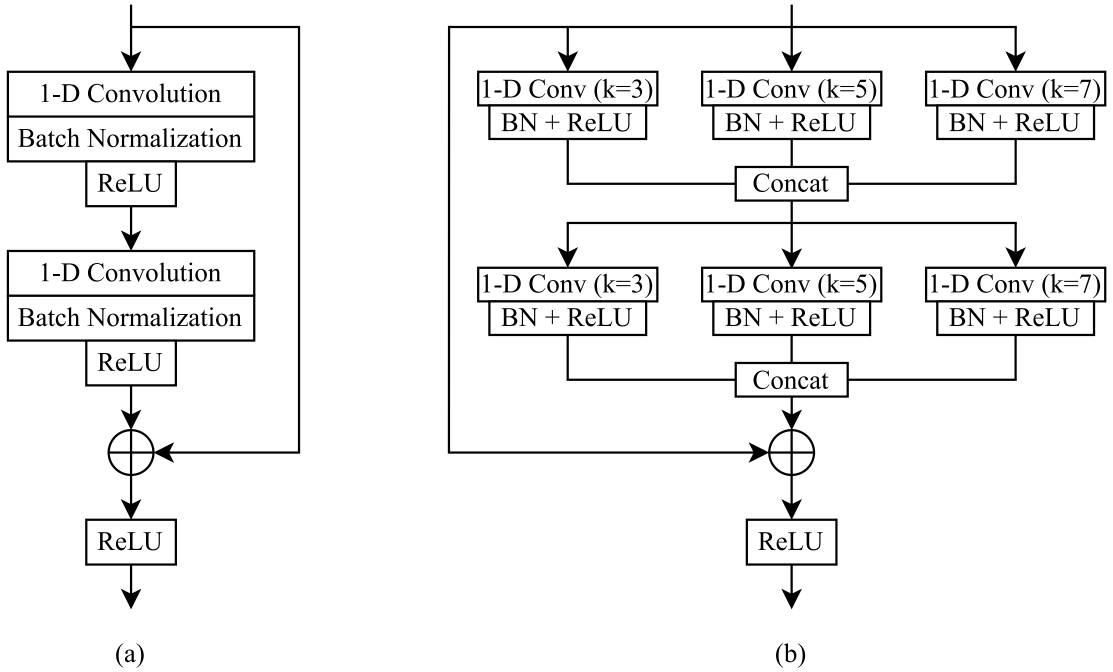

Figure 3.2: (a) Regular TCN block using two 1D convolutions and a residual connection. (b) MS-TCN block. Two groups of three separate kernel sizes are used and their results are concatenated. The DC-TCN model adds Squeeze-and-Excitation modules before the first set of convolutions (one module per layer) and connects several such blocks with dense connections.

3.2 Parameterized Hypercomplex Multiplication Layers

Parameterized Hypercomplex Multiplication (PHM) layers were proposed in [306] as a trainable network component that can replace the typical fully-connected layer found in various neural network architectures such as the Multi-Layer Perceptron (MLP) [307], the Long Short-Term Memory network (LSTM) [37] or the Transformer [50].

A fully-connected layer transforms its input $x \in \mathbb{R}^{d_{in}}$ to an output $y \in \mathbb{R}^{d_{out}}$ via a matrix multiplication and a bias offset:

$$y = W^T x + b, \tag{3.1}$$

where, $W \in \mathbb{R}^{d_{in} \times d_{out}}$ and $b \in \mathbb{R}^{d_{out}}$ are the weight matrix and the bias vector, respectively.

A PHM layer retains the same notation but calculates the weight matrix $\mathbf{W}$ as a summation of Kronecker products between a set of learned matrices:

$$W = \sum_{i=1}^{n} A_i \otimes S_i, \quad i = 1, \ldots, n, \tag{3.2}$$

where, $A_i \in \mathbb{R}^{n \times n}$ and $S_i \in \mathbb{R}^{\frac{d_{in}}{n} \times \frac{d_{out}}{n}}$ are parameter matrices, n is a hyper-parameter defined by the user with $n \in \mathbb{Z}_{\geq 1}$ that determines the number of matrices to be summed and the dimensions d_{in}, d_{out} are divisible by n.

The value of hyper-parameter n determines the amount of parameter sharing within these layers and consequently controls the final layer size without altering its output dimensionality. Due to the Kronecker product, PHM layers allow reusing of weight parameters within the same layer leading to an overall reduction in required trainable parameters for networks that utilize them, effectively lowering overall network sizes.

The Kronecker product between any two matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{S} \in \mathbb{R}^{p \times q}$, indicated here by the symbol $\otimes$, is a block matrix that is defined as:

$$\mathbf{A} \otimes \mathbf{S} = \begin{bmatrix} A_{11}\mathbf{S} & \ldots & A_{1n}\mathbf{S} \\ \vdots & \ddots & \vdots \\ A_{m1}\mathbf{S} & \ldots & A_{mn}\mathbf{S} \end{bmatrix} \in \mathbb{R}^{mp \times nq}, \tag{3.3}$$

where A_{ij} refers to the element at the i^{th} row and j^{th} column of the matrix A. Using the Kronecker product in this fashion, it effectively enables reducing the number of required parameters of the layer by a factor $\approx 1/n$ compared to the standard fully-connected layer, while keeping the same functionality and dimensions, allowing for straightforward integration in existing architectures by simply replacing each desired layer. An example of a Kronecker product between two matrices is shown in Figure 3.3.

An extension of PHM layers, called *Parameterized Hypercomplex Convolution* (PHC) layers was introduced in [308], where the same layer formulation (Equation 3.2) is applied to the filters of convolution layers. More specifically, the operation of a standard convolution layer is defined as:

$$y = W * x + b, \tag{3.4}$$

where the output $y \in \mathbb{R}^{d \times t}$ of dimension t is produced by a convolution operation $(*)$ of the c-channel input $x \in \mathbb{R}^{t \times c}$ with the weight matrix $W \in \mathbb{R}^{c \times d \times k \times k}$ of $k \times k$-sized filters and of output dimension d.

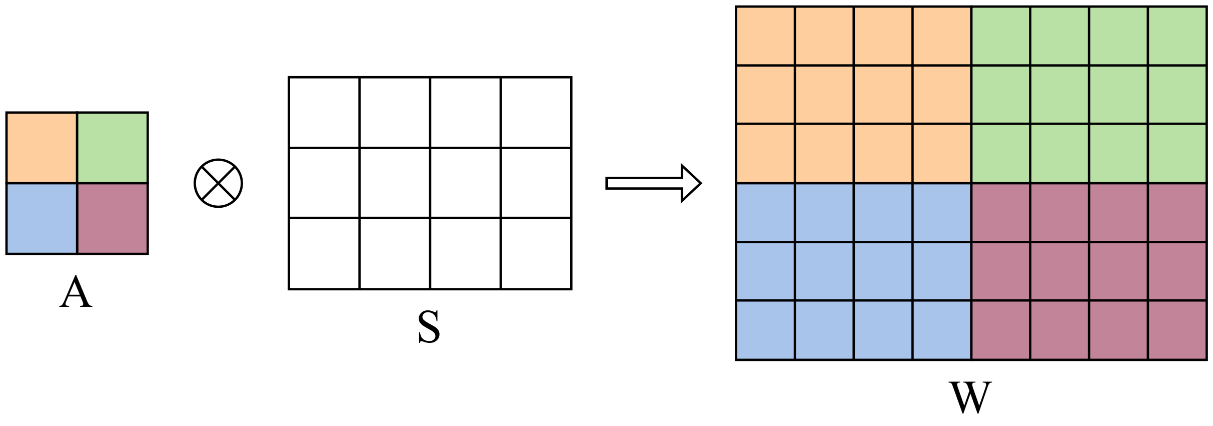

Figure 3.3: Depiction of a Kronecker product between two matrices A and S. The result is a matrix with greater spatial dimensions, while the sum of parameters from A and S is smaller than that of W. PH layers exploit this principle to drastically reduce the required parameters of a layer. The parameter sharing occurs since matrix S participates in several calculations when obtaining matrix W.

Analogous to PHM layers, $A_i \in \mathbb{R}^{n \times n}$ and $S_i \in \mathbb{R}^{\frac{c}{n} \times \frac{d}{n} \times k \times k}$. The matrices A_i and S_i are learned during training using standard back-propagation. Typically, in a convolutional neural network, c and d assume large values compared to the filter size $k \times k$ (where k is typically in the single digits, e.g., $3, 5, 7$ and c, d in the hundreds). Reducing c and d by a factor of n, depending on user setting, achieves a significant reduction of required parameters for convolution layers, which are a rather expensive component in convolutional neural networks.

The authors of [308] also demonstrate that for certain values of n, PHC layers can learn the convolution rules from the data for various domains, such as *real* $(n = 1)$ or *quaternion* $(n = 4)$.

3.3 Visual Speech Recognition Networks

The architectures used for VSR follow the standard two-component design, where a convolutional neural network architecture extracts visual representation features from the input sequence and a sequential model processes the temporal aspect of the extracted features.

As a convolutional feature extractor, we use the 18-layer residual network (ResNet) as proposed in [31], where the very first three-layer block (2D convolution, normalization and pooling) is replaced an equivalent one with three-dimensional operations [26] to improve the model's extraction performance for the task and the final fully-connected layer that normally performs classification is removed. More specifically, since the input sequence constitutes a volume with a dimensionality of $T \times H \times W$, (a sequence with a length of T frames, each with a spatial size of $H \times W$ pixels), the very first layer uses a 3D convolution operation, with a kernel size of $5 \times 7 \times 7$, followed by 3D batch normalization for training stability, a non-linear activation function (e.g., a Rectified Linear Unit) and a 3D max pooling operation which reduces the final spatial dimension to half, keeping computation requirements low. The 3D convolution functions similarly to its 2D equivalent, however in this operation, the filters slide along all three dimensions. A stride of $1 \times 2 \times 2$ is used in order to further reduce the spatial dimensions of the input (as the first value corresponds to the time dimension). The remainder of the CNN consists of a sequence of four residual blocks following the standard design as proposed in [31] that uses two stacks per block, with each stack consisting of 2D convolution, batch normalization and non-linear activation layers. In addition, each block increases the output channels of the convolution layers (kernel size of 3), while a stride of 2 down-scales the feature maps. An average pooling operation is added at the very end to further condense the spatial information.

The extracted features obtained by the CNN of first stage are fed to a sequential model based on a Temporal Convolution Network (TCN) [155], to model the temporal inter-dependencies of the entire sequence. In this Chapter, we use two architectures derived from TCNs, the Multi-Scale-TCN [110] and Densely-Connected-TCN [162], described previously in Section 3.1. At the top of the entire network, a Softmax layer is added to predict the single spoken word from the video input. The overall architecture is depicted in Fig. 3.4.

3.4 Experiments

3.4.1 Dataset Pre-processing

We use the *Lip Reading in the Wild* (LRW) dataset [21] to train and evaluate our models. A detailed overview of the LRW corpus is provided in Chapter 2 under Section 2.3, therefore here we only describe the pre-processing steps performed on

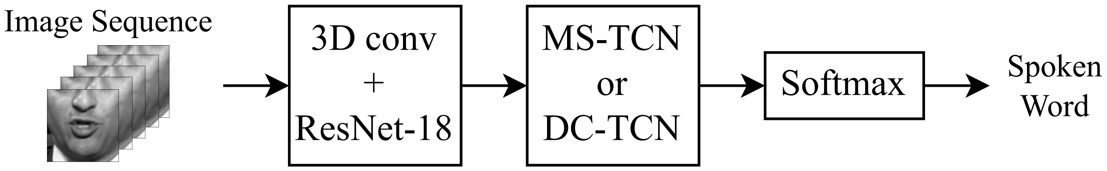

Figure 3.4: The architecture of the end-to-end model used in this Chapter. An 18-layer residual network extracts features, while sequence modeling is performed using a TCN variant and the final classification is obtained using a Softmax layer. PHM layers are added in the residual and temporal convolution networks, replacing the original.

the raw data before training.

We apply a multi-step procedure of processing that is used by previous works (e.g., [110, 162]) and is outlined below: First, a face tracking network is used to detect the face in an image and landmarks are computed using a face alignment network. Then, a mean face shape serves as a guide to remove size and rotation differences and to keep images uniform throughout the training set. Cropping the mouth regions of interest with a 96×96 bounding box follows and the final frame is converted to gray scale and normalized by subtracting the mean and dividing by the standard deviation of the training set. The first two steps of this procedure serve to align the data, keeping the mouth area as close to the center of the image as possible, while the final step simplifies it by reducing the channels from RGB to gray, as there seems to be no difference in performance when using RGB images instead of gray [110].

3.4.2 Training Setup

All model weights are initialized from random values (no pre-trained models are used) and trained in an end-to-end manner. The total number of training epochs is set to 100 and after each epoch the model is validated using the LRW validation set and a checkpoint of the weights is saved. At the end of training, the best-performing checkpoint is evaluated on the LRW test set. The optimizer used is *AdamW* [309] without any warm-up steps and the loss function to be optimized during training is the standard Cross-Entropy loss:

$$l = -\sum_{i=1}^{C} y_i \log(\hat{y}_i), \tag{3.5}$$

where, in our case, $C = 500$ which are the classes of the LRW dataset, y_i represents the ground truth class label and $\hat{y}_i$ is the network prediction.

A batch size of 32 is used for all training experiments, this way each experiment can be completed using a single graphics processing unit with at least $11GB$ of video memory, which is sufficient for storing the gradients for back-propagation. When using the Densely-Connected TCN sequence model, the initial learning rate is set to 0.0003, while for the Multi-Scale TCN model it is increased to 0.003, as we found this offers an overall accuracy improvement of about 1%. Setting the initial learning rate to a higher value leads to an inability of convergence, while reducing it below 0.0003 reduces the convergence time and results in lower overall accuracy at the end of the 100 epochs. During training, the learning rate is annealed using a cosine scheduler, which at the end of every epoch scales (or decays) its value according to:

$$\eta_t = \eta^i_{min} + \frac{1}{2}(\eta^i_{max} - \eta^i_{min}) \left(1 + \cos\left(\frac{T_{cur}}{T_i}\pi\right)\right), \tag{3.6}$$

where η^i_{min} and η^i_{max} are ranges for the learning rate and T_{cur} measures elapsed epochs since the last "restart", updated at each batch iteration t. The value of T_i simulates a warm restart once that number of epochs have elapsed, as $T_i = T_{cur}$ and $cos(\pi) = 0$. In our experiments, we follow [110] which does not use warm restarts, so T_i is always set at the maximum number of epochs for the training run, as this scheme was found to work best. Therefore, the above equation simplifies to:

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min}) \left(1 + \cos\left(\pi\frac{T_{cur}}{T_{max}}\right)\right), \tag{3.7}$$

where η_t, η_{min}, η_{max} refer to the current, minimum and maximum values of the learning rate, respectively, while T_{cur} and T_{max} measure the number of iterations. Since in our case we set the minimum and maximum values for the learning rate at 0 and 1 respectively, we finally have:

$$\eta_t = \frac{1}{2}\left(1 + \cos\left(\pi\frac{epoch}{max\_epoch}\right)\right), \tag{3.8}$$

with $epoch$ referring to the current epoch in the training process (and the value max_epoch corresponds to the total epochs that we set for each experiment. This annealing rate is applied after every epoch.

First, we train the models with regular layers to serve as the baselines for our comparisons. Then, without changing their architecture, we replace all respective convolution layers in the CNN and TCN components with their Parameterized Hypercomplex (PH) counterparts, which include PHM and PHC layers and begin training using new randomly-initialized weights. This training procedure is repeated for several values of the hyper-parameter n to obtain models with varying amounts of compression offering different degrees of compromise between accuracy and model size. A detailed list of all hyper-parameters that are used for all experiments can be found in Table 3.1.

For improved regularization and to increase the robustness of the models, during training, the following data augmentation methods are performed:

- *MixUp* [66], where new training examples and their corresponding targets are constructed by linearly combining existing ones.
- Random spatial cropping to 88×88 as well as random horizontal flipping, the latter occurring with a probability of 0.5.
- Variable length augmentation [110] where temporal cropping is randomly applied to each sequence prior and after the target word boundaries.

Regarding the audio data, each segment is normalized to zero mean and standard deviation of 1, to account for variations in different levels of loudness between the speakers. Babble noise at several Signal-to-Noise levels (chosen randomly for each batch) is also added.

All experiments are conducted using the `PyTorch`[1] framework on an NVIDIA RTX 2080 Ti dedicated graphics processor. Pseudo-random generator seeds are manually set to 1 for *PyTorch*, as well as for *Python*'s and *Numpy*'s *random* libraries, it is therefore possible, that a different seed could produce a higher or lower final accuracy. The *CUDNN* back-end “benchmark” flag is also set to *True*. Parameter counts on all tables are obtained using the `torchinfo`[2] library.

[1] https://pytorch.org/

[2] https://github.com/TylerYep/torchinfo

Table 3.1: Training hyper-parameters used in our experiments. The optimizer employed is *AdamW*.

Category	**Hyperparameter**	**Value**
Optimizer settings	Learning rate	0.0003
	β_1	0.9
	β_2	0.999
	Weight decay	0.01
Scheduling	Rate schedule	cosine
	Maximum epochs	100
	Batch size	32
Regularization	Mixup α	0.4
	DC-TCN dropout	0.2

3.4.3 Evaluation

All models are evaluated on the LRW [21] test set. As input, a cropped center patch of 88×88 is used for each image. The performance index for scoring is top-1 accuracy, measured as the percentage of correct word classifications. Model size refers to the trainable parameter count, measured in millions. The models are compared against other single word VSR methods from the literature and the results are presented in Table 3.2.

As an additional comparison, we compare our most compact methods with related works from the literature that focus on developing low-resource models for practical VSR applications. We consider as "low-resource" all models below 10 million overall parameters and as "very-low-resource" all models below 5. The results are presented in Table 3.3.

Following, since applicability of the compressed models is the overall goal, we offer a practical evaluation of our proposed models which showcases tangible results in conditions resembling real-world scenarios, i.e., application in a device. We evaluate the running times using CPU and GPU with regards to model size and accuracy in Table 3.4.

Table 3.2: Evaluation of our methods on the LRW test set. We compare our method with recent works from the literature. In works where multiple models were proposed, we report the values obtained by the best performing model. Our models are highlighted. SE denotes Squeeze-Excitation blocks and ↑ indicates that higher is better.

Method	**Parameters ($\times 10^6$)**	**Accuracy ↑**
3D conv. + ResNet-18 + MS-TCN [110]	36.4	85.3%
3D conv. + SE-ResNet-18 + 3-layer B-GRU [69]	59.4	88.4% [1]
Alternating *ALSOS* + ResNet-18 + MS-TCN [160]	41.2	87.0%
3D conv. + ResNet-18 + MS-TCN + KD[2] [268]	36.4	88.5%
Vosk + 3D conv. + SE-ResNet-18 + 2-layer B-LSTM [64]	50.0*	88.7%
3D conv. + ResNet-18 + DC-TCN [148]	52.5	90.4%
3D conv. + ResNet-18 + MS-TCN [277]	36.0	87.4%
3D conv. + ResNet-18 + DC-TCN + *MTLAM* [186]	67.0*	91.7%
3D conv. + ResNet-18 + MS-TCN	36.4	87.2%
3D conv. + ResNet-18 + MS-TCN, (PH layers, N=2) (ours)	18.3	86.5%
3D conv. + ResNet-18 + MS-TCN, (PH layers, N=4) (ours)	9.4	85.0%
3D conv. + ResNet-18 + MS-TCN, (PH layers, N=8) (ours)	4.9	84.2%
3D conv. + ResNet-18 + MS-TCN, (PH layers, N=16) (ours)	2.8	83.0%
3D conv. + ResNet-18 + DC-TCN	52.5	89.3%
3D conv. + ResNet-18 + DC-TCN, (PH layers, N=2) (ours)	26.7	89.1%
3D conv. + ResNet-18 + DC-TCN, (PH layers, N=4) (ours)	13.8	88.4%
3D conv. + ResNet-18 + DC-TCN, (PH layers, N=8) (ours)	7.4	87.4%
3D conv. + ResNet-18 + DC-TCN, (PH layers, N=16) (ours)	4.8	86.5%
3D conv. + ResNet-18 + DC-TCN, (PH layers, N=32) (ours)	9.9	87.8%
3D conv. + ResNet-18 + DC-TCN, (PH layers, N=64) (ours)	33.7	88.8%

[1] Using word boundary information [2] KD–Knowledge Distillation * our implementation

3.4.4 Detailed Parameter Analysis

We provide a detailed overview of the parameter savings achieved per network component as a result of substituting standard convolution and linear with PH layers (PHC and PHM, respectively), for the networks using the DC-TCN as a sequence model. We report measurements for the models reported in Table 3.2 that use PHM layers in the SE modules, as well as for the models from the ablative analyses of Tables 3.7 and 3.9. We also report the model sizes in Bytes when the weights saved to disk, as such a value is important when considering device restrictions such as

Table 3.3: Low-resource comparison between recent works from the literature. The evaluation metric is word accuracy on the LRW test set. Parameter counts are shown as reported in each respective paper. Proposed models are highlighted as “(ours)”.

Resource Setting	**Method**	**Parameters** ($\times 10^6$)	**Accuracy ↑**
”Very Low” (< 5M)	MobiVSR-1 [266]	4.5	72.2%
	ViViT + RA [277]	3.9	75.6%
	ShuffleNetV2 (0.5×) + TCN [268]	2.9	79.9%
	ShuffleNetV2 (1×) + TCN [268]	3.8	82.7%
	ResNet18 + MS-TCN, N=16 (ours)	2.8	83.0%
	ResNet18 + MS-TCN, N=8 (ours)	4.9	84.2%
	ResNet18 + DC-TCN, N=16 (ours)	4.8	**86.5%**
”Low” (< 10M)	MobiVSR-2 [266]	5.2	73.0%
	MobiVSR-3 [266]	5.9	73.4%
	MobiVSR-4 [266]	6.6	74.0%
	ResNet18 + MS-TCN, N=4 (ours)	9.4	85.0%
	ShuffleNetV2 (1×) + DS-MS-TCN [268]	9.3	85.3%
	ResNet18 + DC-TCN, N=8 (ours)	7.4	87.4%
	ResNet18 + DC-TCN, N=32 (ours)	9.9	**87.8%**

Table 3.4: Practical evaluation of our methods presented in Table 3.2. Model size refers to the total parameter size when saved to disk. Reported runtime is in milliseconds (ms) and measurements are obtained with the Pytorch profiling tool using a 29-frame gray-scale video sequence input of 88×88 averaged across 20 runs. The GPU used is a NVIDIA RTX 2080 Ti, while the CPU is an Intel Xeon 3204. ↓,↑ indicate that lower and higher is better, respectively.

Method	**Size (MB)**	**Runtime (GPU) ↓**	**Runtime (CPU) ↓**	**Accuracy ↑**
3D conv. + ResNet-18 + MS-TCN	139	11.1	955	87.2%
3D conv. + ResNet-18 + MS-TCN, (PH layers, N=2)	70	13.2	1005	86.5%
3D conv. + ResNet-18 + MS-TCN, (PH layers, N=4)	36	14.1	1155	85.0%
3D conv. + ResNet-18 + MS-TCN, (PH layers, N=8)	19	16.2	1625	84.2%
3D conv. + ResNet-18 + MS-TCN, (PH layers, N=16)	11	20.2	1590	83.0%
3D conv. + ResNet-18 + DC-TCN	201	26.5	1100	89.3%
3D conv. + ResNet-18 + DC-TCN, (PH layers, N=2)	103	29.2	1225	89.1%
3D conv. + ResNet-18 + DC-TCN, (PH layers, N=4)	54	31.0	1375	88.4%
3D conv. + ResNet-18 + DC-TCN, (PH layers, N=8)	29	34.7	1805	87.4%
3D conv. + ResNet-18 + DC-TCN, (PH layers, N=16)	19	37.3	2310	86.5%

storage, amount of memory and bandwidth.

Table 3.5: Detailed parameter analysis per model component. Percentage reductions are calculated from the ”Standard” model. In this table, the parameters for the ResNet-18 include the 3D convolution and batch normalization layers. Total size includes both components as well as the Fully-Connected layer used as classifier at the end of the network. The ”SE with PHM” column denotes whether PHM layers were used in the SE module.

Model	SE with PHM	Parameters ($\times 10^6$)			Size (MB)	Accuracy ↑
		ResNet-18	DC-TCN	Total		
Standard layers	–	11.18	40.53	52.54	201	89.3%
PH layers, N=2	✓	5.60 (-49.9%)	20.28 (-49.9%)	26.72 (-49.1%)	103 (-48.7%)	89.1% (-0.2%)
PH layers, N=4		2.81 (-74.8%)	10.17 (-74.9%)	13.82 (-73.6%)	54 (-73.1%)	88.4% (-0.9%)
PH layers, N=8		1.42 (-87.2%)	5.18 (-87.2%)	7.44 (-85.8%)	29 (-85.5%)	87.4% (-1.9%)
PH layers, N=16		0.80 (-92.8%)	3.21 (-92.0%)	4.84 (-90.7%)	19 (-90.5%)	86.5% (-2.8%)
PH layers, N=32		0.99 (-91.1%)	8.11 (-80.0%)	9.94(-81.0%)	39 (-80.5%)	87.8% (-1.5%)
PH layers, N=64		5.18(-53.6%)	27.73 (-31.5%)	33.74 (-35.7%)	134 (-33.3%)	88.8% (-0.5%)
PH layers, N=2	–	5.60 (-49.9%)	22.31 (-44.9%)	28.75 (-45.2%)	111 (-44.7%)	88.8% (-0.5%)
PH layers, N=4		2.81 (-74.8%)	13.21 (-67.4%)	16.85 (-67.9%)	65 (-67.6%)	87.7% (-1.6%)
PH layers, N=8		1.42 (-87.2%)	8.69 (-78.5%)	10.95 (-79.1%)	43 (-78.6%)	86.6% (-2.7%)
PH layers, N=16		0.80 (-92.8%)	6.73 (-83.3%)	8.36 (-84.0%)	33 (-83.5%)	86.2% (-3.1%)
PH ResNet, N=2	–	5.60 (-49.9%)	38.50 (-5.0%)	44.94 (-14.4%)	179 (-10.9%)	89.3% (±0.0%)
PH DC-TCN, N=2	✓	11.18 (±0.0%)	20.28 (-49.9%)	32.30 (-38.5%)	129 (-35.8%)	88.5% (-0.8%)
PH ResNet, N=16	–	0.80 (-92.8%)	37.01 (-8.6%)	38.64 (-26.4%)	154 (-23.3%)	89.6% (+0.3%)
PH DC-TCN, N=16	✓	11.18 (±0.0%)	3.21 (-92.0%)	15.23 (-71.0%)	60 (-70.1%)	% 87.2(-2.1%)

3.4.5 Audio Speech Recognition

Furthermore, we use PH layers on the audio recognition model proposed in [110] to evaluate their performance in the same task for a different type of sequential input. This model shares some architectural similarities with the ones used for the VSR (see Section 3.3), namely it follows a two-component design, with an 18-layer residual network superseded by a Multi-Scale TCN and Softmax at the end to obtain the output word, since the goal (word recognition) has not changed. Due to the nature of the audio signal, the residual network of this model utilizes 1D convolution layers instead of the typical 2D and the 3D convolution at the front of the model is removed, while the Multi-Scale TCN is unchanged. The training setup is the same as in Section 3.4, where the standard model is trained first, followed by training the models with PH layers for several values of the n hyper-parameter. In line with the previous experiments, the training process is end-to-end with a random initialization of weights and the learning rate used is 0.003, which offers a final accuracy bonus of 0.5% compared to using an initial lower learning rate. The results after training for a maximum of 100 epochs and a comparison with other works from the literature for audio-only data are shown in Table 3.6.

Table 3.6: Audio speech recognition evaluation on the LRW test set. The network architecture used is a (1D) ResNet-18 and MS-TCN. Size refers to the network size measured in millions of parameters ($\times 10^6$) and ↑ indicates that higher is better.

Method	**Size**	**Accuracy ↑**
1D ResNet-18 + BGRU [104]	15.2*	96.7%
BLSTM + BLSTM [30]	12.4*	97.9%
2D ResNet-18 [71]	11.6	92.2%
VGG-16 [71]	14.9	92.0%
1D ResNet-18 + MS-TCN [110]	29.0	98.9%
PH layers, N=2	14.5	98.7%
PH layers, N=4	7.2	98.5%
PH layers, N=8	3.6	98.3%
PH layers, N=16	1.8	98.2%

* our implementation

3.4.6 Ablation Studies

The DC-TCN architecture takes advantage of channel attention in the form of *Squeeze-and-Excitation* (SE) [65] modules. A SE module first generates a global embedding of the input and then captures channel-wise dependencies with an MLP network, two fully-connected layers and a non-linear activation function. Global average pooling is used to aggregate feature maps across their spatial dimensions. The MLP network is comprised of two fully-connected layers in a bottleneck design where the first layer reduces the input channels and the second layer subsequently restores them to the initial amount, while a non-linear activation function is applied between them. A fixed ratio (hyper-parameter) determines the intermediate channel dimensions, with higher ratios providing more computation savings. Finally, a Sigmoid function acts as a gating mechanism to capture channel-wise dependencies and its activations are used to rescale the initial feature maps. These modules are added in the beginning of each dense block, before the convolution operations (see Figure 3.2 (b)) and enable the network to selectively enhance its more useful features while suppressing other, less-effective ones. An illustration of the Squeeze-Excitation module is shown in Figure 3.5.

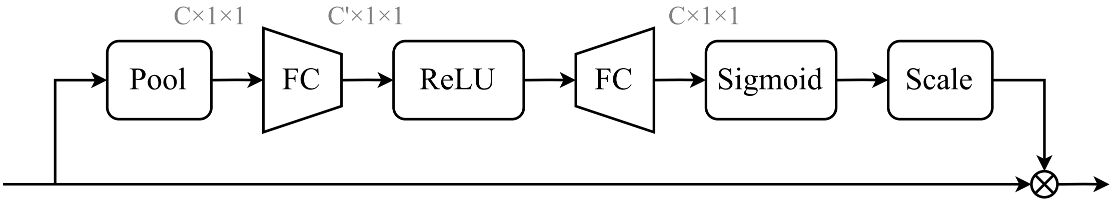

Figure 3.5: The Squeeze-and-Excitation module that is added at each block in the DC-TCN model in our experiments. C' refers to the reduced amount of channels, which is determined by a ratio provided as a hyper-parameter.

Since the DC-TCN uses four dense blocks of three multi-branch layers and each layer in turn contains three SE modules (one per different kernel size), there are a total of 36 SE modules, which amount to a non-trivial percentage of parameters in the overall architecture. We experiment with removing the PH layers from the SE modules specifically (keeping the standard ones) to evaluate whether the reduced network capacity in this component which may affect the channel attention process causes the observed drop in accuracy (Table 3.2). Training hyper-parameters remain as before (refer to Table 3.1 for details). The results are shown in Table 3.7.

Table 3.7: Ablative analysis of omitting parameterized hypercomplex layers in the channel attention mechanism within the DC-TCN architecture. In these models, the 3D convolution layer uses standard convolutions. Evaluation is performed on the LRW test set. ↑ indicates that higher is better.

Model	**Parameters** ($\times 10^6$)	**Accuracy** ↑
Standard layers	52.5	89.3%
PH layers, N=2	28.7	88.8%
PH layers, N=4	16.8	87.7%
PH layers, N=8	10.9	86.6%
PH layers, N=16	8.3	86.2%

We also experimented with using parameterized hypercomplex layers for the 3D convolution at the first stage of the network as well and include these results in Table 3.8. We use the same architecture as the one reported in the previous ablation study, (Table 3.7), meaning that the SE modules in the DC-TCN use standard fully-connected layers. Substituting this layer with a PH equivalent requires an additional step of data pre-processing due to the dimensions of the non-hypercomplex 3D convolution originally used in the network. More specifically, the input tensor undergoes a channel dimensionality expansion so that it matches with the layer. We experimented with two different expansion methods: filling all the additional dimensions with zeros (channel padding with zeros) and with repeating the values of the single dimension (since the images are converted to gray-scale) as many times as required; the former yielded a $+1.0\%$ increase in overall accuracy compared to the latter. While performance is comparable to not including these layers in the first step and higher in some cases (see Table 3.2), a drawback of this setup is the additional computational overhead introduced by this extra pre-processing step which slows down runtime due to the required calculations and memory. We use the same training hyper-parameters of Table 3.1 as in the previous ablation study.

Table 3.8: Impact of using the PHC layer in the 3D convolution of the VSR model using the DC-TCN. All convolution layers in the building blocks (3D convolution, ResNet-18 and DC-TCN) of the architecture have been substituted with PH layers. Evaluation is performed on the LRW test set. ↑ indicates that higher is better.

Network	**Parameters ($\times 10^6$)**	**Accuracy ↑**
Standard	52.5	89.3%
PH layers, N=2	28.7	88.4%
PH layers, N=4	16.8	88.3%
PH layers, N=8	10.9	86.6%
PH layers, N=16	8.3	87.0%

Finally, we explore how these models function when we omit parameterized hypercomplex layers in one of their major components, i.e., the convolutional feature extractor or the sequence processing model. This comparison allows for a better understanding of how the PH layers affect model performance as its representation capacity is altered by compressing individual components. Furthermore, we can investigate where we can obtain the most benefit from using these layers depending on the available resources, which can affect deployment strategies. For this comparison, we use the baseline VSR architecture with the DC-TCN as a sequence model, for the least and most compact models of our experiments, i.e., with the hyper-parameter n set to 2 and 16, respectively. In the cases where PH layers are used in the DC-TCN, they are used in the SE attention mechanism as well, in line with the initial experiments reported in Table 3.2. Again, to keep comparisons meaningful, we do not change any training hyper-parameter (see Table 3.1 for details) that might affect the result. The results are tabulated in Table 3.9.

Table 3.9: Ablative analysis on using PH layers only in one component of the entire architecture. Percentages in parentheses are calculated by comparing with the baseline. Evaluation is performed on the LRW test set. ↑ indicates that higher is better.

Network	**Parameters ($\times 10^6$)**	**Accuracy ↑**
Standard	52.5	89.3%
N=2, PH ResNet	44.9	<u>89.3</u>%
N=2, PH DC-TCN	32.3	88.5%
N=16, PH ResNet	38.6	**89.6%**
N=16, PH DC-TCN	15.2	87.2%

3.5 Discussions

3.5.1 Accuracy-Compression Trade-off

From the experimental evaluation (Table 3.2), substituting standard layers (linear or convolutional) with Parameterized Hypercomplex (PH) counterparts leads to a significant parameter reduction coupled with a minor accuracy loss. The hyper-parameter n determines the amount of compression achieved by using these layers with larger values of n achieving high parameter savings at a corresponding accuracy reduction as network size shrinks. The degradation in VSR performance is arguably the result of reduced network capacity, as for this task and dataset larger models tend to perform better. We note that even at high compression rates ($n = \{8, 16\}$), our models perform similarly to other larger architectures (e.g., [160, 277]) but at a fraction of overall network size. Raising the value of n higher offers diminishing returns in terms of compression as the dimensionality of matrix A in the PH layers increases and so do the parameters required, explaining the larger model sizes (for $n = \{32, 64\}$) as well as the recovering of accuracy. When comparing our most compact models with other low-parameter networks proposed in the literature, our compressed models outperform all other works in both the under 10M and 5M parameter settings, which demonstrates our models' superiority for resource-constrained applications.

Surprisingly, *not* using PH layers in the SE module causes a further minor degradation in the final accuracy, ranging from $0.3\% - 1.1\%$, a result suggesting that these networks do not benefit from additional parameter capacity in the attention mechanism and that the convolution layers are mostly responsible for the overall performance of the architecture (Table 3.7). This also indicates that the networks learn to model the data more efficiently when parameters are restricted in the SE module as well and that using PH layers in this component is beneficial for performance.

Considering the effect of PH layers on the overall performance when used in the 3D convolution as well, an additional minor accuracy drop occurs (see Table 3.8), which, as noticed previously, can be attributed to the reduced network capacity and supported by the observation that smaller networks generally perform worse. Since this convolution's output filters are few, it does not require a large amount of parameters and very little savings in terms of network size can be gained when this particular component uses PH layers. This slight degradation in network performance becomes more evident as n and therefore compression of the overall network is increased. This layer uses very few filters when compared to the other components (e.g. convolutions in the later blocks of the residual network), resulting in some accuracy recovery at $n = 16$ as this value actually slightly increases the total parameters of this layer. It should be noted that this minor increase is not noticeable in the overall parameter count (millions) of the entire model.

Moreover, when PH layers are used only in one of the two main components of the overall architecture (either the deep convolutional visual feature extraction network or in the temporal convolution model used for sequential processing), we do not achieve the same parameter savings as when both components use these layers, which is to be expected. Surprisingly, when PH layers are used only in the residual network, the overall accuracy is not hampered and it actually improves in the case where $n = 16$, suggesting that this component is over-parameterized for this dataset, and also that the DC-TCN, in the presence of a "weaker" convolutional feature extractor, learns to model the data more efficiently, a result that draws parallels to the ablative study on the SE attention modules. At the same time, the DC-TCN with PH layers has a more profound impact on accuracy, which we attribute to the significant compression achieved, as this network occupies the majority ($\sim 77.1\%$) of the overall parameters of the model. These results indicate that using PH layers can offer a cost-less improvement in accuracy alongside a modest network compression and is ideal when the available hardware capabilities permit it.

Further experimentation can be performed by carefully tuning every layer in either network by choosing higher values of n for layers that have high dimensionality, for instance, the deeper layers in the residual network that use convolutions with more output filters. Choosing an appropriate value for the hyper-parameter n on a per-layer basis is possible and we believe would lead to even greater model compression with a correspondent accuracy trade-off. Such a process with a goal of higher parameter savings or a size-to-performance compromise depending on the the use-case would require either a substantial number of experiments set manually or costly automated techniques (i.e., network-assisted) to achieve and is outside the scope of this Chapter, but can be explored in the future.

Finally, in the case of audio speech recognition, the accuracy lost due to network compression is much less pronounced ($< 1.0\%$), where our model with $n = 16$ requires a miniscule fraction of the initial network size (6.2%) for nearly identical performance (98.9% vs 98.2%) which also out-performs other approaches from the literature that are several times larger in size. This observation suggests that due to the reduced dimensionality of the audio modality as well as the relevant simplicity of the dataset with regards to its audio data (i.e., uncomplicated words and absence of noise), large models are not necessary for achieving high audio recognition accuracy.

3.5.2 PH Layers' Effect on Computation

We should mention a current limitation of PH layers which lies in the implementations rather than the theoretical background. While training our models, we discovered that although the use of such layers can achieve significant savings in model sizes, it causes a slight increase in memory demands for the intermediate calculations of the sum factor. This effect is more pronounced at higher values of n which governs the dimensions of matrices A and S as well as the amount of components in the sum (see Equation 3.2) and also depends on the total number of layers in the overall network.

However, the evaluation presented in Table 3.4 showcases that in practice, the penalty in latency due to the computational overhead of the PH layers is not a significant factor. More specifically, when using a graphics accelerator, the difference in running speeds is negligible ($<$ 11 milliseconds), even at the higher compression rates. On the contrary, this effect is more noticeable when inference is executed in a CPU.

We speculate that this is caused by the fact that graphics processing units are heavily optimized specifically for convolutions and matrix multiplications either through software libraries or via specialized hardware (such as Tensor cores), while a CPU can not take advantage of multiple cores the same way a GPU can for the same benefit. The current implementations of the Kronecker product do not seem to utilize parallelization or other linear algebra optimizations, unlike convolutions, that could potentially solve these latency issues. Hardware-specific solutions as well as software optimizations would enable real-world applications of these layers in devices with low memory footprint.

3.5.3 Comparison with Other Compression Methods

As mentioned previously, this approach falls under the general area of network compression where several other techniques have been proposed. Compared to other approaches such as knowledge distillation or model pruning, we can note various benefits of our method. Unlike distillation, where a trained teacher model is typically required to train the student and needs to be trained first, parameterized hypercomplex layers have no such requirement, simplifying the training process and lowering its computational demands, as only one model is necessary. Moreover, training with distillation is often not as straightforward, since the *temperature* hyper-parameter T needs careful tuning, not to mention additional loss factors that are often necessary to adapt the student to the teacher. Balancing the distillation loss with the overall loss factor is also an issue that is not straightforward.

Following, in contrast to pruning methods, a significant difference is that the resulting models actually demonstrate reduced parameter sizes, rather than inducing sparsity by zeroing the pruning parameters, which retains the original model sizes. The reduced size allows for applications by devices with hardware limitations such as memory and storage. Simultaneously, some pruning approaches rely on a sparsity budget which determines the overall model compression. Setting an overly aggressive budget could potentially significantly lower overall performance, while a conservative one might not result in a meaningful compression. In this way, the benefit of these layers lies in their flexibility when setting the hyper-parameter n (which functions as a ”compression budget”) and can strike a balance between performance and compression. Adding to this, an additional limitation of model pruning presents itself in the inability of frameworks to fully take advantage of sparse tensors and operations, which ultimately does not offer an additional benefit at runtime. In contrast, our compressed models exhibit very similar running speeds compared to the baselines even at very high compression rates, as shown in Table 3.4.

In terms of results, to the best of our knowledge, no works exploring pruning techniques in the context of visual speech recognition have been published, therefore more studies are needed to accurately compare these two approaches. In the case of distillation, our models outperform methods such as [268], where the authors apply several iterations of distillation, which is a rather time-consuming process and requires an already trained model that performs well. In contrast, our method follows the straightforward standard training procedure for supervised learning. Overall, our results inspire confidence in the capabilities of hypercomplex layers for model compression as the accuracy trade-off is minor compared to the achieved reduction in model sizes, which does not overly affect runtime. From a theoretical, as well as a practical standpoint, hypercomplex layers demonstrate strong potential for practical visual as well as audio speech recognition and pose a viable alternative to other compression methods.

3.6 Conclusion

In this Chapter, we explored how Parameterized Hypercomplex Multiplication layers can be utilized to develop lightweight models for the task of visual speech recognition. These layers achieve sizable network compression through parameter savings by exploiting a sum of Kronecker products that retain the original layer dimensionality while sharing parameters, effectively shrinking the size of the network, depending on user demands. We proposed taking advantage of these layers by replacing standard network components in established architectures for the task of single word visual speech recognition. Extensive experimentation was conducted for several compression rates and evaluation was carried out on the largest public dataset of isolated English words. Our results showcase that when these layers are used in place of their standard counterparts in a VSR architecture, significant reduction in model size is achievable for a minor accuracy trade-off. When compared to other works in the literature, our models require fewer parameters to operate for comparable accuracy, especially for audio-only recognition. At the same time, our more lightweight models outperform other works with similar amounts of parameters. Future work includes exploring automated hyper-parameter tuning strategies as well as training and initialization schemes to further improve performance and compression.

Chapter 4

Utilizing Cost-Efficient Components

While model compression is a successful approach to reduce existing network sizes, in practice we find that deploying the final models offers little additional benefit in terms of running speeds. This observation stems from the absence of specialized hardware implementations that are able to take advantage of the compressed networks, especially in the case of PH layers that rely on the calculation of a Kronecker product, which might not be optimized. A similar effect can be noticed in the case of network pruning, where the induced sparsity is not exploited by the existing implementations and does not improve the running speeds of the resulting models since they maintain a high weight or neuron dimensionality (with large amounts of zeros). Specific software and hardware is required to fully utilize the compressed models with PH layers, as the required calculations involve a computation of a summation factor, which complicates memory requirements, especially with larger values of n that controls the amount of operands to be summed as well as the dimensions of the matrices that interact in the computations.

Considering the above, in this Chapter, we explore a different approach to model compression which encompasses utilizing components that achieve a two-fold effect, model compression as well as reduction in required computations. Our goal is to not only develop more compact models with reduced size, but also to enable more applications in real-world scenarios by deployment in devices with limited hardware capabilities. A pre-requisite for the latter is manageable computational overhead, which is typically measured in floating point operations and is used as an evaluation metric for networks in various computer vision tasks. In fact, as storage tends to be cheaper than available computation capacity, especially in cases of GPUs, developing lightweight architectures that can be effectively utilized by even CPUs is of paramount importance, as it can facilitate an even wider adoption and decouple the demand for large hardware clusters.

To that end, we develop lightweight architectures for visual speech recognition, by taking advantage of cost-efficient neural network modules that lower the overall computation costs. We design deep neural networks by utilizing cost-effective network components that take advantage of operations with low computational overhead. Our proposed models benefit from low sizes in terms of required parameters as well as reduced computational complexity, making them ideal for various practical applications. We conduct an extensive experimental analysis which showcases that our models feature greatly reduced hardware demands, without compromising their accuracy.

In summary, our contributions are the following:

- We employ *Ghost* modules in a unified word VSR architecture by replacing the standard convolutions in its components (visual feature extractor and sequence model) in order to reduce its overall computational overhead. Using Ghost modules, we further reduce the running costs of two established temporal convolution architectures that are used for sequence modeling, resulting in models that are even more lightweight than their standard versions and achieve comparable accuracy. The final architecture still performs very competitively compared to the original, while being less demanding in resources, measured in terms of model parameters and computational overhead.

- We also design a temporal block architecture, named *Partial Temporal Block*, that splits the input volume in two parts and applies separate operations in each part. Using this component as a building block, we follow three methods from the literature and develop highly efficient temporal convolution networks for sequence classification aimed at applications with very low resources.

- We perform extensive experiments on the largest publicly-available dataset for English words and our results showcase strong visual speech recognition performance. Simultaneously, our proposed models are practical in terms of hardware demands, as showcased in a detailed ablative analysis, allowing for several applications by devices with varying computation capabilities.

4.1 Methodology

4.1.1 Ghost Modules

Ghost modules were proposed in [275] as a component that takes advantage of "cheap operations" to reduce its computation cost compared to the standard convolution layer. A Ghost module achieves low resource overhead in two steps. First, a regular 1×1 convolution generates a set of feature maps from the input. A fixed ratio determines the number of channels in the generated feature maps, controlling the computation cost of the component. Typically, the ratio is set to 0.5 meaning that channels in the produced feature maps equal half of the input volume's channels.

A "cheap operation" utilizes these intermediate feature maps to produce an additional set with the same channel size. The role of the cheap operation can be undertaken by any lightweight function; in the Ghost module, a depth-wise convolution with a kernel size of 3×3 is used. This convolution operates on each filter and processes the spatial information it contains, while preserving the amount of channels. Finally, the two distinct feature maps are concatenated along the channel dimension, meaning that the output volume matches the input's channels.

Compared to the standard convolution operation, this formulation reduces the total amount of computation required since the initial 1×1 convolution generates a feature map with fewer channels and the depth-wise operation, which is much cheaper computationally, is also applied on this volume rather than the whole input. By preserving the original output size of a convolution layer, a Ghost module can act as a drop-in replacement for that layer to reduce the computational overhead in a network architecture. The operations of the Ghost module can be summarized as:

$$X_1 = non - linearity(normalization(Conv_{1\times 1}(X))) \tag{4.1}$$

$$X_2 = non - linearity(normalization(DConv_{3\times 3}(X_1))) \tag{4.2}$$

$$Out = concatenation([X_1, X_2]), \tag{4.3}$$

where X refers to the input volume and $DWConv$ to the depth-wise convolution.

In the Ghost module, Batch Normalization is used and non-linearity is handled by a rectified linear unit (ReLU) layer, although any function can be used (e.g., ReLU6, SiLU, etc.). Its hyper-parameters include input and output channels, cheap operation (depth-wise convolution) kernel size as well as a factor that controls the amount of intermediate channels produced by the point-wise convolution which ultimately governs the amount of computation reduction. The Ghost module is depicted in Figure 4.1(a).

Computational Complexity of Ghost Modules

Assuming a regular 2D convolution with a kernel size of k, C_i input and C_o output channels respectively, when applied to an input volume with spatial dimensions of $H_i \times W_i$, it produces an output volume of $H_o \times W_o \times C_o$ with a computational cost of:

$$H_o \cdot W_o \cdot C_i \cdot C_o \cdot k \cdot k. \tag{4.4}$$

In contrast, a depth-wise convolution uses one filter per input channel (also called *input depth*) and costs:

$$H_o \cdot W_o \cdot C_i \cdot k \cdot k. \tag{4.5}$$

From the above two equations, it can be seen that the depth-wise convolution reduces the computational costs when replacing a standard convolution with the same hyper-parameters.

The depth-wise convolution only spatially filters the input channels without combining them, unlike the standard convolution, therefore, it is typically followed by an additional layer that performs this linear combination through a regular convolution using a kernel size of 1×1, which is also known as a *point-wise convolution*, with a computational cost of:

$$H \cdot W \cdot C_i \cdot C_o, \tag{4.6}$$

for an input feature map of dimensions $H \times W \times C_i$, that will be transformed into an output feature map of dimensions $H \times W \times C_o$. Parameter-wise, the regular convolution requires $C_i \cdot C_o \cdot k \cdot k$ trainable parameters, whereas depth-wise and point-wise require $C_i \cdot k \cdot k$ and $C_i \cdot C_o$, respectively. This combination is called a depth-wise separable convolution and is the cornerstone of several lightweight model designs, e.g., [61, 310, 81] etc.

The computation costs associated with the Ghost module using a cheap operation with a kernel size of k on an input volume with C_i channels, producing an output of dimensions $H_o \times W_o \times C_o$ are:

$$\frac{C_o}{s}(H_o \cdot W_o \cdot C_i \cdot k \cdot k) + (s-1)\frac{C_o}{s}(H_o \cdot W_o \cdot k \cdot k) = \frac{C_o}{s}(H_o \cdot W_o \cdot k \cdot k)(C_i + (s-1)). \quad (4.7)$$

When substituting a regular convolution layer with the Ghost module using a cheap operation with the same kernel size (e.g. 3×3), the theoretical speed-up that can be achieved is:

$$\frac{H_o \cdot W_o \cdot C_i \cdot C_o \cdot k \cdot k}{\frac{C_o}{s}(H_o \cdot W_o \cdot k \cdot k)(C_i + (s-1))} \approx \frac{s \cdot C_i}{C_i + s - 1} \approx s. \quad (4.8)$$

In these two equations, s represents the *ratio* of the block, which practically controls the output of the initial 1×1 convolution as well as that of the subsequent cheap operation, since at the end of the block, the outputs are concatenated, matching the original output dimension.

4.1.2 Ghost Module V2

A drawback related to the representation capabilities of the Ghost module arises from the fact that the initial 1×1 convolution reduces the feature map channel dimensionality (to half) in order to keep the costs of the module low. Subsequently, the second (3×3 depth-wise) convolution operates on a sub-set of the input feature map and might miss some spatial relationships that would otherwise be captured by operating on the full input volume. Since half of the final feature map in the output of the Ghost module is produced from the 1×1 convolution without any spatial interaction between the pixels, the performance of the module is hampered. To alleviate this weakness, the authors of [311] propose an enhancement called *DFC attention* which aims to exploit long-range spatial information, augmenting the Ghost module's intermediate features with richer representations that were lost by the original design.

The DFC attention module aims to improve representation capabilities by exploiting spatial information through attention. It utilizes two fully-connected layers which are applied to the input features in a sequential manner, spanning both the vertical and horizontal directions and aggregating the features in each direction. By operating sequentially on the two different directions instead of simultaneously on a square area, the computational complexity of the attention module is kept low.

Initially, the input feature map is spatially down-sampled both vertically and horizontally with a pooling operation which shrinks the spatial dimensions by half. Since the subsequent layers operate on feature maps of smaller size, the required computations are reduced significantly. Then, the fully-connected layers are applied in a sequential manner, first the vertical (column-wise) layer, followed by the horizontal (row-wise) layer. Finally, the produced feature map passes through a function to scale its values in the $(0, 1)$ range, producing an attention map and an up-sampling operation restores the original spatial dimensions. The following equations show the DFC module's operations on an input volume X:

$$X_1 = pooling(X) \tag{4.9}$$

$$X_2 = normalization(Conv_{1\times1}(X_1)) \tag{4.10}$$

$$X_3 = normalization(Conv_{1\times5}(X_2)) \tag{4.11}$$

$$X_4 = normalization(Conv_{5\times1}(X_3)) \tag{4.12}$$

$$X_5 = non - linearity(X_4) \tag{4.13}$$

The DFC module is implemented with a pooling operation that averages the values, while the non-linearity at the end is handled by a Sigmoid function. Adding DFC attention to the Ghost module incurs an increase in parameter size due to the additional convolutions but only a slightly higher computation cost in FLOPs. More specifically, this component's complexity is $\mathcal{O}(H^2 \cdot W + H \cdot W^2)$, where H and W are the dimensions of the weight matrices for each fully-connected layer that operates in a single dimension of a feature map with spatial dimensions of $H \times W$. By substituting the fully-connected layers with two depth-wise convolutions of kernel sizes $1 \times K_H$ and $K_W \times 1$, the complexity further reduces to $\mathcal{O}(K_H \cdot H \cdot W + K_W \cdot H \cdot W)$. In contrast, operating on a square patch requires a complexity of $\mathcal{O}(H^2 \times W^2)$, which is prohibitively expensive since it scales quadratically as it relies on the dimensions of the intermediate feature maps. The Ghost V2 module which includes the original with the addition of DFC attention is depicted in Figure 4.1(c).

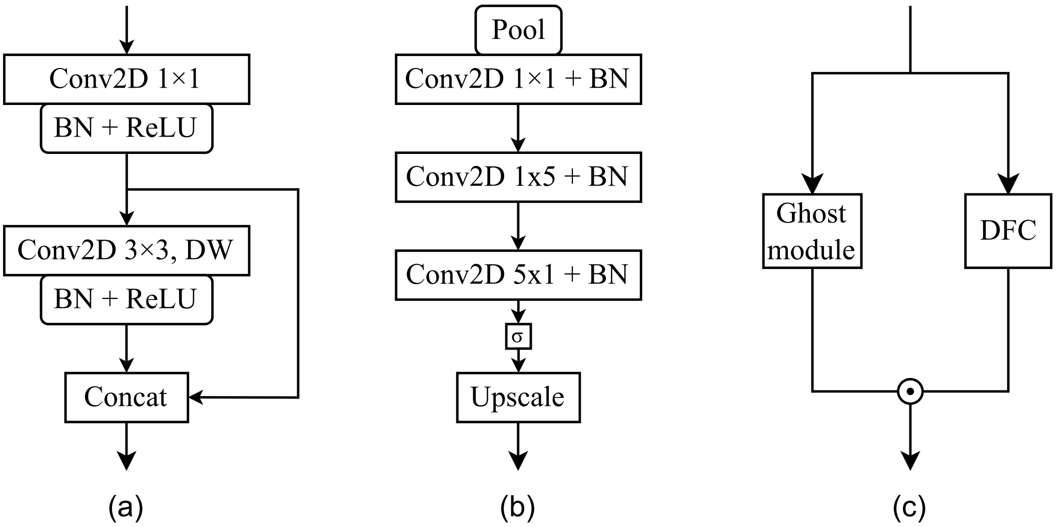

Figure 4.1: Illustration of the Ghost modules. BN indicates the *Batch Normalization* operation, ReLU indicates the *Rectified Linear Unit* function, DW refers to the *depth-wise convolution*, while σ is the logistic Sigmoid and $\odot$ is the element multiplication sign. (a) Original Ghost Module. (b) DFC attention. (c) Ghost Module with DFC attention.

4.1.3 Partial Temporal Block

Reducing the size of the input feature map and operating on the result is an effective approach to reduce the computational overhead of a network component that has been followed by several lightweight networks (e.g., [61, 310]). Within a network block, using the initial layer to reduce the channel dimension of an input volume and applying the subsequent layers in the smaller output allows controlling the amount of calculations and enables the development of lightweight network components with low operating costs. An additional operation, commonly the final one in a block, restores the channel dimension to match that of the input, usually in order to facilitate a residual connection. This design is known as a *bottleneck*, since the intermediate feature maps have a lower number of channels.

A similar approach [185, 312, 313] splits the input feature map across the channel dimension in two parts according to a fixed ratio and applies two separate branches of computation, one in each part. The operations in either branch can have any form, for instance, in [313] a regular convolution followed by two point-wise layers is applied on one branch, while the second branch leaves the input unchanged. To form the output, the results of each separate branch are merged along the channel dimension via concatenation.

Inspired by the practicality and results of methods following this paradigm (e.g., [185, 313]), we design the *Partial Temporal Block*, a 1D equivalent which follows the same principle. Our block allows for a wide network design flexibility as it can be tailored to each specific application constraints (e.g., hardware capabilities, dataset availability and size) and can even be part of a search space, in order to automatically obtain the most optimal setup (e.g., via a NAS method), depending on the problem. For an input volume X, the operations of the partial block can be summarized as:

$$\begin{aligned} X_1, X_2 = \mathit{channel} - \mathit{split}(X) \\ X_3 = F(X_1) \\ X_4 = G(X_2) \\ X_c = \mathit{concatenation}([X_3, X_4]) \\ X_{out} = X_c + X \;, \end{aligned} \tag{4.14}$$

where the channel split operation divides the input in two parts along the channel dimension according to a fixed ratio, F and G can be any type of operation, including sequences of layers and the final concatenation merges the output of each branch in the channel dimension. A skip connection adding the input is also included to facilitate easier training of deep architectures. The block architecture is depicted in Figure 4.2 (a).

Furthermore, following the designs of [185] and [313], we design two additional lightweight partial temporal blocks that require few parameters and have very low computational overhead in terms of FLOPs. Their operations as used within our proposed block are depicted in Figure 4.2 (b) and (c). We note that, for the *ShuffleNet* [185] block design, a channel mixing operation is added at the very end (after concatenation and addition), while for the *FasterNet* [313] block design, the MLP network is applied after concatenation of the branches but before adding the input to the concatenated tensor via the skip connection.

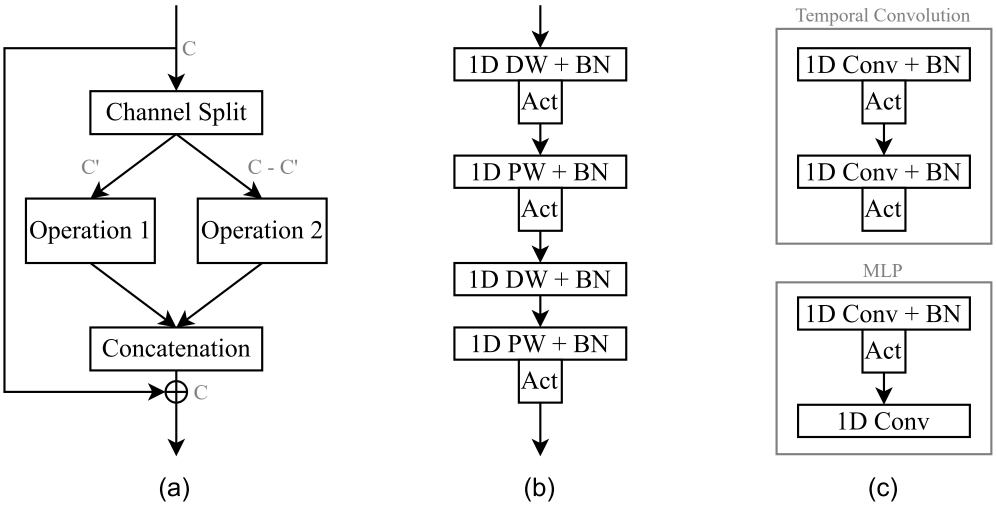

Figure 4.2: Block designs used in the proposed *Partial Temporal Block*. (a) Block architecture, where C represents the amount of channels of the input volume to each component and C' is determined by a hyper-parameter. (b) *ShuffleNet* block architecture. (c) *FasterNet* block components. “DW” and “PW” indicate *depth-wise* and *point-wise* convolutions. “BN” is the *Batch Normalization* layer and “Act” is an activation function (e.g., ReLU).

4.2 Proposed Model

The architecture of our proposed model follows the two-step design (feature extraction and sequence modeling) paradigm for the VSR task and its structure is depicted in Figure 4.3. For feature extraction and sequence modeling, we employ lightweight models based on Ghost modules to greatly reduce network overhead and keep computations at an affordable level. In addition, we utilize our proposed partial temporal blocks in TCN-based architectures building practical models suitable for scenarios with limited resources or low-powered hardware. At the end of the entire network, a fully-connected layer maps the output channels of the sequential model to the classes of the dataset (500 in our case) and a Softmax layer predicts the single spoken word from the video input.

4.2.1 Feature Extraction Model

3D spatio-temporal sub-network. Due to the dimensionality of the input sequence, which consists of a 3D volume $T \times H \times W$ (T frames of dimensions $H \times W$), at the

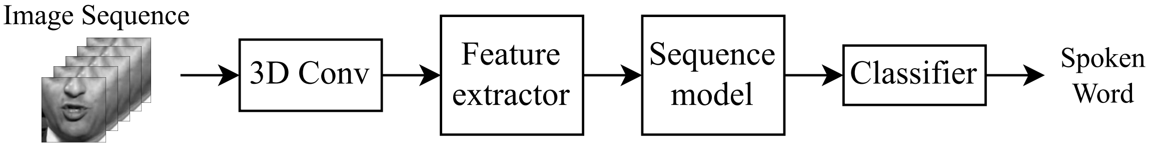

Figure 4.3: Overview of the architecture used for visual speech recognition. We add Ghost modules in the feature extraction network and experiment with a customized architecture. For the sequence model, we employ TCN variants with Ghost modules and our proposed Partial Temporal Block. The Softmax function is used as the classifier. The overall system outputs a spoken word.

beginning of the feature extraction networks we add a layer stack composed of a 3D convolution, batch normalization, non-linear activation and pooling layers. A 3D convolution's filters operate and move along all three dimensions and the same is true for the pooling operation within this sub-network. In our architecture, the kernel size of this convolution is set to $5 \times 7 \times 7$, with 5 corresponding to the time dimension and the output has 64 channels. The form of operation that sub-samples the spatial information selected is average pooling. This sub-network is a widely-used choice in the literature as its computational overhead with regards to its performance benefit is quite low due to the small dimensions of the produced feature map.

Ghost-ResNet. Next in the architecture is a CNN visual feature extractor which outputs high-level representations of the input sequence, containing spatial information about the mouth area and both modules comprise the first stage of the overall end-to-end model. We design a customized residual network derived from the original [31] that is widely adopted by other works in the visual speech recognition literature. It follows the same 18 layer structure organized in four stages, where the output channels double at each stage, starting from 64. In each stage, the base network uses two residual blocks, each containing two stacks of 2D convolution, normalization and non-linear layers and the input is connected with the output through a skip connection. Since the later blocks have a high dimensionality, we employ the original Ghost module and the improved variant with DFC attention to improve the networks' efficiency. Without changing its structure, the lightweight Ghost modules are used in place of the standard convolutions in each residual block, greatly diminishing the computation overhead of the entire network. A comparison showcasing the parameter and computation reductions obtained by utilizing the Ghost modules is provided in a later Section.

4.2.2 Sequence Modeling Networks

The extracted features that are obtained by the convolutional networks in the first stage are fed into a sequential model which serves as the second stage of the overall visual speech recognition network. The model's architecture is derived from the *Temporal Convolution Network* that is widely adopted in the literature for visual speech recognition, replacing older recurrent models such as the LSTM or RNN, due to its lower computational overhead and smoother training process that offer state-of-the-art performance [155].

Ghost TCNs. In this Chapter, we use the architectures of *Multi-Scale* (MS) and *Densely-Connected* (DC) TCNs, as proposed in [110] and [162], respectively. Both of these architectures share some common characteristics, including the use of 1D standard convolutions and are designed with four temporal blocks, with each block using three parallel connections with two stacks of 1D convolution, batch normalization and non-linear activation layers. While the MS-TCN uses a dilation rate that is scaled according to the block index and shared by all convolutions within each block, the DC-TCN opts for fixed dilation rates for each convolution in all blocks. Due to the high costs of these models, we replace both layer stacks in each temporal block with our proposed lightweight layers using Ghost modules instead, to produce more efficient networks with reduced parameters and computation costs. To adapt the Ghost modules to 1D feature maps, we apply small modifications to the convolution and normalization layers. In these networks, as the inputs and the generated feature maps are all one-dimensional, the previously-mentioned DFC attention module cannot be utilized since it operates on two dimensions, therefore we only adopt the original Ghost modules without the DFC attention mechanism. We employ both TCN architectures in our experiments to evaluate the performance of Ghost modules in distinct network setups with different model size and performance.

Partial TCNs. Moreover, using our proposed Partial Temporal Block as the basic component in a TCN architecture, we design lightweight sequence-to-sequence models with very low computation requirements that are ideal for low resource scenarios. As a baseline model, we employ the standard Temporal Convolution layer [155] as the core of our block, applied in one branch. This layer uses a sequence of 1D causal convolutions with batch normalization and non-linear activation functions, repeated twice. The other branch uses no operations, greatly reducing the computational overhead of the block. The overall architecture is comprised of four stages, where each stage is one Partial Temporal Block with an increasing dilation rate that is used in the non-point-wise convolutions. This way, the entire network is especially lightweight in terms of hardware requirements (a detailed parameter analysis will be provided below). Moreover, we design two additional lightweight TCNs based on [185] and [313] (see Figure 4.2 (b) and (c)) that are also comprised of four main stages and have very low computational overhead measured in model parameters and FLOPs.

4.3 Experiments

4.3.1 Training Setup

All models are trained from randomly initialized weights on the LRW training set (see Chapter 2). We train for a total of 80 epochs with AdamW [309], using a batch size of 32. An initial learning rate of 0.0005 with a cosine annealing schedule is used, without any warming up period. To prevent over-fitting, weights are decayed by 0.0001 and dropout is added to the TCN layers for all models with a probability of 0.2. During training, spatial cropping and flipping are randomly applied, as well as MixUp [66] and variable length augmentation [110]. After each epoch, the model's weights are evaluated on the LRW validation set and the best performing checkpoints are saved to be evaluated on the test set after finishing all epochs.

4.3.2 Results & Discussion

Our proposed models are evaluated in the LRW test set and a comparison with other models from the literature is provided in Table 4.1. The metric used to evaluate the methods is word accuracy, measured as a percentage of correct word predictions. We also include size and model complexity measurements, more specifically, the amounts of total network parameters and Floating Point OPerations (FLOPs), as these values are useful to gauge the overall practicality of the methods when considering several applications. More detailed, per-model overviews are provided in Tables 4.5 and 4.6. As in the previous Chapter, *PyTorch* was used to develop all models and *torchinfo* was employed to obtain measurements regarding model size and complexity.

Our experimental evaluation showcases that utilizing the Ghost modules on each

Table 4.1: Experimental evaluation on the LRW test set and comparison with recent methods from the literature. Results are sorted by computational complexity. "FLOPs" refers to Floating Point OPerations, "(G)" indicates that the model is using Ghost modules. Models proposed in this Chapter are highlighted.

Method (Models used)	**FLOPs ($\times 10^9$)**	**Parameters ($\times 10^6$)**	**Acc. ↑ (%)**
ShuffleNet v2 (1×) + MS-TCN [268]	2.23	28.8	85.5
ResNet + MS-TCN [110]	10.31	36.4	85.3
MobiVSR-1 [266]	11.0	4.50	72.2
ResNet + DC-TCN [162]	10.64	52.54	88.3
ResNet + DC-TCN [148]	10.64	52.54	90.4
ResNet + 3×Bi-GRU [69]	10.54	59.5	88.4
ResNet + 2×Bi-LSTM [64]	10.24	50.07	88.7
DenseNet + 3×Bi-GRU [39]	26.12	14.31	83.0
ResNet (G) + MS-TCN	2.31	25.58	87.16
ResNet (G) + MS-TCN (G)	1.78	14.29	86.24
ResNet (G) + DC-TCN	2.67	41.77	88.72
ResNet (G) + DC-TCN (G)	2.03	27.04	88.17
ResNet (G V2) + MS-TCN	2.53	26.86	88.42
ResNet (G V2) + MS-TCN (G)	2.00	15.57	86.16
ResNet (G V2) + DC-TCN	2.88	43.05	88.52
ResNet (G V2) + DC-TCN (G)	2.25	28.32	87.98
ResNet + MS-TCN (G)	9.76	25.06	87.87
ResNet + DC-TCN (G)	10.01	37.81	89.10

component of the architecture (feature extraction or sequence model) bestows a noticeable improvement in computation requirements, since the cheap operations in the Ghost module are much more efficient than the regular convolutions in the original networks. In addition, we gain significant savings in model sizes by lowering parameter counts leading to more compact final models, allowing for applications in a broader range of devices as network size is essential for energy savings due to memory access costs.

Simultaneously, a minor accuracy drop occurs arguably due to the reduced representation capabilities of the Ghost module, which is a drawback also mentioned in [311]. Nevertheless, the residual convolutional network [31] equipped with Ghost modules still performs rather well, being highly competitive with larger networks and it surpasses other works while being more lightweight in terms of both parameter size and computation. Employing Ghost V2 modules in the residual architecture causes an increase in model parameters due to the design of DFC attention which uses two additional convolution layers (see Equation 4.9). However, this added amount is offset when combined with a TCN variant that also uses the Ghost module as its building block, so the overall parameter count drops and is still lower than the baseline network.

The Ghost V2 module provides a minor accuracy improvement when combined with a MS-TCN and with a DC-TCN with Ghost modules, falling behind the original Ghost module in other setups. The improved performance over the original Ghost module can be explained by the larger network size, due to the design of this block which is less efficient in terms of model size and FLOPs. We observe that when using a densely-connected (DC-TCN) [162] sequence model, it outperforms the original only when Ghost modules are also used in the TCN. These results indicate that due to the low spatial dimension of the input, the DFC attention is underutilized in some cases and provides a benefit when combined with a highly efficient sequential model.

A more evident benefit of these modules is the considerable reduction in FLOP count when used in the feature extraction network, which can be more preferable than parameter savings in some scenarios (e.g., for hardware with adequate memory but a low-power processing unit). Depending on the available resources of a device, a model where only one component utilizes Ghost modules can be used to suit the application.

When using Ghost modules in the Temporal Convolution Network variants, the already lightweight TCNs are made even more compact by further reducing their size and FLOPs. More concretely, when replacing the standard convolution layers with Ghost modules in the MS-TCN model, we notice a reduction in overall GFLOPs by 0.53, while the parameter count drops by about 11.2 million. Similarly, in the TCN variant with dense connections, using Ghost modules reduces the total parameter count by around 35%, while the computation cost drops by 0.6 GFLOPs, yet the model still maintains a high recognition accuracy. Table 4.5 shows a more comprehensive comparison of hardware requirements per network component when using the Ghost modules.

4.3.3 Partial TCNs

We evaluate the Temporal Convolution Network variants on LRW when using our proposed Partial Temporal Block as their core component. For brevity, we name these network variants as *Partial TCNs (P-TCN)*. As mentioned previously (see Subsection 4.1.3), we employ three architectures from the literature ([155, 185, 313]) within our block. When combined with the residual network with Ghost modules, highly lightweight models can be produced that still achieve strong accuracy. The results are shown in Table 4.2.

Table 4.2: Experimental evaluation on the LRW test set for our methods using the proposed Partial Temporal Block. In these experiments, the kernel size for all convolution operations that are not *point-wise* is shown. “FLOPs” refers to Floating Point OPerations and parameters are measured in millions ($\times 10^6$). “(G)” indicates that the model is using Ghost modules.

Method	**FLOPs ($\times 10^9$)**	**Parameters**	**Accuracy ↑ (%)**
ResNet + P-TCN (Temporal block, k=7)	9.59	22.80	85.29
ResNet (G) + P-TCN (Temporal block, k=5)	3.27	11.22	83.05
ResNet + P-TCN (ShuffleNet block, k=5)	9.20	13.85	84.44
ResNet (G) + P-TCN (ShuffleNet block, k=3)	3.05	5.50	81.93
ResNet + P-TCN (FasterNet block, k=3)	9.36	20.56	87.03
ResNet (G) + P-TCN (FasterNet block, k=3)	3.20	12.23	86.48

4.4 Ablation Studies

We perform an ablation analysis experimenting with the channel ratio used in the partial temporal block within the TCN-based sequence models. This parameter controls the balance between the channels of each computation branch when splitting the input feature map. In this experiment, we use the *FasterNet* [313] formulation (Figure 4.2 (c)) as the core of our Partial Temporal Block, since it outperforms the other two methods. In this setup, one branch has no calculations and therefore no computations, meaning that the ratio controls the amount of calculations per block; a higher ratio provides more channels to the resource-intensive branch, increasing overall performance at the cost of overhead and vice-versa. For feature extraction, we employ two CNNs: the standard 18-layer residual model [31] and the lightweight version with Ghost modules. We train all models with the procedure mentioned in Subsection 4.3.1 and the results are shown in Table 4.3.

Table 4.3: Ablation analysis on the channel ratio in the partial block. Evaluation is performed in the LRW test set. “FLOPs” refers to Floating Point OPerations ($\times 10^9$). Parameters are shown in millions ($\times 10^6$).

Method	ratio	FLOPs	Param.	Accuracy ↑ (%)
ResNet + P-TCN (FasterNet block)	0.25	9.30	18.9	85.21
	0.5	9.32	19.5	85.37
	0.75	9.36	20.5	87.03
ResNet (G) + P-TCN (FasterNet block)	0.25	3.14	10.6	82.30
	0.5	3.16	11.2	85.40
	0.75	3.20	12.2	86.48

Using a higher ratio, as one would expect, leads to greater overall recognition accuracy, since, after splitting the input tensor, the branch that performs calculations receives a larger volume and operates on a higher percentage of the input, exploiting information from more channels. This is accompanied by a slightly higher FLOP and parameter count of the TCN-based models, which is not significant, especially when using the Ghost module, which greatly shrinks the overall costs. Increasing the ratio from 0.25 to 0.75 only adds 0.05 GFLOPs and 1.6 million parameters while raising accuracy by 1.82% to 4.18%, depending on the feature extraction model. The highest ratio (0.75) allows the CNN with Ghost modules to achieve large accuracy gains, surpassing several networks that are much more expensive.

We also perform an additional experiment where we increase the kernel size of the convolutions in each block, in order to provide the network with a larger effective receptive field and tabulate the results in Table 4.4. For this experiment, we evaluate the Temporal [155] and *ShuffleNet* [185] architectures in our block and set the ratio to 0.75 as it offers the best performance for a negligible impact in computation overhead. As before, we keep the previous training settings.

Generally, using a larger kernel size improves recognition accuracy while slightly raising computation overhead due to the amount of calculations required by the larger kernel. We note however, that this does not apply to all cases, for instance when using

Table 4.4: Ablation analysis on the kernel size used in the branch that performs operations in the Partial Temporal Block. Evaluation is performed in the LRW test set. "FLOPs" refers to Floating Point OPerations ($\times 10^9$). Parameters are shown in millions ($\times 10^6$). "(G)" indicates that the model is using Ghost modules.

Method	kernel size	FLOPs	Param.	Acc. ↑ (%)
ResNet + Partial TCN (Temporal block)	3	9.30	16.31	82.75
	5	9.43	19.55	83.78
	7	9.59	22.80	85.29
	9	9.80	26.04	84.10
ResNet (G) + Partial TCN (Temporal block)	3	3.14	7.98	81.19
	5	3.27	11.22	83.05
	7	3.44	14.46	82.64
	9	3.64	17.71	83.07
ResNet + Partial TCN (ShuffleNet block)	3	9.20	13.84	83.65
	5	9.20	13.85	84.44
	7	9.20	13.86	84.13
	9	9.20	13.87	83.37
ResNet (G) + Partial TCN (ShuffleNet block)	3	3.05	5.50	81.93
	5	3.05	5.52	81.92
	7	3.05	5.53	81.57
	9	3.05	5.54	81.68

the ShuffleNet block a larger kernel size than 5 (e.g., $7, 9$) does not improve accuracy and in fact hampers performance when the residual network with Ghost modules is used. For a more clear overview of the complexity that each component adds to the overall measurements, the reader is referred to Section 4.4.1.

As for the TCN using the Temporal block [155], it scales better with a larger kernel size, improving its performance, compared to the ShuffleNet block, however, this network's FLOPs and parameters increase at a much higher rate since it uses regular convolutions. The same diminishing effect in accuracy gains is noticed for the largest kernel sizes. Similar to the results shown in previous tables (e.g., Table 4.1), when Ghost modules are used in the convolutional feature extraction network, significant reductions in computation and sizes are gained, while the final accuracy suffers slightly.

4.4.1 Parameter Analysis

In addition, we tabulate all measurements related to network size and complexity for all proposed architectures in this Chapter in Tables 4.5 and 4.6, showcasing the efficiency gained by using Ghost modules and our proposed Partial Temporal Block when designing lightweight networks.

Table 4.5: Detailed parameter analysis per network component. The proposed Partial TCN using the *FasterNet* block is also added for comparison.

Model	**FLOPs** ($\times 10^9$)	**Parameters** ($\times 10^6$)
ResNet	8.29	11.16
ResNet (Ghost module)	0.31 (-96%)	0.39 (-96%)
ResNet (Ghost V2 module)	0.52 (-93%)	1.67 (-85%)
MS-TCN	1.12	25.17
MS-TCN (Ghost module)	0.59 (-47%)	13.88 (-44%)
DC-TCN	1.47	41.36
DC-TCN (Ghost module)	0.84 (-42%)	26.63 (-35%)
Partial TCN (FasterNet block, 0.25 ratio)	0.12	7.80
Partial TCN (FasterNet block, 0.5 ratio)	0.15	8.39
Partial TCN (FasterNet block, 0.75 ratio)	0.18	9.38

As mentioned previously, the feature extraction networks that employ Ghost modules achieve significant savings in both size and computation compared to the standard 18-layer residual [31] architecture. In both cases of Ghost modules, computation complexity reduces by 93% to 96%, while parameter count drops by 85% to 96%. Due to the added computation, the architecture using the Ghost V2 module requires slightly more FLOPs and parameters than the original Ghost module. For the sequence models, a more modest reduction in size and overhead (up to 44% and 47% respectively, in the case of the MS-TCN) is achieved, since the original architectures are already quite lightweight. Due to the added complexity of the DC-TCN network, the reductions are lower than in the other architectures, but still significant overall.

Finally, the TCN-based architectures using our proposed Partial Temporal Block become even more lightweight regardless of block design. These variants require a fraction of resources compared to all other architectures and scale favorably with the ratio and kernel size hyper-parameters. Overall, the FasterNet [313] block design is the better choice for our proposed Partial Temporal Block, compared to the other two, as it maintains a very low FLOP and parameter overhead across all channel ratios and with the highest ratio amount (0.75) it outperforms several larger models as well as the other Partial TCN variants. The ShuffleNet [185] design also maintains extremely low FLOP and parameter measurements but falls behind the other designs in performance mainly due to the rather low parameter count. When combined with the Residual network with Ghost modules, it forms a highly compact overall model at around 5.5 million parameters that is more suitable for hardware with very low capabilities, with corresponding performance. The Temporal block design represents a middle ground between the two previous architectures, surpassing the ShuffleNet design, while also maintaining low FLOPs as we increase the kernel size, but this block has high parameter counts.

Table 4.6: Size and complexity analysis of the TCN variants using our proposed Partial Temporal Block for different core components and kernel sizes. Evaluation is performed in the LRW test set. "FLOPs" refers to Floating Point OPerations ($\times 10^9$), while parameters are measured in millions ($\times 10^6$). The ratio for the channel split used in all models in this table is set to 0.75 as it is the most resource-intensive amount.

	Block	**ShuffleNet**	**Temporal**
Kernel size			
3	FLOPs	0.34	0.12
	Parameters	1.20	3.80
5	FLOPs	0.35	0.25
	Parameters	1.20	6.18
7	FLOPs	0.35	0.42
	Parameters	1.21	8.52
9	FLOPs	0.36	0.62
	Parameters	1.21	10.87

4.4.2 Limitations

A current drawback of the Ghost V2 module lies in the DFC attention and its design which employs two convolutions in two directions (vertical and horizontal). This prevents its exploitation by the temporal networks which utilize 1D convolutions and for this reason in our models its use is limited in the residual convolutional architecture which serves as a feature extractor. A possible explanation is that the DFC attention was originally designed for images of higher dimensions (224×224) and its use is sub-optimal in out architecture due to the fact that the 3D convolution and pooling block at the beginning of the overall model reduce the spatial dimensions of the feature map. The additional down-sampling (see Section 4.1.1, Equation 4.9) of the (already low-dimension) feature map removes much of the information contained and hinders the module's ability to exploit it, however the added computations and network capacity help recover some performance as seen when combined with a TCN that also uses Ghost modules. We believe that removing the pooling operations altogether could possibly improve the overall performance and plan on investigating this in the future.

Another limitation that should also be mentioned can be observed in Table 4.4, where we increase the size of the convolution kernels that are used in each Partial Temporal Block. A larger kernel size has a greater receptive field, since more neighboring locations of the feature map are taken into account during calculations. In the case of sequential data (i.e., also in our case), this translates into a wider ”view” of the sequence time-steps. It is expected that as the receptive field increases, so does performance, with larger kernels offering more benefits, since more information is taken into account. However, in our experiments this is not the case, for example when using the ShuffleNet block, a kernel size greater than 5 does not lead to further accuracy gains. We believe this to be due to the dilation amount used by the convolutions in the TCN architecture, which increases at every stage of the network. At the later stages of the TCN, the high amounts of dilation may cause the convolution layers to miss significant short-term information from the sequence as they become influenced by frames that are further away in the sequence that potentially contain information that is not relevant to the current time-step and acts as noise. It is also possible that the extremely low parameter count of this architecture is another limiting factor that prevents learning. An experimental validation of the above claims as well as tweaks in the architecture to recover performance are also left as future work.

4.5 Conclusion

In this Chapter, we proposed taking advantage of low-cost components to develop lightweight architectures for practical visual speech recognition (VSR) applications. Using the recently proposed Ghost modules where an amount of the channels within are calculated with cost-efficient operations, we developed low-resource models for VSR of isolated words. We replaced the standard convolution operations with Ghost modules in the visual extraction and sequence modeling networks creating compact and efficient alternatives that showcase significantly lowered computational resource requirements. Their reduced overhead enables a multitude of applications in several scenarios where speed of operation is critical and hardware resources are constrained. Evaluation on the largest single word speech recognition dataset showed that our models outperform other lightweight architectures while demanding fewer computational resources measured in FLOPs. Simultaneously, the achieved accuracy of the models is competitive with other architectures that are much larger in terms of model size and complexity. Moreover, we proposed a component called ”Partial Temporal Block” for building ultra-lightweight sequential models intended for devices with very limited hardware capabilities, such as IoT and edge devices. This block splits the computation path in two branches and can be customized to fit each use case according to the task and available resources.

Future work includes addressing the weaknesses outlined in this Chapter, i.e., taking advantage of DFC attention via architectural tuning and addressing the lower performance of the larger kernel size convolutions when used with our proposed Partial Temporal Block. We also intend to expand the block's capabilities by exploring automated techniques for optimal operation selection, or by introducing other efficient channel attention methods to increase performance. Finally, specialized training strategies exploiting the latest augmentation and weight averaging approaches are also planned.

Chapter 5

Designing Practical Architectures

The methods presented in the previous two Chapters revolve around reducing an existing network's hardware demands, measured by requirements on system resources such as available memory capacity, processing speed, storage space, etc. So far, the architectures that were employed were based on established models from the published VSR literature with high recognition rates. A common outcome of our experiments is lower performance, which is a direct result of reducing the representational capacity of the initial networks. We observe that adopting models from the VSR literature without considering their architecture (layers, components and overall design) is a sub-optimal strategy from a practical perspective, since these networks were designed to improve recognition performance without any hardware constraints. As a result, the employed networks might not be well-adapted to a strict resource budget, which is implicitly enforced by using lightweight components and this is another contributing factor for the lowered recognition rates achieved by the final networks. Using or developing specialized lightweight architectures can potentially mitigate this performance-to-complexity compromise, achieving higher recognition rates.

Following this rationale, in this Chapter, we design lightweight architectures that still perform competitively with other, more cumbersome ones while being much less demanding on system resources. To that end, we develop end-to-end models for word VSR that are practical in terms of model size and computation complexity measured in parameter counts and Floating Point OPerations (FLOPs) respectively. We explore practical, low-cost networks for feature extraction and sequence modeling by analyzing their hardware overhead and recognition performance, adopting the most effective components in an end-to-end architecture. Our proposed models have low demands in hardware resources while achieving high word recognition accuracy and can be deployed in a wide range of devices to cover more applications and use cases in real-life scenarios. Extensive experimentation on the largest publicly available corpus for word-level VSR without using any audio data showcases the effectiveness of our compact models in visual speech recognition of isolated words.

Our contributions are three-fold:

- We explore several lightweight convolutional neural networks from the image classification literature as feature extractors and benchmark their performance when used in a VSR architecture for word recognition. Since this component is responsible for a significant amount of computation of the overall end-to-end architecture, by selecting a robust yet compact model we can achieve savings in model size and computational complexity, without severely compromising performance.

- We apply the same process to the sequential model that is used to further transform the extracted features, as this component is crucial for strong VSR performance. By adapting various network building blocks to an equivalent one-dimensional causal design, we replace the standard block within a vanilla temporal convolution network to improve its sequence modeling capabilities while keeping size and complexity at affordable levels.

- With insights gathered from the above experiments, we design lightweight yet powerful unified architectures and validate their capabilities by performing several experiments and ablation studies on the largest publicly-available dataset for recognition of English words without using any additional training data.

5.1 Model Structure

We employ the ubiquitous multi-step design that is an established approach with proven results and is adopted by virtually all works in the published VSR literature. Rather than attempting to solve the entire problem at once, it is decomposed into smaller tasks that are easier to manage by separate modules. This offers a high degree of design flexibility, since each sub-task has a different objective and a specialized architecture that performs best for the particular sub-task may be chosen.

This design can be summarized as:

$$\begin{aligned} f &= Feature\_extraction(i) \\ s &= Sequence\_modeling(f) \\ o &= Classification(s), \end{aligned} \tag{5.1}$$

where i represents the input sequence and o is the output word of the network. A depiction of the above can be seen in Figure 5.1.

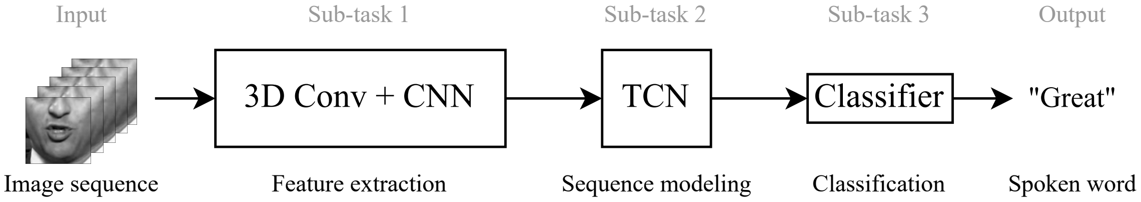

Figure 5.1: Design overview of the general VSR pipeline with mutliple sub-tasks, where the most suitable model handles each sub-task. In this Chapter, we explore several lightweight CNNs and efficient block designs for the TCN while Softmax is used for classification.

The high-level overview of a VSR system in Eq. 5.1 outlines three distinct steps when processing an input sequence. The first two are often handled by neural network models as they offer strong performance for these two tasks. Simultaneously, these are the most computationally intense modules in the entire architecture. When designing a lightweight VSR system, component selection plays a crucial role as their size and structure (i.e., amount of blocks and operations used, including other hyperparameters) determine the computational overhead of the overall architecture.

In the following Subsections, we construct a multitude of architectures utilizing a variety of lightweight networks for the task of feature extraction and then develop various efficient block designs for the task of sequence modeling, corresponding to each sub-task. We benchmark these architectures on the LRW test set, finding the optimal ones for this dataset that perform best while being lightweight in terms of resources (FLOPs and network parameters) and then combine them in unified models.

5.1.1 Visual Feature Extraction

The visual feature extraction step involves spatial processing of the input sequence, generating a set of intermediate representations with a high channel dimensionality. The convolutional networks employed in this step are designed in stages, where the operations in each stage gradually reduce the spatial size of the input while simultaneously increasing the channel dimension.

Since our focus is developing lightweight end-to-end architectures for isolated word VSR, we experiment with several lightweight models proposed in the image classification literature, since we find that these networks tend to perform well as feature extractors. This selection of networks covers a diverse range of approaches in terms of network and block design and can indicate which lightweight model is more suited to the task of feature extraction in the context of visual speech recognition. More specifically, we experiment with the following networks:

- *MobileNetV2* [81] was selected for its high performance and low computation cost. It utilizes an *inverted residual* block design that shifts the connectivity of shortcuts and utilizes a point-wise convolution to increase the amount of channels, as opposed to a standard residual block.

- *MobileNetV4-S* [314] is a recently-proposed model resulting from a Neural Architecture Search (NAS) process, meaning that its structure was searched automatically using an algorithm rather than being manually-designed. MobileNetV4 incorporates a series of innovations regarding block design and was selected to explore the performance and adaptability of an architecture for image classification that was searched with an objective that balances the trade-off between latency and performance.

- *EMO-1M* [315] introduces a building block based on the inverted residual design of MobileNetV2 that is combined with a self-attention mechanism.

- Next, we also choose *InceptionNeXt-A* [316], which combines modern block design principles [317] with three parallel depth-wise convolutions inspired by the Inception [80] block. To keep computation overhead manageable, the block splits its input to equal parts applies each convolution to a different chunk, retaining a skip connection and concatenates the outputs.
- Finally, *StarNet-050* [318] presents an alternative approach to efficient network design by adopting the (element-wise) multiplication operator to combine high dimensional features within a building block.

As in the previous Chapters of this dissertation, each CNN is superseded by a small stack of 3D convolution, Batch Normalization and non-linear activation layers, which serves as a spatio-temporal processing unit extracting more short-term dependencies from the input. This convolution uses a 3D kernel shape of $(3, 5, 5)$, where 3 corresponds to the temporal dimension and $5, 5$ to the spatial, with an output of 32 channels and its computational overhead is marginal compared to the other components of the overall pipeline. Using this small stack is common practice for the task of VSR, e.g., [28, 110, 162], where an additional pooling layer is added to further reduce the spatial dimensions keeping computations low. We find that when using lightweight networks, the pooling layer does lower the network size and computational complexity, but significantly harms recognition performance and for this reason we do not use this layer.

5.1.2 Sequence Modeling Network

The sequence modeling network ingests the features produced by the previous step for further processing by modeling the temporal aspect of the input sequence. The goal is discovering the inter-relationships that exist between features across the length of the sequence, since during complex speech (words, phrases and sentences) the movements of the mouth follow sequential patterns of motion. Capturing this information can lead to higher accuracy when making a prediction.

Related research in the VSR literature has demonstrated that improving the sequence modeling network can bring significant benefits in word recognition performance, as seen by the improvements obtained by the works of e.g., [110] and [162], where the same visual feature extractor network is used. The higher performance obtained can be attributed to replacing recurrent networks with TCN-based models and then modifying the temporal convolution blocks with more powerful designs, since the other components remain unchanged. In a similar fashion, we aim to develop a lightweight and powerful sequence modeling network that keeps the overall computation at affordable levels while raising accuracy. For this reason, we adopt the TCN formulation as the backbone structure for sequence modeling and explore several block designs borrowed from lightweight CNNs. To adapt these blocks to our task of VSR, we convert all 2D layers (i.e., convolution and normalization) to 1D.

More specifically, we use:

- The *Linear* block [319] which utilizes two depth-wise convolutions with a point-wise in-between which is used to fuse the information from the channels,
- the *Fused MB* block [320] that relies on a regular convolution to expand the channel dimensionality and a point-wise convolution to mix the channels,
- the *Inverted Residual* block [81], that reverses the order of operations of the Linear layer (i.e., uses a depth-wise convolution between two point-wise ones) and has been a popular building block in several lightweight architectures, e.g., [63, 279] as well as a starting point for other efficient blocks, e.g. [145, 317, 321, 314].
- The recently-proposed *UIB* block, introduced in [314], which is an advanced and flexible efficient block that is employed on the MobileNetV4 family of models [314].
- The also recently-proposed *CIB* block [321], representing advances in convolution-based lightweight blocks for compact networks and is the basic building block of the *YOLOv10* backbone [321],
- And finally, the *Star* block (variant *V*), proposed in [318] that exploits the multiplication operation to explore how this approach performs for the task of VSR.

Diagrams showing the structures of these blocks are shown in Figure 5.2.

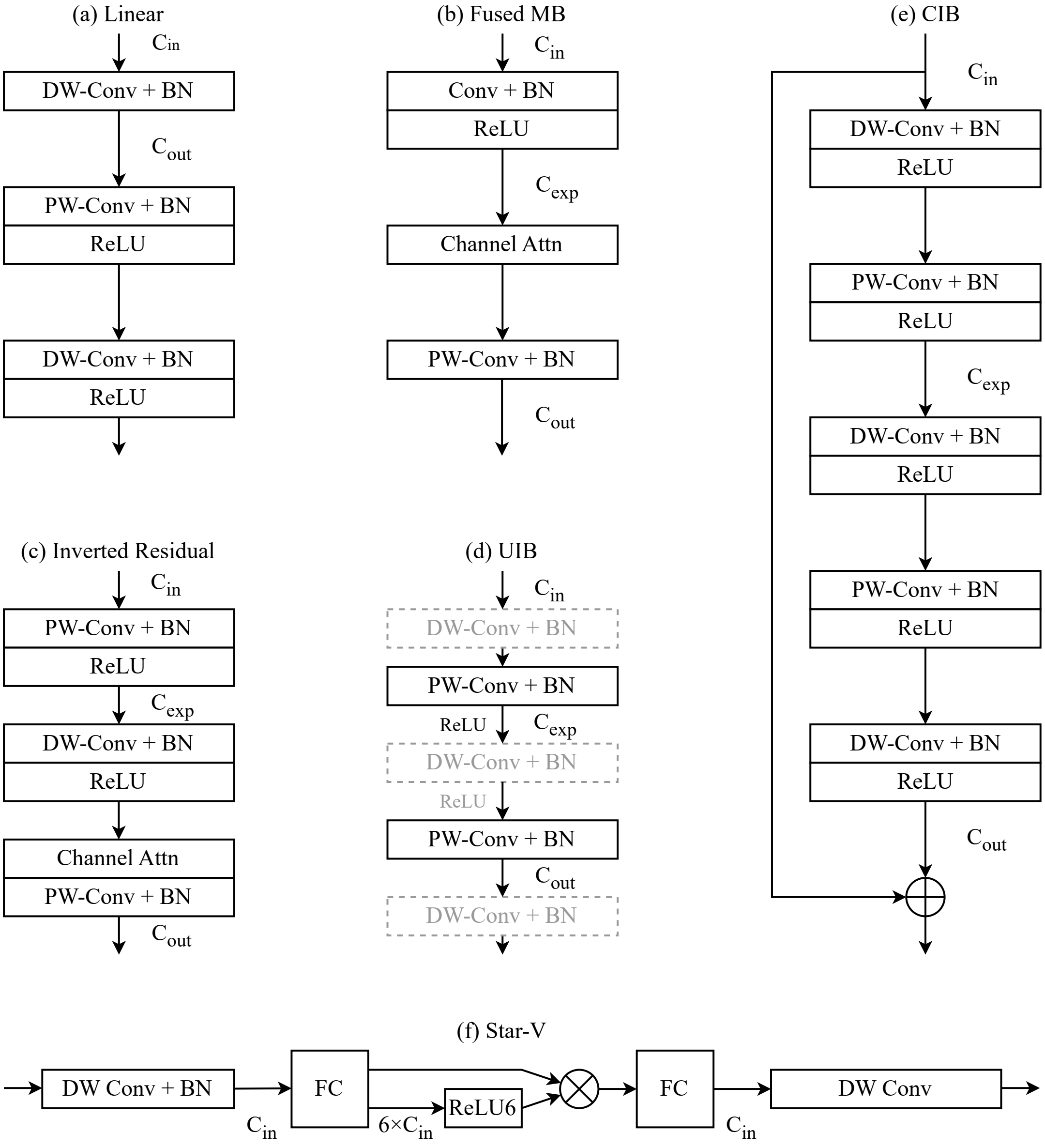

Figure 5.2: Structural illustration of the lightweight building blocks that were used in this Chapter. Convolution and Batch Normalization layers were converted to 1D. C_{in}, C_{exp} and C_{out} refer to input, expanded and output channels within a block. Layers in gray coloring (in UIB) are optional and activated by hyper-parameters. $\otimes$ denotes element-wise multiplication (Hadamard product).

5.2 Experimental Setup

5.2.1 Dataset & Preprocessing

The models proposed in this Chapter are trained and evaluated on the LRW [21] dataset, which is currently the largest openly-available corpus for visual speech recognition of English words in-the-wild. For more details on LRW, the reader is referred to the datasets Section in Chapter 2.

For pre-processing, a series of steps are performed on the raw data. After detecting the speaker's face in each frame, a face alignment network is employed to compute RoI landmarks. Then, normalization of the images occurs by removing variations of size and rotations using a mean face shape. A bounding box of shape 96×96 crops the region around the speaker's mouth area, which is further normalized by mean and standard deviation and finally converted to gray scale to remove color, resulting in a simpler (from a computational standpoint) final image. This procedure is typically employed by works in the literature that are trained on the LRW dataset (e.g., [110, 162]).

5.2.2 Training Setup

We follow the training setup of the previous Chapter, reviewed here. All network weights are initialized randomly without using any pre-trained checkpoints and all models proposed in this Chapter use the following configuration. The LRW training set is used and after each epoch the model is validated on the validation set. We train for a fixed amount of 80 epochs in total, saving the best-performing (in the validation set) checkpoint. At the end of training, the best-performing weights are loaded and the model is evaluated on the LRW test set. Stochastic Gradient Descent with an initial learning rate of 0.02 is used to update the weights. A decay factor of 0.0001 is applied to all weights to prevent over-fitting. The GPUs used to train and evaluate the models are Nvidia RTX 2080Ti with 11 GB of VRAM, therefore we use a batch size of 32 which allows fitting an entire model to a single GPU. The learning rate scheduling followed is cosine annealing, which has been found to perform very well for this dataset (refer to Subsection 3.4.2 in Chapter 3). We also use the same training-time augmentations as in the previous Chapter.

5.2.3 Evaluation and Discussions

In keeping up with the structure of Section 5.1, the results and discussions presented herein follow the same order, beginning with the feature extraction experiments and followed by an evaluation of the temporal convolution blocks. To measure parameter and Floating Point OPerations (FLOPs), the `torchinfo`[1] python package is used. All measurements are obtained using a single sequence as the input (29 frames with 88×88 resolution), to simulate applying the architecture to a video of the LRW test set in an in-the-wild scenario, providing an accurate representation of real-world resource requirements. In addition, all models are trained using the same setup as described in Section 5.2.2.

Feature Extraction

To benchmark the lightweight feature extractors, we employ a simple TCN-based sequential model following the architecture of [155]. This model is structured in 4 stages, each containing a block with a sequence of temporal convolution, batch normalization and rectified linear unit as an activation functions, repeated twice. The kernel size for all convolutions is set to 3 while the input and output channels are set to 512. The dilation rate used by both convolutions in each block is exponentially increased at every stage (to 2^{stage}), beginning from 1. To keep comparisons fair and meaningful, we use this TCN sequence model in combination with each lightweight feature extraction network discussed earlier, creating different end-to-end architectures. As a baseline for comparison, we use an 18-layer residual network [31] (also using this TCN configuration), which is favored by many VSR works for its strong performance (e.g., [28, 110, 162]) at the cost of very high complexity (compared to the lightweight networks). Results are presented in Table 5.1.

Unexpectedly, the lightweight feature extractors achieve lower complexity and model size at the cost of performance. The MobileNetV4-S [314] model is the strongest performer over all the lightweight architectures at 84.8% accuracy, 2.9% lower than the baseline. While its network size is smaller than other models, e.g., InceptionNext-A [316], its complexity in terms of FLOPs is the highest following the baseline, which is arguably the reason it achieves a higher accuracy. A similar behavior can be noted for the EMO-1M [315] and StarNet-050 [318] models, which are close in recognition

[1] `https://github.com/TylerYep/torchinfo`

Table 5.1: Benchmark results of lightweight feature extractors on the LRW test set compared against the much larger and more expensive residual network baseline. Each model is combined with a 4-layer TCN.

Model	**FLOPs ($\times 10^9$)**	**Params ($\times 10^6$)**	**Accuracy**
	Total (CNN)	**Total (CNN)**	**↑ (%)**
ResNet baseline (18-layer) [31]	31.2 (30.8)	17.7 (11.1)	87.7
MobileNetV2 [81]	0.9 (0.5)	8.5 (1.9)	82.9
MobileNetV4-S [314]	1.9 (1.5)	7.7 (1.2)	84.8
InceptionNext-A [316]	0.8 (0.4)	9.6 (3.0)	83.1
EMO-1M [315]	1.4 (1.1)	7.8 (1.2)	83.6
StarNet-050 [318]	0.7 (0.4)	7.0 (0.4)	82.7

accuracy with similar parameter counts and slightly lower complexity (FLOPs). The InceptionNext-A model measures higher in parameters compared to the others due to its multi-convolution design which nevertheless keeps the operations rather low, which could be a reason for its performance, similarly to the StarNet-050 model.

All lightweight feature extractors reduce the overall size by an amount ranging from 8.1 (45%) to 10.7 (60%) million parameters. A more remarkable improvement lies in the overall computational complexity of the models, which is explained by the more efficient block designs and amounts to a 94% reduction in the case of MobileNetV4-S (the highest FLOP count and performance among the lightweight models) and 98% for StarNet-050 (which achieves the lowest results in both metrics). The remaining models in our experiments still achieve reductions $\geq$ 95% in complexity. For the purposes of this Chapter, the MobileNetV4-S model achieves the highest accuracy over the other lightweight models, 1.2% higher than EMO (which comes second in performance) at very similar FLOP and parameter counts and will be used as the feature extractor in the following experiments.

Temporal Convolution Blocks

Having benchmarked the lightweight networks for visual feature extraction, we shift our attention to the sequence modeling component of the architecture. All experiments in this Subsection use a unified model that combines a MobileNetV4-S, which is the best-performing lightweight feature extractor, with a TCN-based architecture that employs a customized temporal convolution block described in detail in Section 5.1.2. The comparison is shown in Table 5.2 and demonstrates how the different lightweight block designs perform in the task of VSR of isolated words when combined with a low-resource convolutional feature extractor. Simultaneously, it showcases which of the blocks can be used to recover some performance that is lost due to using smaller networks, creating an architecture that is compact but performs competitively.

Table 5.2: Benchmark results of using different temporal block configurations with the MobileNetV4-S feature extraction architecture on the LRW test set compared against the baseline TCN.

Temporal Block	**FLOPs ($\times 10^9$)**	**Params ($\times 10^6$)**	**Accuracy**
	Total (TCN)	**Total (TCN)**	**↑ (%)**
TCN block (baseline) [155]	1.9 (0.2)	7.7 (6.2)	84.8
Linear [319]	1.7 (0.03)	2.5 (1.0)	83.5
FusedMB [320]	2.1 (0.5)	16.1 (14.6)	86.8
Inverted Residual [81]	1.8 (0.1)	5.6 (4.2)	83.4
CIB [321]	1.7 (0.1)	5.7 (4.2)	86.7
UIB [314]	1.9 (0.2)	9.8 (8.4)	86.8
Star-V [318]	2.0 (0.3)	14.0 (12.6)	88.1

In this evaluation, we can see that the Linear [319] and Inverted Residual [81] blocks cause a drop of $1.3\% - 1.4\%$ in performance, arguably the result of reducing the network parameters, since the FLOPs are largely unaffected ($0.1 - 0.2$ GFLOP reduction). Of these two blocks, the former (Linear) is more efficient and suitable for very low resource scenarios, reducing the baseline parameters by more than $3\times$ and achieving a slightly higher accuracy than the Inverted Residual. The FusedMB [320] block performs similarly to the CIB [321] and UIB [314] blocks with nearly the same accuracy, however it uses a regular convolution which, when combined with a high expansion ratio, raises the overall parameters and complexity of the network (more than $2\times$ that of the baseline). The CIB block reduces both FLOPs and parameters and simultaneously raises the accuracy by 1.8%, which can be attributed to its more modern design and can be considered as an upgrade to the standard TCN without any drawbacks and is also a strong candidate when designing a lightweight solution. The UIB block performs identically to the FusedMB but is a preferable choice as it is more efficient regarding parameters (by about 39%) as well as FLOPs (0.2 fewer GFLOPs) and when compared to the standard TCN block it raises the parameter count by 2.1 M and accuracy by 2.0%, results that can be explained by its modernized design, similar to CIB. Nevertheless, the CIB block is a more efficient choice than UIB as it almost matches its performance (0.1% difference), but with fewer GFLOPs and parameters (0.2 and 4.1M).

In these experiments, the best performing design is the Star-V [318] block, which surpasses the standard TCN by 2.7% accuracy at the cost of a larger size (1.8× more parameters) due to the large kernels used by its convolutions. Even so, the computational overhead of this model is maintained at manageable levels (0.1 GFLOPs more than the baseline), since the convolutions used are depth-wise. The combination of a TCN with Star-V blocks and a MobileNetV4-S feature extractor achieves the highest accuracy in this comparison, 0.4% more than the much larger and computation-heavy residual network (of Table 5.1) while being 10 million parameters smaller and having an impressive 15.65× fewer FLOP count.

Next, we perform another round of benchmarking of the lightweight feature extractors as well as the larger ResNet from the previous Section by combining them with a TCN using the best-performing temporal block (Star-V) and tabulate the results in Table 5.3.

Table 5.3: Combining the TCN with Star block (variant V) with other visual feature extraction networks. Measurements (FLOPs and parameters) include both components.

Temporal Block	**FLOPs ($\times 10^9$)**	**Params ($\times 10^6$)**	**Acc. ↑ (%)**
ResNet (18-layer) [31]	31.3	24.0	90.0
MobileNetV2 [81]	1.0	14.8	87.0
InceptionNext-A [316]	0.9	15.9	86.6
EMO-1M [315]	1.6	14.1	86.7
StarNet-050 [318]	0.9	13.3	87.0

Compared to the standard TCN design used to benchmark the feature extractors (Table 5.1), using the Star-V block can raise performance by up to 4.3% while only adding 0.2 GFLOPs which is the case for StarNet-050 [318], making this combination ideal for situations with rather constrained computational capabilities (e.g., edge devices). We observe a similar outcome for all other lightweight models, where non-trivial raises in accuracy are achieved, showing that using this block can recover lost performance with a minimal impact on computation. Invariably, the number of parameters is increased due to the design of the Star-V block (see Figure 5.2), as explained previously, representing a trade-off with the improvement in network accuracy which some applications might find an acceptable compromise. When paired with the larger and more powerful 18-layer residual network it achieves a 2.3% improvement over the standard TCN, reaching a 90.0% recognition accuracy. These results also show that the Star-V block scales well with all networks regardless of their size and complexity and establish it as a powerful building component for a sequence modeling network when designing efficient end-to-end models for VSR.

5.2.4 Comparison With Other Methods

Finally, we compare our best-performing model with other approaches from the literature on the task of word-level VSR on the LRW dataset. Since we do not use additional training data, word boundaries or audio cues, for a fair comparison, we compare with models that meet these criteria. The reasoning behind this choice is that using additional training data such as extra datasets or video sequences is not feasible for several languages taking into account the additional effort required to collect and annotate the data, not to mention the additional training time which can be a factor in some applications. Similarly, word boundaries indicate the frame where the word is present in the video clip, which is additional information that is not available in-the-wild and thus does not reflect real-world conditions. As for audio, while some works utilize the audio stream in architectures that leverage both audio and video modalities, we consider it out of scope as we develop a video-only approach since the audio is not always available (e.g., in silent footage). For each method, we also include size and computational complexity measurements, providing a more complete comparison between the different models.

Compared to other methods from the VSR literature, our model achieves slightly lower recognition performance, which is a consequence of its smaller size and complexity. Since most architectures in this comparison utilize a residual network baseline, our model costs 5× fewer FLOPs, as it uses the much smaller MobileNetV4-S feature extraction network, which also saves about 10 million parameters. Simultaneously, the

Table 5.4: Comparison of our method (highlighted) with recent works from the word VSR literature.

Model architecture	**FLOPs** ($\times 10^9$)	**Params** ($\times 10^6$)	**Accuracy** ↑ **(%)**
ResNet + MS-TCN [110]	10.31	36.4	85.3
ResNet + DC-TCN [162]	10.64	52.54	88.3
ResNet (G) + MS-TCN (G) (Chapter 4)	1.78	14.29	86.24
ResNet (G) + DC-TCN (G) (Chapter 4)	2.03	27.04	87.58
ResNet (G) + DC-TCN (Chapter 4)	2.67	41.77	88.72
ResNet + DC-TCN (G) (Chapter 4)	10.01	37.81	89.10
ShuffleNet v2 (1×) + MS-TCN [268]	2.23	28.8	85.5
ResNet + 3×Bi-GRU [69]	10.54	59.5	88.4
ResNet + 2×Bi-LSTM [64]	10.24	50.07	88.7
MobileNetV4 + TCN (Star-V block)	2.03	14.0	88.1

TCN with Star-V blocks is much more compact than the recurrent architectures or the larger TCN variants that use multiple convolutions per block. The model of [268] is comparable in overhead to our models but is surpassed in accuracy, while being more than double in overall size and the same applies to the lightweight networks that were proposed in the previous Chapter. In fact, regarding parameter counts, our end-to-end model is the smallest one in this comparison and also the most efficient in terms of complexity, yet in spite of its small size, it falls behind some of the larger models by only about 1.0% accuracy. These results demonstrate our proposed model's strong performance at minimal size and network complexity, attributes that make it an ideal choice for applications where a highly efficient model is needed.

5.3 Ablation Studies

5.3.1 Star blocks

The work of [318] introduces several variants for the *Star* block with similar architectural designs and common characteristics. All blocks start and end with a depth-wise convolution operation and include point-wise convolutions that expand and subsequently restore the input channels according to a fixed expansion rate. Another similarity is the use of the RELU6 function for non-linearity and the multiplication operation for feature mixing:

$$\text{ReLU6}(x) = \min\big(\max(0, x), 6\big) \tag{5.2}$$

This function is widely employed in lightweight networks as it empirically performs better than ReLU under low-precision conditions [322].

The reader is referred to the Supplementary material of [318] for more details on the architectures of each Star block variant. In our case, we convert each variant to 1D for use in the TCN and train separate models, presenting the results in Table 5.5. A MobileNetV4-S is used for feature extraction.

Table 5.5: Ablation study on the architecture of the Star block used in the TCN model. For size and complexity, only the TCN is measured.

Block	**FLOPs ($\times 10^9$)**	**Parameters ($\times 10^6$)**	**Accuracy ↑ (%)**
Star-I	0.36	12.6	87.9
Star-II	0.36	12.6	87.0
Star-III	0.73	25.2	88.1
Star-IV	0.36	12.6	87.9
Star-V	0.36	12.6	88.1

The best-performing designs of the Star block are variants III and V, achieving the same accuracy, while variants I and IV are following closely. In this comparison, variant II achieves the lowest performance at 87.0%, which is still higher than the other lightweight blocks in Table 5.2. In terms of complexity and parameters, since all variants share a similar architecture, we notice identical measurements in FLOPs and parameters with the exception of variant III, which applies an additional pointwise convolution after expanding channels, causing the increase in parameters and FLOPs. Given these measurements, the best choice for the task of VSR is variant V, which achieves the highest performance while being as efficient as the other variants.

5.3.2 Architecture configuration

The previous experiments used a temporal convolution network with a four-stage design and 512 channel outputs. This amount of stages is used, to the best of our knowledge, by virtually all works in the VSR literature that employ a TCN model for sequence modeling, since they typically adopt the models of [110] or [162] that are also four-stage architectures. Similarly, convolutional networks (e.g., ResNets [31], MobileNets [81]) also commonly use four stages in their designs. Regarding the amount of channels in each block, the use of 512 is empirical, striking a balance between complexity, size and accuracy. In Table 5.6 we show the results of an ablation study on the stages and channels of the TCN, experimenting with architectures that are shallower (fewer stages with more channels per stage to compensate) or deeper (more stages with fewer channels, respectively). As before, a MobileNetV4-S is used for feature extraction.

Table 5.6: Ablation study on the configuration of the TCN architecture. For size and complexity, only the TCN is measured.

Configuration	**FLOPs**	**Params**	**Accuracy**
Stages ; Channels / stage	**($\times 10^9$)**	**($\times 10^6$)**	**↑ (%)**
2 ; 1024	0.51	17.8	86.0
3 ; 768	0.53	18.5	87.2
4 ; 512	0.36	12.6	88.1
6 ; 256	0.18	6.4	87.6
8 ; 128	0.10	3.4	86.2

This ablation study showcases that a network with four stages and 512 channels is the best approach for the task of VSR as this configuration out-performs all other setups. Making the network shallower by reducing the number of stages gradually degrades performance as more stages are required to process the information, while compensating by increasing the channels of the convolution operations in each temporal block leads to an increase in network overhead. Similarly, making the network deeper by adding stages also lowers its performance (albeit by a lower amount than removing stages), which can be caused by two factors: lowering the amount of channels per layer, thus hampering the expressiveness of the network and its ability to capture information and impeding backward gradient flow during training since the back-propagation path becomes longer. The deeper networks perform marginally better than the very shallow ones, while being much more efficient in terms of overhead and are suitable for special cases with high resource restrictions.

A key attribute of the TCN design is the dilation factor used in each convolution which increases with every block. It is therefore possible that in the case of shallow networks (dilation= $\{1, 2\}$) this rate is rather low and could be another cause for the lowered performance since the second block does not process a broader amount of information, potentially missing key temporal relationships from neighboring frames. For the deeper networks, the later stages employ high rates of dilation (since it is doubled at every block) which allows them to cover temporally distant information that might be irrelevant to the current frame. Investigating these hypotheses is left as future work.

5.3.3 Block Hyperparameters

As an additional ablative study, we investigate how parameter selection for some block designs affects the model's computational overhead and recognition performance. This allows us to fine-tune each block according to specific conditions and requirements, since for instance some applications might benefit from a temporal block design other than Star-V due to hardware factors, therefore it is worth considering alternative architectures for such cases. We use the next three best-performing blocks after Star-V, which are FusedMB [320], CIB [321] and UIB [314], that achieve comparable accuracy on LRW (around 86.7%) as shown in Table 5.2. For FusedMB, we modify the kernel size used in the first convolution and activate or deactivate the Squeeze-Excitation (SE) [65] attention module that the block utilizes, while for CIB we only experiment with the kernel size which is shared by all convolution layers in the block. Regarding UIB, its structure allows for additional convolution layers that can be added (see Figure 5.2(d)), allowing for a greater design flexibility than FusedMB, we therefore enable all extra convolutions and change the kernel sizes, to investigate how the added receptive fields affect performance. Since these blocks follow an inverted residual philosophy, the *expansion ratio* is a hyper-parameter that controls the module's capacity and overhead. The architecture setup of the TCN in our models uses 4 computation stages with 512 channels used as inputs to each stage, so in order to avoid an excessive increase in computational complexity due to using a high expansion ratio, we use a value of 2 for both blocks. For visual feature extraction, we employ the MobileNetV4-S network and the whole end-to-end model is trained with the same settings as all previous experiments. The results are presented in Table 5.7.

Table 5.7: Hyper-parameter study for two temporal block designs derived from *FusedMB* and *UIB*. A kernel size of 0 indicates that a particular convolution layer is disabled. For size and complexity, we measure only the TCN component. Results are on the LRW test set.

Method	**Hyper-parameter(s)**	**Value**	**FLOPs ($\times 10^6$)**	**Params ($\times 10^6$)**	**Accuracy ↑ (%)**
Baseline	Kernel size	3	229	6.29	84.86
FusedMB [320]	Kernel size, expansion, SE [65]	3; 2; ✓	292	10.49	86.39
		3; 2; –	290	8.40	86.45
		5; 2; ✓	524	14.69	86.86
		5; 2; –	522	12.59	86.65
CIB [321]	Kernel size, expansion	3; 2	122	4.24	86.26
		5; 2	123	4.26	86.79
UIB [314]	Kernel size, expansion, activation	3,3,0; 2; ReLU	122	4.23	84.42
		3,5,0; 2; ReLU	122	4.24	84.25
		5,3,0; 2; ReLU	122	4.24	84.96
		5,5,0; 2; ReLU	123	4.24	84.42
		3,3,3; 2; ReLU	122	4.24	84.88
		3,5,3; 2; ReLU	123	4.25	85.00
		5,5,3; 2; ReLU	123	4.25	85.09
		5,3,5; 2; ReLU	123	4.25	85.27
		5,5,5; 2; ReLU	123	4.26	85.03
		7,0,0; 4; GeLU	244	8.42	86.89

These results showcase that FusedMB is robust to hyper-parameter selection and achieves a consistent accuracy benefit over the standard TCN. However, the maximum recognition rate of the models using this block still trails below that of Star-V. Regarding overhead, the kernel size is the most significant factor affecting the block's parameter size and complexity, as going from a kernel size 3 to 5 almost doubles both FLOPs and total parameter counts. The addition of a SE module to this block seems to benefit a larger kernel size, increasing accuracy by 0.31%, but actually causes a minor drop in performance when the kernel size is smaller. Simultaneously, the SE module hardly affects the block's overhead and can be beneficial in some instances. The high parameter size of this block is a direct result of using a standard convolution, seeing that UIB and CIB that employ depth-wise convolutions amount to fewer parameters as well as reduced processing.

The lightweight design of CIB is a better performing choice than FusedMB, while also being much more efficient in both metrics. In fact, for a kernel size of 5, it surpasses all FusedMB setups, at fractions of hardware requirements; achieving reductions around $57-76\%$ FLOPs and $49-71\%$ parameters, depending on the setup. A higher kernel size for this block leads to greater performance, raising the recognition rate by 0.53% without affecting its overhead. Compared to the other two methods, this module provides the best performance-to-size ratio as it achieves nearly the same accuracy with specific UIB and FusedMB configurations, making it an ideal choice for certain scenarios with severe hardware limitations. CIB outperforms the baseline TCN at a very compact size, reducing both FLOPs and parameters and as mentioned previously (Subsection 5.2.3), it can replace the baseline block without any drawbacks.

For UIB, introducing an additional convolution at the beginning of the block (using only the second convolution between the point-wise convolutions equates the block to an Inverted Residual one [81]) with higher accuracy over the baseline TCN, but falls behind other options (and in this case the FusedMB block). Activating all convolutions in this block further improves accuracy, with the best option being a $5, 3, 5$ kernel size setup for the three convolutions. The number of convolutions and their kernel size has little effect on overall parameters or FLOPs, since these convolutions are depth-wise and rather low in complexity. With three convolutions, this block resembles CIB in its structure and differs only in the amount of non-linear activation functions; in CIB they are used after every convolution operation, while in UIB the first and last depth-wise convolutions with the last point-wise do not use any. Still, the results of this block are consistently behind FusedMB, Star-V and CIB by varying amounts in the $1-2\%$ range. Another difference is the choice of activation function, with UIB using the rectified linear unit (ReLU) and CIB opting for SiLU (sigmoid linear unit) instead. It is possible that the difference in performance between these two blocks is the result of a combination of these two factors, which is further reinforced by the fact that UIB with a Gaussian linear unit (GeLU) function obtains a better recognition rate. The choice of non-linear activation function remains an open problem and warrants further investigation.

As an additional experiment, we adopted the hyper-parameter settings of the block used in [317], which incorporates modern practices for block design and note that it achieves higher accuracy than all other UIB setups and slightly higher than FusedMB (only by 0.03%) while being less resource-intensive than the latter. This special case employs only one convolution operation with a high kernel size and an expansion ratio of 4, which causes the increase in overhead over the other UIB setups. This result, leads us to believe that this hyper-parameter is the most impactful for recognition performance and is a probable explanation for the improved results, considering that the best-performing block (Star-V) uses an expansion ratio of 6 and that we also obtain similar results regardless of kernel size and setup (a phenomenon not entirely limited to the UIB block). We also believe that it is possible to further improve performance for both these blocks by using a higher ratio than the current value of 2, which acts as a limiter for representational capacity. This will inevitably cause an increase in network overhead, which will be mostly evident in parameter counts, similarly to Star-V. Further experiments are required to verify these claims.

5.4 Conclusion

In this Chapter, we presented a systematic approach into designing lightweight architectures for practical visual speech recognition applications. A VSR system predicts spoken words from the input sequence in two distinct computation steps, visual feature extraction and temporal sequence modeling, each handled by a different component. For visual feature extraction, we benchmarked several lightweight models from the image classification literature, finding that significant reductions in FLOPs ($\geq 94\%$) and parameter counts ($\geq 45\%$) are achievable, compared to the expensive baseline network, at the cost of lowering recognition accuracy. For modeling the temporal aspect of the extracted features, a multitude of efficient designs were explored as temporal block replacements in a standard TCN model, taking advantage of its favorable properties and high performance for the task of VSR. Our findings show that when an efficient feature extraction model is combined with a robust sequence model, significant gains in accuracy are possible, mitigating the losses that occur from lowering the network capacity. We also observe that the Star block which takes advantage of the Hadamard product is a very strong performer in the task of single word VSR, demonstrating impressive performance at low network overhead, that also scales favorably with larger networks. Our most efficient model combines a MobileNetV4-S with a four-stage TCN using Star-V blocks and is very competitive with other much larger methods from the VSR literature. The ablation studies performed indicate that while the kernel size of the convolutions and the expansion ratio used in the lightweight temporal block designs can both impact overall performance, the latter is a more significant hyper-parameter as it governs the capacity of the block and subsequently its performance and resource overhead. Future work involves exploring techniques to further improve recognition accuracy, bridging the performance gap with larger models.

Chapter 6

Improving Lightweight VSR Performance

The previous chapters presented approaches for practical isolated word VSR that involved reducing the hardware overhead of the models in terms of size and complexity, where the focus was primarily directed to the employed models, their components and their overall architecture. These reductions were achieved by adopting various strategies such as compression (Chapter 3) to shrink network sizes, using cost-efficient modules (Chapter 4) to lower complexity or designing lightweight end-to-end models (Chapter 5) toward both these objectives.

In this Chapter, we consider an alternative approach that involves improving an existing lightweight network's recognition performance without causing significant changes (i.e., raises) in its overhead. Different from the previous chapters, we shift our focus on exploring methods and techniques aimed at enhancing the representational capabilities of an established lightweight end-to-end network. The machine learning and computer vision literature offers a wide selection of methods specifically aimed at improving network performance, one of the most widely-used being the addition of attention modules to an existing network. These modules typically operate on the channel dimension of the feature maps produced by the network's blocks and are designed to not be highly demanding in resources, making them suitable for our purposes.

We use the model proposed in the previous Chapter which was developed with practical applications in mind and serves as a baseline model for our experiments. We explore channel attention modules that are added to the sequential modeling component, complementing the already impressive performance achieved by the Star-V blocks. Furthermore, we introduce training-time regularization in the form of dropout following two strategies regarding its application to the model. A benefit of this technique is that it does not affect the overhead of the model, improving its recognition rate at no extra cost in terms of resources. Finally, we combine the best-performing attention module and dropout strategy in a unified model and compare our results with other lightweight methods from the VSR literature.

6.1 Methodology

As a baseline architecture, we employ the end-to-end model of the previous Chapter, which represents a strong end-to-end baseline for practical VSR of isolated words with low computational complexity and high recognition performance.

The unified architecture is comprised by a series of networks, each fulfilling a different objective. The first network is a small spatio-temporal stack consisting of a 3D convolution, normalization and non-linear activation layers. It is a lightweight component, since the kernel size and output channels of the 3D convolution are set to $3, 5, 5$ and 32 respectively. Following, for visual feature extraction, the *MobileNetV4-S* [314] network is employed, which is a recently-proposed lightweight model derived from a novel two-phase Neural Architecture Search (NAS) process combined with an efficient inverted bottleneck block that improves performance. Next, for sequence modeling, the unified architecture is called Star-TCN and uses a TCN variant with an upgraded temporal block that employs a structure derived from the *Star-V* block [318]. It leverages the expressiveness and implicit high dimensions allowed by the use of the Hadamard product (point-wise multiplication operation between Tensors)

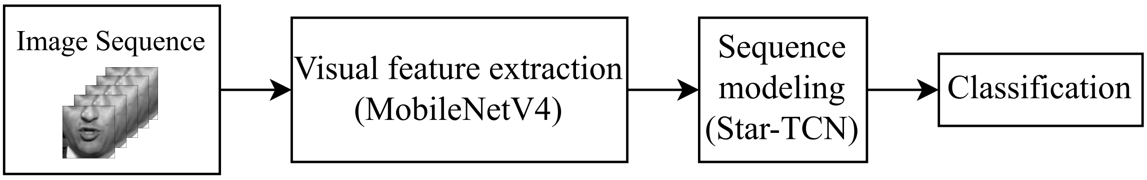
Image Sequence
Visual feature extraction (MobileNetV4)
Sequence modeling (Star-TCN)
Classification

Figure 6.1: Overview of the end-to-end VSR model used in this work. An attention method as well as regularization techniques are added to the sequence modeling component. We experiment with several channel attention methods from the literature and various strategies for adjusting the regularization amount. The input is a video sequence and the output is the prediction of the spoken word.

with the efficiency of depth-wise convolutions in a design that is lightweight yet still achieves high performance. The Star-TCN is structured in 4 main computation stages with one Star-V block comprising each stage and 512 channels are used as inputs to each block, representing a balance between word recognition performance and model complexity. At the end of the entire architecture, a fully-connected layer is added and classification is performed using the Softmax function. Figure 6.1 illustrates an overview of the entire model.

6.1.1 Incorporating Attention Mechanisms

Attention mechanisms have been proposed as methods or components that improve the representational capabilities of deep neural networks across various tasks, allowing them to capture contextual properties from the input as well as to emphasize relationships between feature maps or channels that are more influential for better performance at each task. Another design objective often taken into consideration when designing such mechanisms is maintaining the existing computation levels or adding minimal additional overhead to the base networks, which aligns with the goals of this work.

We adopt several attention mechanisms from the computer vision literature that operate on the channel dimension of a feature map, enabling seamless incorporation in the existing TCN architecture discussed previously. In fact, the only structural modification required lies within the temporal block, without any other changes to the TCN itself, allowing for straightforward development and testing of a multitude of setups. More specifically, we employ:

- *Squeeze-Excitation* (SE) [65] which utilizes an MLP and a gating unit to calculate the importance of each channel and re-weigh it accordingly to its contribution. It offers a strong attention mechanism in a compact package with proven results in several computer vision tasks.
- *Efficient Channel Attention* (ECA) [276] that follows the same philosophy as Squeeze-Excitation but maintains a lower computational overhead by utilizing a single convolution layer to model the channel interactions of the feature map.
- Next, *Shift-and-Balance Attention* (SBA) [323] which proposes regulating the impact of the attention branch on the feature map by a learnable scaling factor that controls and balances the attention and feature map branches.
- *Skip-Squeeze-and-Excitation* (SSE) [324], that similarly to ECA, avoids increasing the complexity of the base network by employing a single fully-connected layer rather than an MLP.
- We also use *Convolutional Block Attention Module* (CBAM) [135], which is another widely-used attention method with strong performance. It operates on the channel and spatial dimensions of the feature map in a sequential manner.
- Finally, *Gated Channel Transformation* (GCT) [325] that also aims to reduce the number of parameters and complexity by employing simple operations to embed and normalize each channel before applying a gating mechanism.

Structurally, the attention modules are illustrated in Figure 6.2 and implementation details with specific hyper-parameters for each module are provided in Table 6.1.

6.1.2 Introducing Regularization

Regularization refers to any approach that aims to improve a network's generalization capabilities by minimizing its performance discrepancy between training and test sets, a phenomenon that is known as over-fitting. Methods for regularization have been a topic of research interest by the machine learning community and follow a broad range of strategies, some with specific applications. The case where a model cannot sufficiently model the training data due to limited representational capacity or an excessive amount of training samples leading to low overall performance is called

Method	**Hyper-parameter**
Squeeze-Excitation [65]	MLP reduction ratio = 8
Convolutional Block Attention Module [135]	MLP reduction ratio = 16
Gated Channel Transformation [325]	Normalization = L_2
Skip-Squeeze-Excitation [324]	Output scaling amount = 1
Shift-and-Balance Attention [323]	MLP reduction ratio = 2, gate = Sigmoid
Efficient Channel Attention [276]	kernel size = $(log_2 C + 1)/2$

Table 6.1: Implementation details for the attention modules used in this work. C implies input channels, which are set to 512 for all modules.

under-fitting. Given that the methods proposed in this dissertation exhibit behavior that leads us to believe this to be the case, we study regularization as a method to reduce its effect.

A rather popular method employed for regularization that has stood the test of time is dropout. During training, at the forward propagation stage, dropout deactivates neurons by setting their connections to zero with probability p that is also known as *dropout rate*. The remaining (non-zero) connections are then scaled accordingly using a coefficient. This can be seen as a form of noise injection that encourages the network to avoid overly depending on specific hidden units, reducing outliers and produces a different neural network at each training pass, functioning as an implicit ensemble of models. While initially intended as a tool to mitigate over-fitting in neural networks, [326] has shown that dropout, when applied with a scheduling strategy that dynamically adjusts its value can also be effective for instances of under-fitting.

6.2 Experimental Evaluation

For all experiments, the trained models are evaluated on the LRW test set [21]. The scoring method is word recognition accuracy expressed as a percentage of correct predictions. We also report network size and computational complexity measured in parameters and Floating Point OPerations, respectively, in order to provide a clear overview of each model's requirements and overhead in terms of hardware. Such measurements are valuable when considering the application of a model in practical

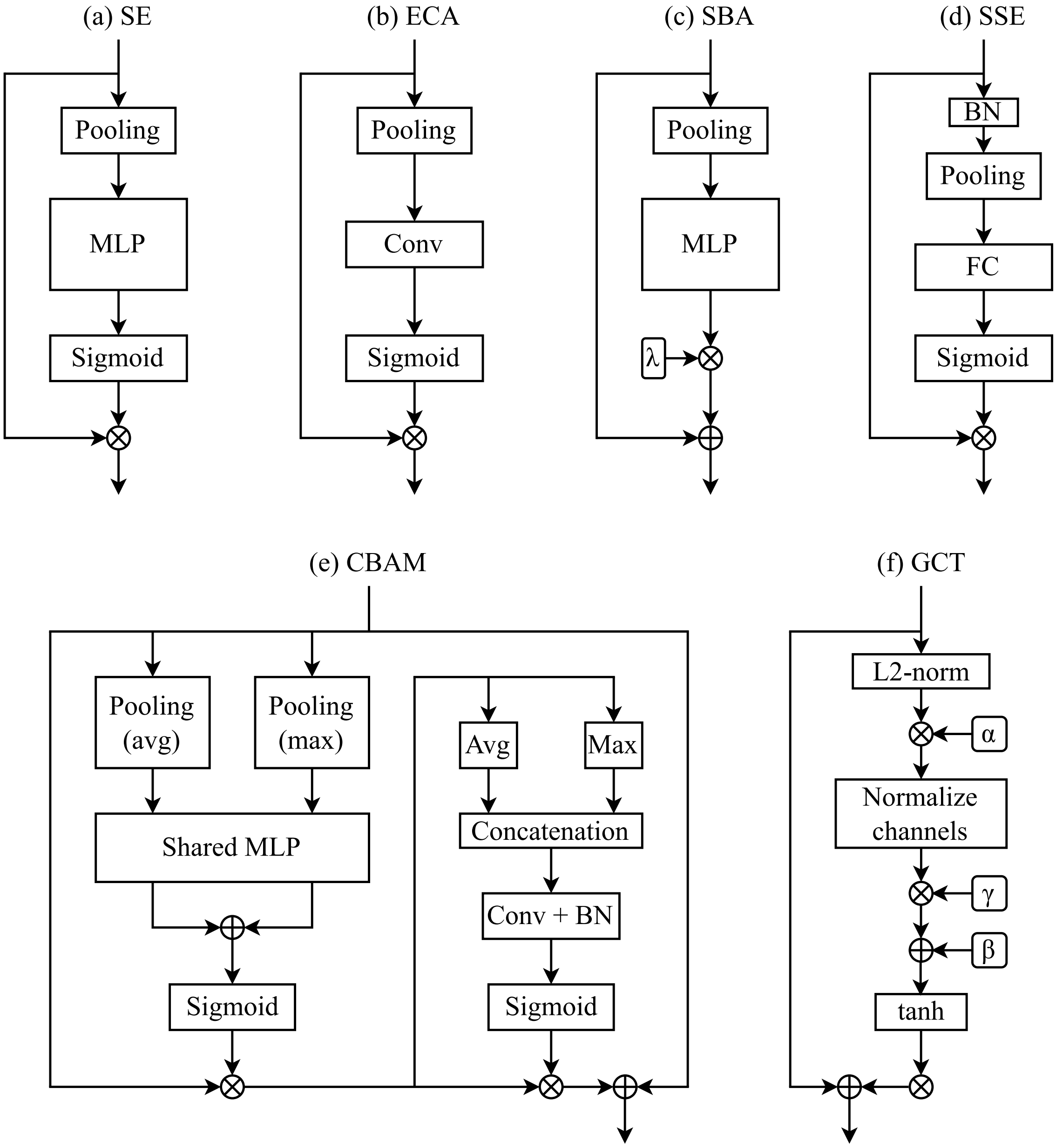

Figure 6.2: The attention modules employed in this Chapter.

scenarios with realistic conditions such as resource constraints.

The training strategy remains the same as in the previous Chapter, the optimizer used is Stochastic Gradient Descent, with an initial learning rate of 0.02, annealed by a cosine schedule (see Equations 3.6 and 3.8 in Chapter 3). For more details, refer to Subsection 5.2.2 in the previous Chapter.

6.2.1 Attention Modules

First, we evaluate the effect of each different attention module on the baseline model, which comprises a MobileNetV4-S feature extractor with a TCN that employs temporal blocks based on the Star-V design. While all methods explored in this work operate on the channel dimension of the feature maps, their structures differ, providing valuable insights as to which approach is more effective when combined with an end-to-end lightweight model for VSR. The results are tabulated in Table 6.2 and use the hyper-parameters shown earlier in Table 6.1. Ablative studies on the values of these hyper-parameters as well as the location within the Star-V blocks that the attention methods are integrated are performed in a later Subsection.

Model	Accuracy ↑ (%)	Model size ($\times 10^6$)	Complexity (FLOPs $\times 10^9$)
MobileNetV4 + TCN (Star-V block) (Chapter 5)	88.1	14.0	2.03
MobileNetV4 + TCN (Star-V block) + SE [65]	88.4 (+0.03)	14.3 (+0.3)	2.03 (+0.00)
MobileNetV4 + TCN (Star-V block) + CBAM [135]	87.9 (−0.02)	14.2 (+0.2)	2.03 (+0.00)
MobileNetV4 + TCN (Star-V block) + GCT [325]	88.2 (+0.01)	14.1 (+0.1)	2.03 (+0.00)
MobileNetV4 + TCN (Star-V block) + SSE [324]	88.5 (+0.04)	15.1 (+1.1)	2.04 (+0.00)
MobileNetV4 + TCN (Star-V block) + SBA [323]	88.2 (+0.01)	15.1 (+1.1)	2.04 (+0.01)
MobileNetV4 + TCN (Star-V block) + ECA [276]	88.6 (+0.05)	14.0 (+0.0)	2.03 (+0.00)

Table 6.2: Evaluation of different attention modules combined with the base model on the LRW test set. For network measurements, a single 29-frame sequence of size 88 by 88 pixels is used. Size and complexity measurements are for the whole end-to-end model.

In terms of accuracy, GCT [325] and SBA [323] offer the least amount of improvement, with GCT being much more efficient in terms of network parameters, adding 0.1M to the total count, while SBA adds 1.1M. In fact GCT is the most economical of all the benchmarked attention modules, followed closely by CBAM [135] and SE [65]. The best performance gain is obtained by the use of the ECA module [276], which also does not raise the parameter or FLOP counts. Even with four such layers (one at each block) in the TCN, it uses a regular convolution layer that operates on the pooled input feature map (with a dimension size of 1), without changing its size, which is a negligible cost. SSE [324] is the next best-performing option, being 0.1%

lower than ECA, but with 1.1M additional parameters and 0.01 more GFLOPs. The performance of the model with the SE reaches 88.4% with the module adding +0.3%, with only 0.3M more parameters.

Our results show that apart from CBAM, all other attention modules are beneficial for model accuracy, ranging from 0.1% to 0.5% and are rather lightweight, adding a negligible amount of FLOPs to the overall computation cost, which is a total fraction of network FLOPs. Of these methods, ECA is the best performance-enhancing option, which is simultaneously the most efficient. Including an attention module to the architecture of the previous Chapter can be seen as a performance upgrade at no extra computation cost when introducing some of the attention modules in our comparison. CBAM brings the least amount of improvement and actually lowers overall performance, an effect for which we offer two possible explanations. First, the module's design is more suitable for CNN applications, since the use of a spatial attention sub-module employs a convolution with a large kernel size (7) that fails to provide the same benefit for sequential data as it includes more time-steps in its calculation, potentially adding noise from features that are temporally distant. It is also possible that the weight initialization of the model is the cause of this result. We are planning on investigating this in the future by using a smaller kernel size in this block and conducting more experiments with different pseudo-random generator seeds which affect weight initialization.

6.2.2 Comparison With Other Methods

Since our goal is to not only improve recognition performance, but also to keep the overall overhead of the network at manageable levels, we choose ECA as the attention method to include in the architecture, as it out-performs the other methods without raising complexity. We then introduce regularization in the form of dropout after each temporal block in the TCN structure and train the end-to-end model with attention from the beginning, again using randomly initialized weights. For dropout, we use two approaches: a constant probability that stays static throughout the whole training process and a schedule that adjusts the dropout rate as training progresses. The reasoning behind using a dynamic rate for dropout is that it reflects the change in training dynamics as more epochs are completed and the network's training progresses while the learning rate is annealed to lower amounts. Adjustments to dropout probability during training can be accomplished in various ways and an ablation study is performed in a following Subsection. After training two structurally identical end-to-end models with different dropout strategies, we compare them with other lightweight methods from the literature on VSR and show the results in Table 6.3.

Model	**Accuracy** ↑ **(%)**	**Model size** **($\times 10^6$)**	**Complexity** **(FLOPs $\times 10^9$)**
ShuffleNet v2 + MS-TCN * [268]	85.5	28.8	2.23
ShuffleNet v2 + DS-MS-TCN * [268]	85.3	9.3	1.26
ResNet-18 + MS-TCN * [268]	87.9	36.4	10.3
ResNet (G) + MS-TCN (G) (Chapter 4)	86.2	14.2	1.78
ResNet (G) + DC-TCN (G) (Chapter 4)	87.5	27.0	2.03
ResNet (G V2) + MS-TCN (G) (Chapter 4)	86.1	15.5	2.00
ResNet (G V2) + DC-TCN (G) (Chapter 4)	87.9	28.3	2.25
MobileNetV4 + TCN (Star-V block) (Chapter 5)	88.1	14.0	2.03
MobileNetV4 + TCN (Star-V block) + ECA + Dropout ($p = 0.05$)	88.92	14.0	2.03
MobileNetV4 + TCN (Star-V block) + ECA + Late dropout ($p = 0.1, e = 60$)	88.93	14.0	2.03

Table 6.3: Comparison of our lightweight models with added attention module and regularization with other practical methods from the VSR literature. Evaluation is performed on LRW. Models marked with * are trained using knowledge distillation methods. For dropout, p is the probability and e indicates the epoch number of activation.

Our lightweight models with the ECA module and dropout regularization during training achieve the highest recognition rates on LRW over all other lightweight architectures. Compared to the networks proposed in [268], our models consistently out-perform them regardless of feature extraction network used, even for the much larger residual network with nearly $5\times$ more FLOPs. The *Depthwise-Separable TCN* proposed in that work is more efficient than our architecture in terms of size and complexity but falls behind in accuracy due to its reduced capacity, while the regular MS-TCN costs close to $2\times$ the parameters of our model and 10% more FLOPs. Furthermore, those models are trained with a knowledge distillation method that requires additional training of a teacher model, which is a more time-consuming process than standard neural network training that is employed in this Chapter and might be a limiting factor in situations where a lightweight solution is required in a timely manner. When compared to our proposed lightweight models from Chapter 4, the networks presented in these experiments are similar in complexity but are smaller in size and perform better.

6.3 Ablation Studies

We perform a series of ablation studies intended to provide more insights and understanding into how each module impacts overall performance and which choices are more important from that perspective. We apply each method on the baseline TCN with Star-V temporal blocks and to keep comparisons fair, we use the same training settings for all experiments, as described previously.

6.3.1 Attention Module Location and Hyperparameters

Since the attention modules operate on the resulting feature maps produced in the intermediate layers of the sequential modeling network, their location influences network performance and should be considered when incorporating these modules in the base architecture. We experiment with several possible locations within each temporal block, illustrated in Figure 6.3 and the results of each location are reported in Table 6.4. In these experiments, we employ the Squeeze-Excitation attention module [65] with an MLP reduction ratio of 4.

Module Location	**Accuracy ↑ (%)**
None	88.12
'1'	88.21 (+0.09)
'2'	88.09 (−0.03)
'3'	88.02 (−0.10)
'4'	88.31 (+0.19)
'5'	87.74 (−0.38)

Table 6.4: Effect of attention module location on performance. An illustration of each location is provided in Figure 6.3.

We observe that attention location has a tangible impact on accuracy, with positions "1" and "4" offering the greatest benefit. Apart from these two, all other positions negatively affect model accuracy, indicating that attention methods are sensitive to

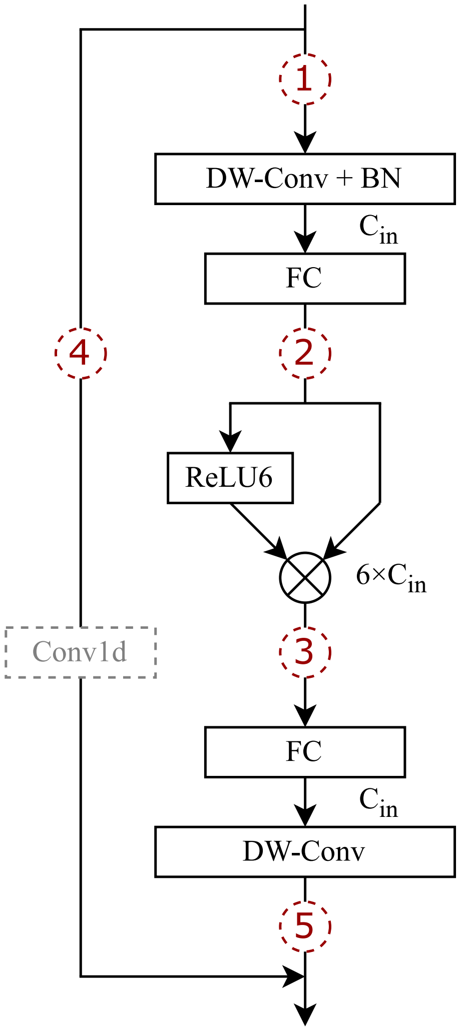

Figure 6.3: Possible locations (marked in red) for the attention modules within the Star-V block. The optional 1D convolution in the left is used to downsample the inputs in case where there is a mismatch between input and output and the same applies to location “5”. C_{in} indicates the input channel dimension of the block.

their location within the Star-V block. Positions “2” and “3” slightly lower performance and seem to hamper the block’s representational capacity as they are applied “within” the MLP network (between FC layers). The last position which is a typical placement for SE blocks in the computer vision literature has the most adverse effect on accuracy in the case of Star-V block, causing a 0.38% drop, while the first position raises it by 0.09%. This indicates that recalibrating the channels before the block's operations are applied is more favorable than using the attention module for that purpose at the end of the block. More surprisingly, the fourth position performs best, showing that an even better approach is to apply an attention module to the skip connection that is a direct pathway from the input to the output of this block. This way, the attention operates on the unchanged inputs and the resulting weighed features are not mixed with the operations of the Star-V block, preventing any possible interference of the attention module with the Hadamard product, a reasoning that is further reinforced if we consider that in positions "1" and "4" obtain the best performance.

Moving on, using location "4" as the place of attention and layer integration, we provide all hyper-parameter values used in our experiments when determining the best-performing attention modules. While not all methods share the same hyper-parameters, a common one among them is a *reduction ratio* which controls the MLP dimensions and subsequently its representational capacity and computational overhead. For the methods that use this hyper-parameter ([65], [135]), we compare different values, while [323] is also parameterized by a non-linear activation function, which directly impacts how the feature maps are scaled (e.g., Sigmoid will only output values in the range $(0, 1)$, while *tanh* allows negative values up to -1 that can effectively invert some values). For this method, the MLP reduction ratio is set to 2, as it was found to perform best (not included in the next table for brevity). Other hyper-parameters include *kernel size* ([276]), with similar effects, *normalization function* ([325]) which does not affect model size and *output scaling* for [324], which is a simple multiplication operation of a scalar with a Tensor. The results are outlined in Table 6.5.

From the above results, we observe that hyper-parameter selection can have a noticeable impact on overall performance while hardly affecting network overhead for all attention methods, regarding FLOP counts. For SE and CBAM, two methods that utilize a reduction ratio for the MLPs, higher values cause a smaller increase in parameters without raising complexity. In the case of SE, the best-performing reduction ratio is 8 or 16 as the difference in accuracy is negligible. Simultaneously, both these reduction ratios add only 0.1 million parameters to the TCN and for the latter, negligible complexity. Including the SE in the Star-V blocks is a no-cost improvement for performance. In contrast, for CBAM, any amount of MLP reduction ratio does

Method	**Hyper-parameter(s)**	**Value(s)**	**FLOPs (M)**	**Params (M)**	**Accuracy ↑ (%)**
Base TCN model (without attention)	–	–	366	12.6	88.12
Squeeze-Excitation [65]	MLP reduction ratio	4	367	13.1	88.31 (+0.19)
		8	367	12.8	88.41 (+0.29)
		16	366	12.7	88.40 (+0.28)
Convolutional Block Attention Module [135]	MLP reduction ratio	1	371	14.7	87.66 (−0.46)
		2	368	13.6	87.67 (−0.45)
		4	367	13.1	87.82 (−0.30)
		8	367	12.8	87.53 (−0.59)
		16	367	12.7	87.91 (−0.21)
Gated Channel Transformation [325]	Normalization	L_1	366	12.6	87.94 (−0.18)
		L_2	366	12.6	88.25 (+0.13)
Skip-Squeeze-Excitation [324]	Output scaling amount	1	367	13.6	88.56 (+0.44)
		2	367	13.6	88.03 (−0.09)
Shift-and-Balance Attention [323]	function	Tanh	367	13.6	87.96 (−0.16)
		Sigmoid	367	13.6	88.27 (+0.15)
		ReLU	367	13.6	87.98 (−0.14)
		Softmax	367	13.6	88.12 (±0.00)
		Linear	367	13.6	88.18 (+0.06)
Efficient Channel Attention [276]	Kernel size; Channels	3; 1	366	12.6	88.45 (+0.33)
		5; 1	366	12.6	88.66 (+0.52)
		7; 1	366	12.6	88.63 (+0.51)

Table 6.5: Effect of different values for hyper-parameters of each attention module. For FLOPs and Parameters we only measure the TCN model.

not offer a positive impact and in fact lowers the recognition rate. The highest reduction ratio adds only 0.1 million parameters (similar to SE) and performs best for this attention module, indicating that limiting the capacity of this module implicitly lowers its impact and degradation of accuracy. For GCT, using L_2 normalization is a better choice than L_1, which has a minor negative effect on word recognition. The simplicity of this module regarding its operations (it does not use any convolution layers) is highlighted as it does not affect the TCN's FLOPs or parameters in any way. Similarly, the output scaling rate for SSE performs better at 1 (no scaling), providing a $+0.5\%$ bonus, while a higher scaling ratio works against performance. The absence of a reduction ratio for this module (since it uses one single convolution layer, rather than the typical two that comprise MLPs), allows it to maintain a higher parameter count than SE but achieves a slightly better recognition rate. The activation function employed in SBA plays an important role as some functions degrade accuracy, with Sigmoid being the most suitable option for this dataset. Having no activation function (as denoted by "Linear") actually outperforms the remaining functions, showing that some non-linearities are not suitable for this block and hamper its performance. Finally, for ECA, any size of the convolution kernel offers a benefit for accuracy, with 5 being the best choice, surpassing both 3 and 7, albeit the latter by a slight margin. The lightweight design of this module adds a fraction of parameters (which is negligible in the overall count), since its convolution is applied on 1 channel which is the result of a pooling operation. This module is the most efficient in this list, achieving the largest increase in accuracy without affecting the size or complexity of the model.

6.3.2 Dropout Amount and Scheduling

Regarding regularization, its effects on the overall accuracy of the trained model cannot be neglected. A dropout probability that is too high would lead to a large percentage of neurons being disabled at every training iteration, potentially hampering the network's learning process by overly restricting the network. On the contrary, a very low probability might not affect the training process (as much as desired) and not produce a meaningful result, i.e. improve the network's generalization. We therefore study the impact of dropout on final performance, taking into account several factors. First, the values used in some published works in the VSR literature that employed dropout in the TCN serve as good indicators that have been tested in the LRW dataset, albeit with different networks. Next, we also consider the size of the dataset itself, with regards to the representational capacity of the models in our architecture. Since the networks employed in this work are much smaller than those typically used in the literature we use more conservative values for dropout, in order to prevent excessive under-fitting. Table 6.6 showcases the results.

Our experiments show that a very small amount of dropout is enough to offer a small gain to overall accuracy, at no extra cost since the network size remains the same. Compared to other published models from the literature that employ relatively high dropout probabilities (e.g., 0.2 is used in [110, 162, 194, 213] and as high as 0.5 in [165]), we find that for our lightweight TCN, smaller values perform better, with a value of 0.05 to 0.1 being the best choice. We observe that for higher values of dropout probability, performance starts to degrade, which is not unexpected and reinforces our claims in the earlier discussion.

The above experiments use a dropout rate that remains non-changing during training. A logical choice is to adjust the probability of dropout as training progresses,

Dropout p	Accuracy ↑ (%)
0.0	88.12
0.05	88.26 (+0.14)
0.1	88.24 (+0.12)
0.2	88.00 (−0.12)
0.25	87.68 (−0.44)

Table 6.6: Ablation study on the amount of dropout added to the layers of the TCN. The strategy used is constant rate throughout training. We use the baseline end-to-end model without any additional components such as attention methods. Results are on the LRW test set.

akin to learning rate scheduling. We therefore experiment with dropout strategies borrowed from [326] that have shown potential for performance improvement by employing a dropout schedule that activates at certain epochs and modifies the probability. As in the above experiments, we use the baseline end-to-end network without any attention modules. The results are shown in Table 6.7.

Our study showcases that regularization strategies should be applied in a careful manner as smaller networks are more sensitive to noise. When the dropout probability is rather low at 0.05, the "Early" strategy that keeps it activated until the specified epoch is finished, is actually detrimental to performance and a similar effect happens when we use a higher rate at 0.1. This suggests that for lightweight networks the early epochs of training are rather important for final performance and that using dropout during these epochs harms the overall capabilities of the network. This claim is reinforced by the results for the "Late" strategy that activates dropout after the target epoch is reached, allowing the network to train for a few epochs without regularization. In this strategy, a higher value for dropout does improve accuracy, if activated near the final epochs of training (e.g., after epoch 40), while the opposite happens for a smaller dropout value. A probable explanation is that a larger value assists the network in generalization by forcing adaptation after some features have been learned, but a smaller dropout probability does not activate as often and acts as noise, disrupting the learning process. Another supporting result of this assumption is the model behavior when dropout is activated after epoch 20, where the drop rate of 0.1 prevents learning of more fine-grained features as it acts on the model for

Dropout schedule	Probability	Epoch	Accuracy ↑ (%)
No dropout	-	-	88.12
Early	0.05	5	88.06 (−0.06)
		10	88.06 (−0.06)
		20	88.08 (−0.04)
	0.1	5	87.97 (−0.15)
		10	87.77 (−0.35)
		20	88.09 (−0.03)
Late	0.05	20	88.14 (+0.02)
		40	88.09 (−0.03)
		60	87.88 (−0.24)
	0.1	20	88.08 (−0.04)
		40	88.12 (±0.00)
		60	88.28 (+0.16)

Table 6.7: Ablation study on the scheduling strategy for the dropout rate during training. A "late" schedule enables dropout after a certain epoch, while the "early" schedule deactivates it after that epoch, indicated in the third column. Results are on the LRW test set.

longer, but 0.05 is not as counter-productive to the learning process.

6.3.3 Combination of Methods

Finally, we offer a complete ablative study that highlights the importance and effect of each particular module on the final performance of the model, combining all previous methods discussed in this Chapter.

While each method in isolation provides a small benefit for overall performance, our results reveal that a combination of methods is the best approach, allowing the end-to-end model to reach a recognition level that is higher than using each method individually. The effects of regularization have a positive impact not only on the baseline model, but also on the one using ECA modules. This synergistic effect occurs for both dropout application strategies, as each one contributes to a higher accuracy without any additional overhead, suggesting that the regularization effect of dropout

Method	**Accuracy ↑ (%)**
Base model (MobileNetV4-S + Star-V block TCN)	88.12
+ Dropout $p = 0.05$, no schedule	88.26 ($+0.14$)
+ Dropout $p = 0.1$, late ($e = 60$)	88.28 ($+0.16$)
+ ECA attention (kernel size = 5)	88.66 ($+0.54$)
+ Dropout $p = 0.05$ + ECA (kernel size = 5)	88.92 ($+0.80$)
+ Dropout $p = 0.1$, late ($e = 60$) + ECA (kernel size = 5)	88.93 ($+0.81$)

Table 6.8: Ablation study on the different performance-improving methods considered in this Chapter. Results are on the LRW test set.

enhances learning regardless of model size and complexity. For this architecture, the late activation strategy slightly outperforms using a static dropout rate (by 0.01%), with and without the use of an attention module, although the difference between these two is negligible.

6.4 Conclusion

In this Chapter, we explored methods for improving the recognition performance of a lightweight VSR end-to-end network for isolated words. These networks tend to employ resource-saving components with weaker representational capacity, and as a result achieve lower accuracy than larger architectures. An existing architecture was employed and augmented with channel attention modules that were designed to improve performance without adding significant overhead to network size and complexity. Several such modules were introduced in the temporal blocks of the sequence modeling component and experiments in the largest dataset for word recognition in English were performed. Next, regularization in the form of dropout was added during training of the model to facilitate enhancing the learning process and to reduce the amount of under-fitting. Rather than keeping the dropout probability for the whole training duration static, scheduling strategies that modify the dropout rate according to the current epoch were investigated. Ablation studies were carried out to determine the most optimal locations and hyperparameters for the attention module as well as the values and thresholds for dropout and its scheduling strategies. Our results demonstrate that without increasing the network's computational overhead, incorporating an attention module and introducing regularization bring small improvements in recognition accuracy, however, careful selection of hyperparameters is important. When combined, these two methods cumulatively uplift performance, achieving significant raises.

Chapter 7

Conclusion

This dissertation focuses on the task of visual speech recognition (VSR) from a practical perspective. Applications of VSR are found in numerous domains covering several facets of every day life and can offer significant, even life-changing benefits for a great number of people. The overwhelming majority of research in the literature for this task is aimed at improving recognition rates, disregarding the resource overhead of the oftentimes complex models that are proposed. This in turn effectively prevents further exploitation in realistic scenarios since the hardware requirements can only be met in specific conditions, typically in high compute environments. In contrast, the methods proposed in the previous Chapters represent several approaches towards the goal of developing lightweight and practical models for VSR with broad application potential. The resulting models have low hardware requirements, enabling deployment in a wide spectrum of scenarios by more energy-efficient devices. These factors are gaining increasing importance as energy consumption is becoming a concern that affects a great number of individuals at a global scale. The main contributions of this thesis are summarized as follows.

Chapter 2 presents a comprehensive review of the literature on VSR of isolated words in the English language, covering over 140 published works with a basis in deep learning methods. A common methodology shared by all methods is splitting the problem in two smaller tasks: visual feature extraction and sequential modeling. Since CNNs are invariably employed for the former, we offer a basic taxonomy of works according to the model used for the latter task. The review also underscores that a disproportionate amount of published works focuses on the topic of practical VSR, which serves as a motivation of this dissertation. In the last part of the chapter we also present a through analysis of published datasets for isolated word VSR, categorized by language, that includes technical information about the samples for more than 30 datasets.

In Chapter 3 we explored a network compression technique to lower an existing architecture's overall size and memory footprint. We replace standard convolution and fully-connected layers in the base models with equivalent layers that exploit a sum of Kronecker products to produce their weight matrices used in calculations. The use of Kronecker product in this fashion allows using matrices of smaller dimensions, achieving significant parameter savings. A user-defined hyper-parameter specifies the number of Kronecker products to be summed and ultimately controls the amount of parameter reductions, or network compression. Our experiments show that large reductions in model sizes are possible but there is a small penalty in recognition rate, which becomes more pronounced at higher rates of compression due to the reduction in network capacity. An ablative analysis explores the impact of using parameter-saving layers in the different components of the end-to-end architecture.

In the same spirit, in Chapter 4 we presented a different method for network compression which also reduces computational overhead of the unified architectures. This method involves a module that generates intermediate feature maps of smaller dimensions and then applies an inexpensive operation on that result. This module is used in place of standard convolutions in both components of a two-stage VSR design, creating cost-efficient networks with reduced requirements. Furthermore, we proposed a temporal block design called *Partial Temporal Block* that splits the input feature maps in two parallel computation branches across the channel dimension. This block is highly customizable, allowing different operations to be applied in each branch, and can considerably lower the overhead of a network. Using our block as a core component in a TCN, we evaluate three lightweight designs from the literature developing sequence modeling networks that are rather compact and low in complexity.

A systematic approach is followed in Chapter 5 where rather than using an existing network, we investigate a multitude of structures to develop a lightweight and powerful end-to-end architecture. Adopting the ubiquitous two-step pipeline that is typical for VSR, we first explore convolutional neural networks for extraction of visual features from the input. We benchmark several lightweight CNN-based models from the image classification literature that cover various approaches for network design, finding the one that performs best for the task. Next, using that model for visual feature extraction, we design TCN-based sequential models that utilize a variety of temporal block structures. We adapt the two-dimensional blocks to the 1D sub-task of sequential modeling for use in our TCNs by converting the appropriate operations, creating various networks with distinct block design philosophies, that are also benchmarked. The most performant components for both sub-tasks are employed in a unified model that far surpasses other lightweight networks from the literature in all measurement indices, being smaller in size, lighter in overhead and achieving word accuracy that is comparable to other, much larger models.

Finally, using our end-to-end model as a baseline, in Chapter 6 we explore methods to enhance its capabilities while maintaining its computational complexity at manageable levels. A mechanism that aligns with this objective is channel attention, therefore we add such methods to the temporal blocks of the TCN model that is used for sequence modeling. We employ several attention mechanisms from the computer vision literature that are adapted where necessary to operate on the channel dimension of the visual features that are fed to the TCN. In addition, we introduce regularization to the unified model during training in the form of dropout added after each temporal block to improve the overall recognition rates without any additional cost in model overhead. Rather than using a constant rate throughout the duration of training, several strategies that modify the dropout probability are investigated. Ablative experiments show the optimal location and hyperparameters for the attention mechanisms as well as the dropout scheduling strategy that is most beneficial for performance. A combination of channel attention and regularization is shown to provide the largest increase in accuracy.

7.1 Insights

The insights gained from the work carried out in this dissertation regarding lightweight and practical model design for VSR highlight a few key aspects that should be taken into account when designing such systems where the aim is a practical application. First and foremost, the two-stage design is a tried and tested method for the task of VSR that has stood the test of time and currently is the most efficient approach for practical networks. This design also applies if the objective is solely obtaining a high recognition rate, i.e., without considering the computational costs. Since the most resource-heavy component is typically the visual feature extraction network, a lightweight end-to-end system should employ a cost-effective model and several lightweight models from the computer vision literature have been shown to perform well regardless of their compact size and are strong candidates for this purpose. Similarly, the sequence modeling component is an essential part of the overall architecture and TCN-based models are currently the most suitable choice as they offer strong performance with low hardware demands, compared to RNNs. Moreover, their structure has a high design flexibility that allows designing customized models tailored to the available resources. An inevitable result of designing practical networks with reduced hardware demands is a reduction in network capacity due to the lightweight operations and smaller sizes, which translates into a degradation of performance, however this effect can be mitigated with various methods, such as using a high-performance TCN-based model, or introducing attention methods and regularization to improve recognition performance. Naturally, the impact on performance is greater the more compact a model becomes and more often than not, a compromise between accuracy and size has to be made, nevertheless, as research progresses, the gap between compact and larger networks is becoming narrower.

7.2 Future Work

A few notable limitations are underlined at each Chapter, which serve as motivation for future research directions. Improving the accuracy of our lightweight models is a priority, since from our experiments it is evident that the lighter and more practical networks fall behind the larger ones in terms of performance. While this is not unexpected, we believe that extending our proposed models with more, newer architectures or devising more specific training strategies can aid our efforts towards bridging that gap. Expanding the principles of this work to phrase- or sentence-level VSR is also an interesting direction as these models are hampered by the same constraints discussed in Chapter 2, namely, deployment difficulties due to the considerable size and complexity of the involved architectures. Recognition of structured speech remains an open problem that comes with its own set of challenges and expands the benefits of VSR in a broader scope of communication. Enabling more applications of VSR will assist individuals and offer more services in scenarios that are not satisfied by decoding isolated words.

7.3 Concluding Remarks

In closing, we discuss the ethics of VSR technology, and SR in general, which can be used for either benevolent or malevolent purposes. A valid concern is that VSR systems can be used to violate the privacy of users through surveillance, which applies to all forms of VSR, but is especially applicable to more sophisticated recognition systems that are able to decode structured speech such as phrases or sentences. Currently, the methods for VSR are limited in their applications due to a few factors, mainly by data availability and method generalization. The available datasets typically contain videos of a low spatial resolution and are overall constrained both in the number of samples and the richness of vocabulary, preventing training of highly capable (more realistic) recognizers. Presently this is not a concerning issue, however in the future it will become a more pressing matter, given that the daily rates of data recording and technological progress are constantly increasing. Furthermore, the performance of current VSR methods is evaluated on curated data depicting unobstructed views of speakers in semi-controlled conditions that can be regarded as ideal and deteriorates rapidly with variations of speaker appearance, head pose and other occlusions (e.g., wearing masks), making it an unlikely vector for malicious intent, when compared to ASR. Indeed, the latter is more likely to be utilized for eavesdropping or surveillance purposes, since the performance of VSR is closely dependent on video quality, while ASR is unaffected. Preventative legislative measures and frameworks have to be developed in a timely manner, laying the groundwork for future regulations regarding all forms of SR to protect the privacy of individuals, mitigating potential risks.

In closing, we believe that the benefits of VSR far outweigh any potential malicious application and that it will be used for good rather than harm.

Bibliography

[1] T. F. Cootes, G. J. Edwards, and C. J. Taylor, “Active Appearance Models,” in *Proceedings of the 5th European Conference on Computer Vision*. Springer Berlin Heidelberg, 1998, pp. 484–498.

[2] I. Matthews, T. F. Cootes, J. A. Bangham, S. Cox, and R. Harvey, “Extraction of Visual Features for Lipreading,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 198–213, 2002.

[3] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. W. Senior, “Recent Advances in the Automatic Recognition of Audiovisual Speech,” *Proceedings of the IEEE*, vol. 91, no. 9, pp. 1306–1326, 2003.

[4] A. Fernandez-Lopez and F. M. Sukno, “Survey on Automatic Lip-Reading in the Era of Deep Learning,” *Image and Vision Computing*, vol. 78, pp. 53–72, 2018.

[5] K. Kumar, T. Chen, and R. M. Stern, “Profile View Lip Reading,” in *Proceedings of the 32nd IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 4. IEEE, April 15–20 2007, pp. IV–429.

[6] T. Saitoh and R. Konishi, “Profile Lip Reading for Vowel and Word Recognition,” in *Proceedings of the 20th International Conference on Pattern Recognition*. IEEE, August 23–26 2010, pp. 1356–1359.

[7] Z. Zhou, G. Zhao, X. Hong, and M. Pietikäinen, “A Review of Recent Advances in Visual Speech Decoding,” *Image and Vision Computing*, vol. 32, no. 9, pp. 590–605, 2014.

[8] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata, “Lipreading Using Convolutional Neural Network,” in *Proceedings of the INTERSPEECH*. ISCA, September 14–18 2014.

[9] ——, “Audio-Visual Speech Recognition Using Deep Learning,” *Applied Intelligence*, vol. 42, pp. 722–737, 2015.

[10] Y. Takashima, Y. Kakihara, R. Aihara, T. Takiguchi, Y. Ariki, N. Mitani, K. Omori, and K. Nakazono, “Audio-Visual Speech Recognition Using Convolutive Bottleneck Networks for a Person with Severe Hearing Loss,” *IPSJ Transactions on Computer Vision and Applications*, vol. 7, pp. 64–68, 2015.

[11] Y. Li, Y. Takashima, T. Takiguchi, and Y. Ariki, “Lip Reading Using a Dynamic Feature of Lip Images and Convolutional Neural Networks,” in *Proceedings of the 15th IEEE/ACIS International Conference on Computer and Information Science*. IEEE, June 26–29 2016, pp. 1–6.

[12] M. Schuster and K. K. Paliwal, “Bidirectional Recurrent Neural Networks,” *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet Classification with Deep Convolutional Neural Networks,” *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[14] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, “Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion,” *Journal of Machine Learning Research*, vol. 11, no. 12, pp. 3371–3408, December 2010.

[15] Y. Takashima, R. Aihara, T. Takiguchi, Y. Ariki, N. Mitani, K. Omori, and K. Nakazono, “Audio-Visual Speech Recognition Using Bimodal-Trained Bottleneck Features for a Person with Severe Hearing Loss,” in *Proceedings of the INTERSPEECH*. ISCA, September 8–12 2016, pp. 277–281.

[16] K. Veselý, M. Karafiát, and F. Grézl, “Convolutive Bottleneck Network Features for LVCSR,” in *Proceedings of the 7th IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE, December 11–15 2011, pp. 42–47.

[17] D. Cristinacce and T. Cootes, “Feature Detection and Tracking with Constrained Local Models,” in *Proceedings of the 17th British Machine Vision Conference*, vol. 3. BMVA Press, September 4–7 2006, pp. 929–938.

[18] M. Wand, J. Koutník, and J. Schmidhuber, "Lipreading with Long Short-Term Memory," in *Proceedings of the 41st IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, March 20–25 2016, pp. 6115–6119.

[19] C. Bregler and Y. Konig, """Eigenlips" for Robust Speech Recognition," in *Proceedings of the 19th IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. ii. IEEE, April 19–22 1994, pp. II/669–II/672 vol.2.

[20] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proceedings of the 18th IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, June 20–25 2005, pp. 886–893 vol. 1.

[21] J. S. Chung and A. Zisserman, "Lip Reading in the Wild," in *Proceedings of the 13th Asian Conference on Computer Vision*. Springer, November 20–24 2017, pp. 87–103.

[22] ——, "Learning to Lip Read Words by Watching Videos," *Computer Vision and Image Understanding*, vol. 173, pp. 76–85, 2018.

[23] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the Devil in the Details: Delving Deep Into Convolutional Nets," in *Proceedings of the 25th British Machine Vision Conference*. British Machine Vision Association, September 1–5 2014.

[24] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Lip Reading Sentences in the Wild," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, July 21–26 2017, pp. 3444–3453.

[25] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-Based Models for Speech Recognition," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[26] T. Stafylakis and G. Tzimiropoulos, "Combining Residual Networks with LSTMs for Lipreading," in *Proceedings of the INTERSPEECH*. ISCA, 2017, pp. 3652–3656.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," in *Proceedings of the 14th European Conference on Computer Vision*. Springer, October 11–14 2016, pp. 630–645.

[28] T. Stafylakis and G. Tzimiropoulos, "Deep Word Embeddings for Visual Speech Recognition," in *Proceedings of the 43rd IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, April 15–20 2018, pp. 4974–4978.

[29] S. Ioffe, "Probabilistic Linear Discriminant Analysis," in *Proceedings of the 9th European Conference on Computer Vision*. Springer Berlin Heidelberg, May 2006, pp. 531–542.

[30] T. Stafylakis, M. H. Khan, and G. Tzimiropoulos, "Pushing the Boundaries of Audiovisual Word Recognition Using Residual Networks and LSTMs," *Computer Vision and Image Understanding*, vol. 176, pp. 22–32, 2018.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 26–July 1 2016, pp. 770–778.

[32] Y. Lu and H. Li, "Automatic Lip-Reading System Based on Deep Convolutional Neural Network and Attention-Based Long Short-Term Memory," *Applied Sciences*, vol. 9, no. 8, p. 1599, April 2019.

[33] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-Based LSTM for Aspect-Level Sentiment Classification," in *Proceedings of the 21st Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, November 1–5 2016, pp. 606–615.

[34] C. T. Huyen, "German Word Level Lip Reading with Deep Learning," BsC Thesis, Hochschule für angewandte Wissenschaften Hamburg, Germany, June 4 2019.

[35] L. Courtney and R. Sreenivas, "Using Deep Convolutional LSTM Networks for Learning Spatiotemporal Features," in *Proceedings of the 1st Asian Conference on Pattern Recognition*. Springer International Publishing, November 26–29 2019, pp. 307–320.

[36] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting,” *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[37] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[38] C. Wang, “Multi-Grained Spatio-Temporal Modeling for Lip-Reading,” in *Proceedings of the 30th British Machine Vision Conference*. BMVA Press, September 9–12 2019, p. 276.

[39] S. Yang, Y. Zhang, D. Feng, M. Yang, C. Wang, J. Xiao, K. Long, S. Shan, and X. Chen, “LRW-1000: A Naturally-Distributed Large-Scale Benchmark for Lip Reading in the Wild,” in *Proceedings of the 14th IEEE International Conference on Automatic Face & Gesture Recognition*. IEEE, May 14–18 2019, pp. 1–8.

[40] D. Kastaniotis, D. Tsourounis, A. Koureleas, B. Peev, C. Theoharatos, and S. Fotopoulos, “Lip Reading in Greek Words at Unconstrained Driving Scenario,” in *Proceedings of the 10th International Conference on Information, Intelligence, Systems and Applications*. IEEE, July 15–17 2019, pp. 1–6.

[41] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate,” in *Proceedings of the 3rd International Conference on Learning Representations*, May 7–9 2015.

[42] D. Parekh, A. Gupta, S. Chhatpar, A. Yash, and M. Kulkarni, “Lip Reading Using Convolutional Auto Encoders As Feature Extractor,” in *Proceedings of the 5th IEEE International Conference for Convergence in Technology*. IEEE, March 29–31 2019, pp. 1–6.

[43] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, “Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction,” in *Proceedings of the 21st International Conference on Artificial Neural Networks*. Springer, June 14–17 2011, pp. 52–59.

[44] X. Weng and K. Kitani, “Learning Spatio-Temporal Features with Two-Stream Deep 3D CNNs for Lipreading,” in *Proceedings of the 30th British Machine Vision Conference*. BMVA Press, September 9–12 2019.

[45] J. Carreira and A. Zisserman, “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset,” in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, July 21–26 2017, pp. 6299–6308.

[46] M. Riva, M. Wand, and J. Schmidhuber, “Motion Dynamics Improve Speaker-Independent Lipreading,” in *Proceedings of the 45th IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, May 4–8 2020, pp. 4407–4411.

[47] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, July 21–26 2017, pp. 4700–4708.

[48] X. Chen, J. Du, and H. Zhang, “Lipreading with DenseNet and resBi-LSTM,” *Signal, Image and Video Processing*, vol. 14, pp. 981–989, January 2020.

[49] X. Zhou, J. Li, and X. Zhou, “Cascaded CNN-resBiLSTM-CTC: an End-to-End Acoustic Model for Speech Recognition,” 2018. [Online]. Available: https://arxiv.org/abs/1810.12001

[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is All You Need,” *Advances in Neural Information Processing Systems*, vol. 30, December 4–9 2017.

[51] M. Wand and J. Schmidhuber, “Fusion Architectures for Word-Based Audiovisual Speech Recognition,” in *Proceedings of the INTERSPEECH*. ISCA, October 25–29 2020, pp. 3491–3495.

[52] M. Wand, J. Schmidhuber, and N. T. Vu, “Investigations on End-To-End Audiovisual Fusion,” in *Proceedings of the 43rd IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, April 15–18 2018, pp. 3041–3045.

[53] Y. Zhao, C. Ma, Z. Feng, and M. Song, “Speech Guided Disentangled Visual Representation Learning for Lip Reading,” in *Proceedings of the 23rd International Conference on Multimodal Interaction*, ser. ICMI ’21. Association for Computing Machinery, October 18–22 2021, pp. 687–691.

[54] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, “Utterance-Level Aggregation for Speaker Recognition in the Wild,” in *Proceedings of the 44th International*

Conference on Acoustics, Speech and Signal Processing. IEEE, May 12–17 2019, pp. 5791–5795.

[55] S. Bhaskar and T. Thasleema, "CNN Based Feature Extraction for Visual Speech Recognition in Malayalam," in *Proceedings of Data Analytics and Management*. Springer Singapore, June 26 2021, pp. 1–8.

[56] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," in *Proceedings of the 28th IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 7–12 2015, pp. 1–9.

[57] R. Zhang, M. Chen, B. Steeper, Y. Li, Z. Yan, Y. Chen, S. Tao, T. Chen, H. Lim, and C. Zhang, "SpeeChin: A Smart Necklace for Silent Speech Recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 4, pp. 1–23, December 2021.

[58] W. Dweik, S. Altorman, and S. Ashour, "Read My Lips: Artificial Intelligence Word-Level Arabic Lipreading System," *Egyptian Informatics Journal*, vol. 23, no. 4, pp. 1–12, 2022.

[59] Ü. Atila and F. Sabaz, "Turkish Lip-Reading Using Bi-LSTM and Deep Learning Models," *Engineering Science and Technology, an International Journal*, vol. 35, p. 101206, 2022.

[60] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proceedings of the 3rd International Conference on Learning Representations*, May 7–9 2015, pp. 1–14.

[61] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, July 21–26 2017, pp. 1251–1258.

[62] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," in *Proceedings of the 31st IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 18–23 2018, pp. 6848–6856.

[63] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "MnasNet: Platform-Aware Neural Architecture Search for Mobile," in *Proceedings of the 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 15–20 2019, pp. 2820–2828.

[64] D. Ivanko, D. Ryumin, A. Kashevnik, A. Axyonov, and A. Karnov, "Visual Speech Recognition in a Driver Assistance System," in *Proceedings of the 30th European Signal Processing Conference*. IEEE, August 29–September 2 2022, pp. 1131–1135.

[65] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 2011–2023, April 2019.

[66] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond Empirical Risk Minimization," in *Proceedings of the 6th International Conference on Learning Representations*, April 30–May 3 2018. [Online]. Available: https://openreview.net/forum?id=r1Ddp1-Rb

[67] D. Ivanko, D. Ryumin, and M. Markitantov, "End-to-End Visual Speech Recognition for Human-Robot Interaction," in *Proceedings of the 4th International Conference ON Modernization, Innovations, Progress: Advanced Technologies in Material Science, Mechanical and Automation Engineering*. AIP, April 12–30 2022, pp. 82–90.

[68] Y. Kartynnik, A. Ablavatski, I. Grishchenko, and M. Grundmann, "Real-Time Facial Surface Geometry from Monocular Video on Mobile Gpus," in *Proceedings of the 3rd CVPR Workshop on Computer Vision for Augmented and Virtual Reality*, June 17 2019.

[69] D. Feng, S. Yang, and S. Shan, "An Efficient Software for Building Lip Reading Models Without Pains," in *Proceedings of the 10th IEEE International Conference on Multimedia and Expo Workshops*. IEEE, July 5–9 2021, pp. 1–2.

[70] M. A. Haq, S.-J. Ruan, W.-J. Cai, and L. P.-H. Li, "Using Lip Reading Recognition to Predict Daily Mandarin Conversation," *IEEE Access*, vol. 10, pp. 53 481–53 489, 2022.

[71] D. Ryumin, D. Ivanko, and E. Ryumina, “Audio-Visual Speech and Gesture Recognition by Sensors of Mobile Devices,” *Sensors*, vol. 23, no. 4, p. 2284, February 2023.

[72] D. Liu, Z. Wang, L. Wang, and L. Chen, “Multi-Modal Fusion Emotion Recognition Method of Speech Expression Based on Deep Learning,” *Frontiers in Neurorobotics*, vol. 15, p. 697634, July 2021.

[73] J. Peymanfard, V. Saeedi, M. R. Mohammadi, H. Zeinali, and N. Mozayani, “Leveraging Visemes for Better Visual Speech Representation and Lip Reading,” 2023. [Online]. Available: https://arxiv.org/abs/2307.10157

[74] B. Shi, W.-N. Hsu, K. Lakhotia, and A. Mohamed, “Learning Audio-Visual Speech Representation by Masked Multimodal Cluster Prediction,” in *Proceedings of the 10th International Conference on Learning Representations*, April 25–29 2022. [Online]. Available: https://openreview.net/forum?id=Z1Qlm11uOM

[75] G. Xing, L. Han, Y. Zheng, and M. Zhao, “Application of Deep Learning in Mandarin Chinese Lip-Reading Recognition,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2023, no. 1, p. 90, September 2023.

[76] S. Sukhbaatar, J. Weston, R. Fergus *et al.*, “End-to-End Memory Networks,” *Advances in Neural Information Processing Systems*, vol. 28, December 7–12 2015.

[77] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and Their Compositionality,” *Advances in Neural Information Processing Systems*, vol. 26, December 5–10 2013.

[78] S. Rudregowda, S. Patilkulkarni, V. Ravi, H. Gururaj, and M. Krichen, “Audio-visual Speech Recognition Based on a Deep Convolutional Neural Network,” *Data Science and Management*, vol. 7, no. 1, pp. 25–34, October 2023.

[79] T. Exarchos, G. N. Dimitrakopoulos, A. G. Vrahatis, G. Chrysovitsiotis, Z. Zachou, and E. Kyrodimos, “Lip-Reading Advancements: A 3D Convolutional Neural Network/Long Short-Term Memory Fusion for Precise Word Recognition,” *BioMedInformatics*, vol. 4, no. 1, pp. 410–422, February 2024.

[80] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” in *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 27–30 2016, pp. 2818–2826.

[81] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in *Proceedings of the 31st IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 18–23 2018, pp. 4510–4520.

[82] D. Ryumin, A. Axyonov, E. Ryumina, D. Ivanko, A. Kashevnik, and A. Karpov, “Audio-Visual Speech Recognition Based on Regulated Transformer and Spatio-Temporal Fusion Strategy for Driver Assistive Systems,” *Expert Systems with Applications*, p. 124159, May 2024.

[83] D. Ivanko, A. Axyonov, D. Ryumin, A. Kashevnik, and A. Karpov, “Multi-Speaker Audio-Visual Corpus RUSAVIC: Russian Audio-Visual Speech in Cars,” in *Proceedings of the 13th Conference on Language Resources and Evaluation*. European Language Resources Association, June 20–25 2022, pp. 1555–1559.

[84] S. Petridis, T. Stafylakis, P. Ma, F. Cai, G. Tzimiropoulos, and M. Pantic, “End-to-End Audiovisual Speech Recognition,” in *Proceedings of the 43rd IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, April 15–20 2018, pp. 6548–6552.

[85] Y. Zhao, R. Xu, and M. Song, “A Cascade Sequence-to-Sequence Model for Chinese Mandarin Lip Reading,” in *Proceedings of the 1st ACM International Conference on Multimedia in Asia*, ser. MMAsia ’19. Association for Computing Machinery, December 2019, pp. 1–6.

[86] X. Zhang, H. Gong, X. Dai, F. Yang, N. Liu, and M. Liu, “Understanding Pictograph with Facial Features: End-to-End Sentence-Level Lip Reading of Chinese,” in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, January 27–February 1 2019, pp. 9211–9218.

[87] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, “Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks,” *Advances in Neural Information Processing Systems*, vol. 28, pp. 1171–1179, December 7–12 2015.

[88] Y. M. Assael, B. Shillingford, S. Whiteson, and N. de Freitas, "Lipnet: Sentence-Level Lipreading," 2016. [Online]. Available: https://arxiv.org/abs/1611.01599

[89] S. Cheng, P. Ma, G. Tzimiropoulos, S. Petridis, A. Bulat, J. Shen, and M. Pantic, "Towards Pose-Invariant Lip-Reading," in *Proceedings of the 45th IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, May 4–8 2020, pp. 4357–4361.

[90] V. Blanz and T. Vetter, "Face Recognition Based on Fitting a 3D Morphable Model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1063–1074, 2003.

[91] Y. Zhang, S. Yang, J. Xiao, S. Shan, and X. Chen, "Can We Read Speech Beyond the Lips? Rethinking RoI Selection for Deep Visual Speech Recognition," in *Proceedings of the 15th IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE, November 16–20 2020, pp. 356–363.

[92] T. DeVries and G. W. Taylor, "Improved Regularization of Convolutional Neural Networks with Cutout," 2017. [Online]. Available: https://arxiv.org/abs/1708.04552

[93] Z. Miao, H. Liu, and B. Yang, "Part-Based Lipreading for Audio-Visual Speech Recognition," in *Proceedings of the 7th IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, October 11–14 2020, pp. 2722–2726.

[94] M. Luo, S. Yang, S. Shan, and X. Chen, "Pseudo-Convolutional Policy Gradient for Sequence-to-Sequence Lip-Reading," in *Proceedings of the 15th IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE, November 16–20 2020, pp. 273–280.

[95] H. Liu, Z. Chen, and B. Yang, "Lip Graph Assisted Audio-Visual Speech Recognition Using Bidirectional Synchronous Fusion," in *Proceedings of the INTERSPEECH*. ISCA, October 25–29 2020, pp. 3520–3524.

[96] S. Yan, Y. Xiong, and D. Lin, "Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, February 2–7 2018.

[97] X. Zhao, S. Yang, S. Shan, and X. Chen, "Mutual Information Maximization for Effective Lip Reading," in *Proceedings of the 15th IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE, November 16–20 2020, pp. 420–427.

[98] J. Xiao, S. Yang, Y. Zhang, S. Shan, and X. Chen, "Deformation Flow Based Two-Stream Network for Lip Reading," in *Proceedings of the 15th IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE, November 16–20 2020, pp. 364–370.

[99] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," in *Proceedings of the NIPS 2014 Deep Learning Workshop*, December 12–13 2014.

[100] B. Xu, J. Wang, C. Lu, and Y. Guo, "Watch to Listen Clearly: Visual Speech Enhancement Driven Multi-Modality Speech Recognition," in *Proceedings of the 20th IEEE/CVF Winter Conference on Applications of Computer Vision*. IEEE, March 1–5 2020, pp. 1637–1646.

[101] B. Xu, C. Lu, Y. Guo, and J. Wang, "Discriminative Multi-Modality Speech Recognition," in *Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 14–19 2020, pp. 14 433–14 442.

[102] Z. Qiu, T. Yao, and T. Mei, "Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks," in *Proceedings of the 16th IEEE International Conference on Computer Vision*, October 22–29 2017, pp. 5533–5541.

[103] P. Zhang, J. Xue, C. Lan, W. Zeng, Z. Gao, and N. Zheng, "ElAatt-RNN: Adding Attentiveness to Neurons in Recurrent Neural Networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 1061–1073, 2019.

[104] H. Liu, Y. Wang, and B. Yang, "Mutual Alignment Between Audiovisual Features for End-to-End Audiovisual Speech Recognition," in *Proceedings of the 25th International Conference on Pattern Recognition*. IEEE, January 10–15 2021, pp. 5348–5353.

[105] H. Liu, W. Li, and B. Yang, "Robust Audio-Visual Speech Recognition Based on Hybrid Fusion," in *Proceedings of the 25th International Conference on Pattern Recognition*. IEEE, January 10–15 2021, pp. 7580–7586.

[106] H. Liu, W. Xu, and B. Yang, "Audio-Visual Speech Recognition Using a Two-Step Feature Fusion Strategy," in *Proceedings of the 25th International Conference on Pattern Recognition*. IEEE, January 10–15 2021, pp. 1896–1903.

[107] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-Local Neural Networks," in *Proceedings of the 31st IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 18-23 2018, pp. 7794–7803.

[108] M. Hao, M. Mamut, N. Yadikar, A. Aysa, and K. Ubul, "How to Use Time Information Effectively? Combining with Time Shift Module for Lipreading," in *Proceedings of the 46th IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, June 6–11 2021, pp. 7988–7992.

[109] J. Lin, C. Gan, and S. Han, "TSM: Temporal Shift Module for Efficient Video Understanding," in *Proceedings of the 17th IEEE/CVF International Conference on Computer Vision*. IEEE, October 27–November 02 2019, pp. 7083–7093.

[110] B. Martinez, P. Ma, S. Petridis, and M. Pantic, "Lipreading Using Temporal Convolutional Networks," in *Proceedings of the 45th IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, May 4–8 2020, pp. 6319–6323.

[111] H. Li, M. Mamut, N. Yadikar, Y. Zhu, and K. Ubul, "Channel Enhanced Temporal-Shift Module for Efficient Lipreading," in *Proceedings of the 15th Chinese Conference on Biometric Recognition*. Springer, September 10–12 2021, pp. 474–482.

[112] M. K. Tellamekala, M. Valstar, M. Pound, and T. Giesbrecht, "Audio-Visual Predictive Coding for Self-Supervised Visual Representation Learning," in *Proceedings of the 25th International Conference on Pattern Recognition*. IEEE, January 10–15 2021, pp. 9912–9919.

[113] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation Learning with Contrastive Predictive Coding," 2018. [Online]. Available: https://arxiv.org/abs/1807.03748

[114] E. Egorov, V. Kostyumov, M. Konyk, and S. Kolesnikov, "LRWR: Large-Scale Benchmark for Lip Reading in Russian Language," 2021. [Online]. Available: https://arxiv.org/abs/2109.06692

[115] D. Tran, H. Wang, L. Torresani, and M. Feiszli, "Video Classification with Channel-Separated Convolutional Networks," in *Proceedings of the 17th IEEE/CVF International Conference on Computer Vision*. IEEE, October 27–November 2 2019, pp. 5552–5561.

[116] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A Closer Look at Spatiotemporal Convolutions for Action Recognition," in *Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 18–23 2018, pp. 6450–6459.

[117] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, M. Li, and A. Smola, "ResNeSt: Split-Attention Networks," in *Proceedings of the 35th IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, June 19–20 2022, pp. 2735–2745.

[118] B. Sen, A. Agarwal, R. Mukhopadhyay, V. Namboodiri, and C. Jawahar, "Personalized One-Shot Lipreading for an Als Patient," in *Proceedings of the 32nd British Machine Vision Conference*. BMVA Press, November 22–25 2021, p. 428.

[119] K. R. Prajwal, R. Mukhopadhyay, V. P. Namboodiri, and C. Jawahar, "A Lip Sync Expert is All You Need for Speech to Lip Generation in the Wild," in *Proceedings of the 28th ACM International Conference on Multimedia*, ser. MM '20. Association for Computing Machinery, 2020, pp. 484–492.

[120] H. Mabrouk, O. Abugabal, N. Sakr, and H. M. Eraqi, "Lip-Listening: Mixing Senses to Understand Lips Using Cross Modality Knowledge Distillation for Word-Based Models," 2022. [Online]. Available: https://arxiv.org/abs/2207.05692

[121] M. Kim, J. Hong, S. J. Park, and Y. M. Ro, "Multi-Modality Associative Bridging Through Memory: Speech Sound Recollected From Face Video," in *Proceedings of the 18th IEEE/CVF International Conference on Computer Vision*. IEEE, October 11–17 2021, pp. 296–306.

[122] J. Weston, S. Chopra, and A. Bordes, "Memory Networks," in *Proceedings of the 3rd International Conference on Learning Representations*, May 7–9 2015.

[123] M. Miled, M. A. B. Messaoud, and A. Bouzid, “Lip Reading of Words with Lip Segmentation and Deep Learning,” *Multimedia Tools and Applications*, vol. 82, no. 1, pp. 551–571, January 2023.

[124] V. Caselles, R. Kimmel, and G. Sapiro, “Geodesic Active Contours,” *International Journal of Computer Vision*, vol. 22, pp. 61–79, February 1997.

[125] C. Li, C. Xu, C. Gui, and M. D. Fox, “Distance Regularized Level Set Evolution and Its Application to Image Segmentation,” *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3243–3254, December 2010.

[126] N. Eveno, A. Caplier, and P.-Y. Coulon, “Accurate and Quasi-Automatic Lip Tracking,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 5, pp. 706–715, 2004.

[127] H. Li, N. Yadikar, Y. Zhu, M. Mamut, and K. Ubul, “Learning the Relative Dynamic Features for Word-Level Lipreading,” *Sensors*, vol. 22, no. 10, May 2022.

[128] C. Feichtenhofer, H. Fan, J. Malik, and K. He, “SlowFast Networks for Video Recognition,” in *Proceedings of the 17th IEEE/CVF International Conference on Computer Vision*. IEEE, October 27–28 2019, pp. 6202–6211.

[129] G. Tan, Y. Wang, H. Han, Y. Cao, F. Wu, and Z.-J. Zha, “Multi-Grained Spatio-Temporal Features Perceived Network for Event-Based Lip-Reading,” in *Proceedings of the 35th IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE. IEEE, June 18–24 2022, pp. 20 094–20 103.

[130] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis *et al.*, “Event-Based Vision: A Survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 154–180, January 2020.

[131] H. Wang, G. Pu, and T. Chen, “A Lip Reading Method Based on 3D Convolutional Vision Transformer,” *IEEE Access*, vol. 10, pp. 77 205–77 212, 2022.

[132] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, “CvT: Introducing Convolutions to Vision Transformers,” in *Proceedings of the 18th*

IEEE/CVF International Conference on Computer Vision. IEEE, October 11–17 2021, pp. 22–31.

[133] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *Proceedings of the 9th International Conference on Learning Representations*, May 3–7 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[134] S. Jeon and M. S. Kim, "Noise-Robust Multimodal Audio-Visual Speech Recognition System for Speech-Based Interaction Applications," *Sensors*, vol. 22, no. 20, p. 7738, October 2022.

[135] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," in *Proceedings of the 15th European Conference on Computer Vision*. Springer, September 8–14 2018, pp. 3–19.

[136] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, June 2014.

[137] S. Lee and C. Lee, "Revisiting Spatial Dropout for Regularizing Convolutional Neural Networks," *Multimedia Tools and Applications*, vol. 79, no. 45, pp. 34 195–34 207, June 2020.

[138] D. Feng, S. Yang, S. Shan, and X. Chen, "Audio-Driven Deformation Flow for Effective Lip Reading," in *Proceedings of the 26th International Conference on Pattern Recognition*, August 21–25 2022, pp. 274–280.

[139] N. F. Aljohani and E. S. Jaha, "Visual Lip-Reading for Quranic Arabic Alphabets and Words Using Deep Learning," *Computer Systems Science and Engineering*, vol. 46, no. 3, pp. 3037–3058, April 2023.

[140] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *NeurIPS 2014 Deep Learning and Representation Learning Workshop*, December 2014.

[141] Z. Su, S. Fang, and J. Rekimoto, "LipLearner: Customizable Silent Speech Interactions on Mobile Devices," in *Proceedings of the 41st ACM CHI Conference on Human Factors in Computing Systems*, ser. CHI '23. Association for Computing Machinery, April 23–28 2023, pp. 1–21.

[142] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable Visual Models from Natural Language Supervision," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, July 18–24 2021, pp. 8748–8763. [Online]. Available: https://proceedings.mlr.press/v139/radford21a

[143] X. Sun, J. Xiong, C. Feng, H. Li, Y. Wu, D. Fang, and X. Chen, "EarSSR: Silent Speech Recognition Via Earphones," *IEEE Transactions on Mobile Computing*, pp. 1–17, January 2024.

[144] K. Cho, B. van Merriënboer, Ç. GuÌ‡çehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation," in *Proceedings of the 19th Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, October 25–29 2014, pp. 1724–1734.

[145] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for MobileNetV3," in *Proceedings of the 17th IEEE/CVF International Conference on Computer Vision*, October 27–November 2 2019, pp. 1314–1324.

[146] S. Jeon, J. Lee, D. Yeo, Y.-J. Lee, and S. Kim, "Multimodal Audiovisual Speech Recognition Architecture Using a Three-Feature Multi-Fusion Method for Noise-Robust Systems," *ETRI Journal*, vol. 46, no. 1, pp. 22–34, February 2024.

[147] Y. Xiang, M. Mamut, N. Yadikar, G. Ibrahim, and K. Ubul, "The Collaboration of 3D Convolutions and CRO-TSM in Lipreading," in *Proceedings of the 49th IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, April 14–19 2024, pp. 4890–4894.

[148] P. Ma, Y. Wang, S. Petridis, J. Shen, and M. Pantic, "Training Strategies for Improved Lip-Reading," in *Proceedings of the 47th IEEE International Conference*

on Acoustics, Speech and Signal Processing. IEEE, May 23–27 2022, pp. 8472–8476.

[149] Z. X. Huang, M. Mamut, G. Ibrahim, N. Yadikar, and K. Ubul, "A Lip-Reading Recognition Method That Integrates a 3D Dual-Stream Convolutional Neural Network," in *Proceedings of the 3rd International Conference on Artificial Intelligence, Human-Computer Interaction and Robotics*. IEEE, November 15–17 2024, pp. 43–48.

[150] G. M. Rahmatullah, S.-J. Ruan, and L. P.-H. Li, "Recognizing Indonesian Words Based on Visual Cues of Lip Movement Using Deep Learning," *Measurement*, vol. 250, p. 116968, March 2025.

[151] A. Gu and T. Dao, "Mamba: Linear-Time Sequence Modeling with Selective State Spaces," 2024. [Online]. Available: https://arxiv.org/abs/2312.00752

[152] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," 2018. [Online]. Available: https://arxiv.org/abs/1708.02002

[153] S.-W. Chung, J. S. Chung, and H.-G. Kang, "Perfect Match: Self-Supervised Embeddings for Cross-Modal Retrieval," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 3, pp. 568–576, March 2020.

[154] C. Kim, H. V. Shin, T.-H. Oh, A. Kaspar, M. Elgharib, and W. Matusik, "On Learning Associations of Faces and Voices," in *Proceedings of the 10th Asian Conference on Computer Vision*. Springer International Publishing, December 2–6 2018, pp. 276–292.

[155] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," 2018. [Online]. Available: https://arxiv.org/abs/1803.01271

[156] I. Loshchilov and F. Hutter, "SGDR: Stochastic Gradient Descent with Warm Restarts," in *Proceedings of the 5th International Conference on Learning Representations*, April 24–26 2017. [Online]. Available: https://openreview.net/forum?id=Skq89Scxx

[157] D. Kastaniotis, D. Tsourounis, and S. Fotopoulos, "Lip Reading Modeling with Temporal Convolutional Networks for Medical Support Applications," in *Pro-*

ceedings of the 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics. IEEE, October 17–19 2020, pp. 366–371.

[158] H. Chen, J. Du, Y. Hu, L.-R. Dai, B.-C. Yin, and C.-H. Lee, "Automatic Lip-Reading with Hierarchical Pyramidal Convolution and Self-Attention for Image Sequences with No Word Boundaries," in *Proceedings of the INTERSPEECH*. ISCA, August 30–September 3 2021, pp. 3001–3005.

[159] I. C. Duta, L. Liu, F. Zhu, and L. Shao, "Pyramidal Convolution: Rethinking Convolutional Neural Networks for Visual Recognition," 2020. [Online]. Available: https://arxiv.org/abs/2006.11538

[160] D. Tsourounis, D. Kastaniotis, and S. Fotopoulos, "Lip Reading by Alternating Between Spatiotemporal and Spatial Convolutions," *Journal of Imaging*, vol. 7, no. 5, p. 91, 2021.

[161] C. Sheng, M. Pietikäinen, Q. Tian, and L. Liu, "Cross-Modal Self-Supervised Learning for Lip Reading: When Contrastive Learning Meets Adversarial Training," in *Proceedings of the 29th ACM International Conference on Multimedia*, ser. MM '21. Association for Computing Machinery, October 20–24 2021, pp. 2456–2464.

[162] P. Ma, Y. Wang, J. Shen, S. Petridis, and M. Pantic, "Lip-Reading with Densely Connected Temporal Convolutional Networks," in *Proceedings of the 21st IEEE/CVF Winter Conference on Applications of Computer Vision*. IEEE, January 5–9 2021, pp. 2857–2866.

[163] C. Sheng, X. Zhu, H. Xu, M. Pietikäinen, and L. Liu, "Adaptive Semantic-Spatio-Temporal Graph Convolutional Network for Lip Reading," *IEEE Transactions on Multimedia*, vol. 24, pp. 3545–3557, August 2021.

[164] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *Proceedings of the 5th International Conference on Learning Representations*, April 24–26 2017. [Online]. Available: https://openreview.net/forum?id=SJU4ayYgl

[165] B. Xu and H. Wu, "Lip Reading Using Multi-Dilation Temporal Convolutional Network," in *Proceedings of the 2nd International Conference on Signal Processing and Machine Learning*, vol. 3150, May 12–18 2022, pp. 50–59.

[166] M. Kim, J. H. Yeo, and Y. M. Ro, “Distinguishing Homophenes Using Multi-Head Visual-Audio Memory for Lip Reading,” in *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, vol. 36, no. 1, February 22–March-1 2022, pp. 1174–1182.

[167] Y. Xiao, L. Teng, A. Zhu, X. Liu, and P. Tian, “Lip Reading in Cantonese,” *IEEE Access*, vol. 10, pp. 95 020–95 029, September 2022.

[168] W. Tian, H. Zhang, C. Peng, and Z.-Q. Zhao, “Lipreading Model Based on Whole-Part Collaborative Learning,” in *Proceedings of the 47th IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, May 23–27 2022, pp. 2425–2429.

[169] Y. Ge, X. Zhang, C. L. Choi, K. C. Cheung, P. Zhao, F. Zhu, X. Wang, R. Zhao, and H. Li, “Self-Distillation with Batch Knowledge Ensembling Improves Imagenet Classification,” 2021. [Online]. Available: https://arxiv.org/abs/2104.13298

[170] G. Song and W. Chai, “Collaborative Learning for Deep Neural Networks,” *Advances in Neural Information Processing Systems*, vol. 31, pp. 1837–1846, December 3–18 2018.

[171] X. Zhang, C. Zhang, J. Sui, C. Sheng, W. Deng, and L. Liu, “Boosting Lip Reading with a Multi-View Fusion Network,” in *Proceedings of the 22nd IEEE International Conference on Multimedia and Expo*. IEEE, July 18–22 2022, pp. 1–6.

[172] X. Wang, L. Bo, and L. Fuxin, “Adaptive Wing Loss for Robust Face Alignment Via Heatmap Regression,” in *Proceedings of the 17th IEEE/CVF International Conference on Computer Vision*. IEEE, October 27–November 2 2019, pp. 6971–6981.

[173] Z.-Q. Zhang, J. Zhang, J.-S. Zhang, M.-H. Wu, X. Fang, and L.-R. Dai, “Learning Contextually Fused Audio-Visual Representations for Audio-Visual Speech Recognition,” in *Proceedings of the 29th IEEE International Conference on Image Processing*. IEEE, October 16–19 2022, pp. 1346–1350.

[174] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised Pre-Training for Speech Recognition," in *Proceedings of the INTERSPEECH*. ISCA, September 15–19 2019, pp. 3465–3469.

[175] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, December 6–12 2020.

[176] H. Yang, T. Luo, Y. Zhang, M. Song, L. Xie, Y. Yan, and E. Yin, "Improved Word-Level Lipreading with Temporal Shrinkage Network and NetVLAD," in *Proceedings of the 24th International Conference on Multimodal Interaction*, ser. ICMI '22. Association for Computing Machinery, November 7–11 2022, pp. 504–508.

[177] M. Zhao, S. Zhong, X. Fu, B. Tang, and M. Pecht, "Deep Residual Shrinkage Networks for Fault Diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4681–4690, September 2019.

[178] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition," in *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 27–30 2016, pp. 5297–5307.

[179] C. Peng, J. Li, J. Chai, Z. Zhao, H. Zhang, and W. Tian, "Lip Reading Using Deformable 3D Convolution and Channel-Temporal Attention," in *Proceedings of the 31st International Conference on Artificial Neural Networks*. Springer Nature Switzerland, September 6–9 2022, pp. 707–718.

[180] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable Convolutional Networks," in *Proceedings of the 16th International Conference on Computer Vision*. IEEE, October 22–29 2017, pp. 764–773.

[181] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: a Simple Data Augmentation Method for Automatic Speech Recognition," in *Proceedings of the INTERSPEECH*. ISCA, September 15–19 2019.

[182] T. Furlanello, Z. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born Again Neural Networks," in *Proceedings of the 35th International Conference on*

Machine Learning. PMLR, Jul 10–15 2018, pp. 1607–1616. [Online]. Available: https://proceedings.mlr.press/v80/furlanello18a.html

[183] J. Feng and R. Long, "Cross-Language Lipreading by Reconstructing Spatio-Temporal Relations in 3D Convolution," *Displays*, vol. 76, p. 102357, December 2022.

[184] Y. Fu, Y. Lu, and R. Ni, "Chinese Lip-Reading Research Based on ShuffleNet and CBAM," *Applied Sciences*, vol. 13, no. 2, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/2/1106

[185] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design," in *Proceedings of the 15th European Conference on Computer Vision*. Springer, September 8–14 2018, pp. 116–131.

[186] J. H. Yeo, M. Kim, and Y. M. Ro, "Multi-Temporal Lip-Audio Memory for Visual Speech Recognition," in *Proceedings of the 48th IEEE International Conference on Acoustics, Speech and Signal Processing*, June 4–10 2023, pp. 1–5.

[187] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal Convolutional Networks for Action Segmentation and Detection," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, July 21–26 2017, pp. 156–165.

[188] T. Arakane, C. Kai, and T. Saitoh, "Can You Read Lips with a Masked Face?" in *Proceedings of the 18th International Conference on Machine Vision and Applications*. IEEE, July 23–25 2023, pp. 1–5.

[189] H. Chen, W. Li, Z. Cheng, X. Liang, and Q. Zhang, "TCS-LipNet: Temporal & Channel & Spatial Attention-Based Lip Reading Network," in *Proceedings of the 32nd International Conference on Artificial Neural Networks*. Springer, September 26–29 2023, pp. 413–424.

[190] X. Zhang, C. Zhang, T. Wang, J. Tang, S. Lao, and H. Li, "Slow-Fast Time Parameter Aggregation Network for Class-Incremental Lip Reading," in *Proceedings of the 31st ACM International Conference on Multimedia*, ser. MM '23. Association for Computing Machinery, October 29–November 3 2023, pp. 747–756.

[191] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, “iCaRL: Incremental Classifier and Representation Learning,” in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition*, IEEE. IEEE, July 21–26 2017, pp. 5533–5542.

[192] J. Huang, L. Teng, Y. Xiao, A. Zhu, and X. Liu, “Lip Reading Using Temporal Adaptive Module,” in *Proceedings of the 30th International Conference on Neural Information Processing*. Springer Nature Singapore, November 20–23 2023, pp. 347–356.

[193] Z. Liu, L. Wang, W. Wu, C. Qian, and T. Lu, “TAM: Temporal Adaptive Module for Video Recognition,” in *Proceedings of the 18th IEEE/CVF International Conference on Computer Vision*. IEEE, October 10–17 2021, pp. 13 688–13 698.

[194] X. Li, Z. Tan, Z. Cheng, and X. Wu, “Lipreading Using Joint Preception Temporal Convolutional Network,” in *Proceedings of the 19th International Conference on Mobility, Sensing and Networking*. IEEE, December 14–16 2023, pp. 612–619.

[195] Y. He, L. Yang, H. Wang, Y. Zhu, and S. Wang, “Speaker-Adaptive Lipreading Via Spatio-Temporal Information Learning,” in *Proceedings of the 49th IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, April 14–19 2024, pp. 10 411–10 415.

[196] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, “Parameter-Efficient Transfer Learning for NLP,” in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, June 10–15 2019, pp. 2790–2799. [Online]. Available: https://proceedings.mlr.press/v97/houlsby19a.html

[197] H. Chen, Q. Wang, J. Du, G.-S. Wan, S.-F. Xiong, B.-C. Yin, J. Pan, and C.-H. Lee, “Collaborative Viseme Subword and End-to-End Modeling for Word-Level Lip Reading,” *IEEE Transactions on Multimedia*, pp. 1–13, April 2024.

[198] W. F. Twaddell, “On Defining the Phoneme,” *Language*, vol. 11, no. 1, pp. 5–62, 1935. [Online]. Available: http://www.jstor.org/stable/522070

[199] C. G. Fisher, “Confusions Among Visually Perceived Consonants,” *Journal of Speech and Hearing Research*, vol. 11, no. 4, pp. 796–804, 1968.

[200] A. A. Montgomery and P. L. Jackson, "Physical Characteristics of the Lips Underlying Vowel Lipreading Performance," *The Journal of the Acoustical Society of America*, vol. 73, no. 6, pp. 2134–2144, 06 1983.

[201] C. Neti, G. Potamianos, J. Luettin, I. Matthews, H. Glotin, D. Vergyri, J. Sison, and A. Mashari, "Audio-Visual Speech Recognition," pp. 1–86, October 2000.

[202] J. Jiang, Z. Zhao, Y. Yang, and W. Tian, "GSLip: A Global Lip-Reading Framework with Solid Dilated Convolutions," in *Proceedings of the 42nd International Joint Conference on Neural Networks*. IEEE, June 30–July 5 2024, pp. 1–8.

[203] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "Global Context Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 6881–6895, 2020.

[204] Y. J. Ahn, J. Park, S. Park, J. Choi, and K.-E. Kim, "SyncVSR: Data-Efficient Visual Speech Recognition with End-to-End Crossmodal Audio Token Synchronization," in *Proceedings of the INTERSPEECH*. ISCA, September 1–5 2024, pp. 867–871.

[205] Z. Gu and J. Jiang, "RAL:redundancy-Aware Lipreading Model Based on Differential Learning with Symmetric Views," 2024. [Online]. Available: https://arxiv.org/abs/2409.05307

[206] S. Daou, A. Ben-Hamadou, A. Rekik, and A. Kallel, "Cross-Attention Fusion of Visual and Geometric Features for Large-Vocabulary Arabic Lipreading," *Technologies*, vol. 13, no. 1, 2025. [Online]. Available: https://www.mdpi.com/2227-7080/13/1/26

[207] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," in *Proceedings of the 6th International Conference on Learning Representations*, April 30–May 3 2018. [Online]. Available: https://openreview.net/forum?id=rJXMpikCZ

[208] B. Hao, D. Zhou, X. Li, X. Zhang, L. Xie, J. Wu, and E. Yin, "LipGen: Viseme-Guided Lip Video Generation for Enhancing Visual Speech Recognition," 2025. [Online]. Available: https://arxiv.org/abs/2501.04204

[209] X. Wu, Z. Tan, Z. Cheng, and Y. Ru, “STDNet: Improved Lip Reading Via Short-Term Temporal Dependency Modeling,” *Virtual Reality & Intelligent Hardware*, vol. 7, no. 2, pp. 173–187, 2025.

[210] X. Zhang, Y. Hu, X. Liu, Y. Gu, T. Li, J. Yin, and T. Liu, “A Novel Approach for Visual Speech Recognition Using the Partition-Time Masking and Swin Transformer 3D Convolutional Model,” *Sensors*, vol. 25, no. 8, 2025.

[211] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows,” in *Proceedings of the 18th IEEE/CVF International Conference on Computer Vision*. IEEE, October 11–17 2021, pp. 10 012–10 022.

[212] F. Bai, W. Li, M. Lu, and Q. Zhang, “LipMSTA: Multi-Scale Spatio-Temporal Attention for Lip-Reading,” in *Proceedings of the 21st International Conference on Intelligent Computing*. Springer, July 26–29 2025, pp. 378–389.

[213] B. H. Lee, W. Shin, and S. W. Han, “TD3Net: A Temporal Densely Connected Multi-Dilated Convolutional Network for Lipreading,” *Journal of Visual Communication and Image Representation*, vol. 111, p. 104540, September 2025.

[214] N. Takahashi and Y. Mitsufuji, “Densely Connected Multidilated Convolutional Networks for Dense Prediction Tasks,” in *Proceedings of the 34th IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 20–25 2021, pp. 993–1002.

[215] K. Paleček, “Experimenting with Lipreading for Large Vocabulary Continuous Speech Recognition,” *Journal on Multimodal User Interfaces*, vol. 12, no. 4, pp. 309–318, July 2018.

[216] G. Zhao, M. Barnard, and M. Pietikainen, “Lipreading with Local Spatiotemporal Descriptors,” *IEEE Transactions on Multimedia*, vol. 11, no. 7, pp. 1254–1265, August 2009.

[217] K. Paleček, “Lipreading Using Spatiotemporal Histogram of Oriented Gradients,” in *Proceedings of the 24th European Signal Processing Conference*. IEEE, August 29–September 2 2016, pp. 1882–1885.

[218] S. Nadeem Hashmi, H. Gupta, D. Mittal, K. Kumar, A. Nanda, and S. Gupta, "A Lip Reading Model Using CNN with Batch Normalization," in *Proceedings of the 11th International Conference on Contemporary Computing*. IEEE, August 2–4 2018, pp. 1–6.

[219] X. Zhang, F. Cheng, and S. Wang, "Spatio-Temporal Fusion Based Convolutional Sequence Learning for Lip Reading," in *Proceedings of the 17th IEEE/CVF International Conference on Computer Vision*. IEEE, October 27–November 2 2019, pp. 713–722.

[220] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," *stat*, vol. 1050, p. 21, 2016.

[221] P. Wiriyathammabhum, "Spotfast Networks with Memory Augmented Lateral Transformers for Lipreading," in *Proceedings of the 17th International Conference on Neural Information Processing*. Springer, 2020, pp. 554–561.

[222] G. Lample, A. Sablayrolles, M. Ranzato, L. Denoyer, and H. Jégou, "Large Memory Layers with Product Keys," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[223] S.-W. Chung, H. G. Kang, and J. S. Chung, "Seeing Voices and Hearing Voices: Learning Discriminative Embeddings Using Cross-Modal Self-Supervision," in *Proceedings of the INTERSPEECH*. ISCA, October 25–29 2020.

[224] M. Luo, S. Yang, X. Chen, Z. Liu, and S. Shan, "Synchronous Bidirectional Learning for Multilingual Lip Reading," in *Proceedings of the The 31st British Machine Vision Conference*, September 7–10 2020, pp. 1–13.

[225] H. Huang, C. Song, J. Ting, T. Tian, C. Hong, Z. Di, and D. Gao, "A Novel Machine Lip Reading Model," *Procedia Computer Science*, vol. 199, pp. 1432–1437, February 2022, the 8th International Conference on Information Technology and Quantitative Management (ITQM 2020 & 2021): Developing Global Digital Economy after COVID-19.

[226] A.-C. Jitaru, L.-D. Ştefan, and B. Ionescu, "Toward Language-Independent Lip Reading: A Transfer Learning Approach," in *Proceedings of the 10th International Symposium on Signals, Circuits and Systems*. IEEE, July 15–16 2021, pp. 1–4.

[227] A. C. Jitaru, Ş. Abdulamit, and B. Ionescu, "LRRo: A Lip Reading Data Set for the Under-Resourced Romanian Language," in *Proceedings of the 11th ACM Multimedia Systems Conference*, ser. MMSys '20. Association for Computing Machinery, 2020, pp. 267–272.

[228] S. Ren, Y. Du, J. Lv, G. Han, and S. He, "Learning from the Master: Distilling Cross-Modal Advanced Knowledge for Lip Reading," in *Proceedings of the 34th IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 19–25 2021, pp. 13 325–13 333.

[229] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum Learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*. Association for Computing Machinery, June 14–18 2009, pp. 41–48.

[230] Y. Takashima, R. Takashima, R. Tsunoda, R. Aihara, T. Takiguchi, Y. Ariki, and N. Motoyama, "Unsupervised Domain Adaptation for Lip Reading Based on Cross-Modal Knowledge Distillation," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2021, no. 44, pp. 1–9, December 2021.

[231] Y. Ganin and V. Lempitsky, "Unsupervised Domain Adaptation by Backpropagation," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 37. PMLR, July 7–9 2015, pp. 1180–1189. [Online]. Available: https://proceedings.mlr.press/v37/ganin15

[232] M. Kim, J. Hong, S. J. Park, and Y. M. Ro, "CroMM-VSR: Cross-Modal Memory Augmented Visual Speech Recognition," *IEEE Transactions on Multimedia*, vol. 24, pp. 4342–4355, October 2021.

[233] C. Sheng, L. Liu, W. Deng, L. Bai, Z. Liu, S. Lao, G. Kuang, and M. Pietikäinen, "Importance-Aware Information Bottleneck Learning Paradigm for Lip Reading," *IEEE Transactions on Multimedia*, vol. 25, pp. 6563–6574, September 2022.

[234] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep Variational Information Bottleneck," in *Proceedings of the 5th International Conference on Learning Representations*, April 24–26 2017. [Online]. Available: https://openreview.net/forum?id=HyxQzBceg

[235] C. Li, Y. Zhang, and H. Du, “FSMS: An Enhanced Polynomial Sampling Fusion Method for Audio-Visual Speech Recognition,” in *Proceedings of the 6th IEEE Information Technology and Mechatronics Engineering Conference*, vol. 6. IEEE, March 4–6 2022, pp. 1601–1604.

[236] S. Chopra, R. Hadsell, and Y. LeCun, “Learning a Similarity Metric Discriminatively, with Application to Face Verification,” in *Proceedings of the 18th IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, June 20–25 2005, pp. 539–546 vol. 1.

[237] X. Pan, P. Chen, Y. Gong, H. Zhou, X. Wang, and Z. Lin, “Leveraging Unimodal Self-Supervised Learning for Multimodal Audio-Visual Speech Recognition,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, May 22–27 2022, pp. 4491–4503.

[238] X. Chen, H. Fan, R. Girshick, and K. He, “Improved Baselines with Momentum Contrastive Learning,” 2020. [Online]. Available: https://arxiv.org/abs/2003.04297

[239] C.-C. Yang, W.-C. Fan, C.-F. Yang, and Y.-C. F. Wang, “Cross-Modal Mutual Learning for Audio-Visual Speech Recognition and Manipulation,” in *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, vol. 36, no. 3, February 22–March 1 2022, pp. 3036–3044.

[240] S. Ding and R. Gutierrez-Osuna, “Group Latent Embedding for Vector Quantized Variational Autoencoder in Non-Parallel Voice Conversion,” in *Proceedings of the INTERSPEECH*. ISCA, September 15–19 2019, pp. 724–728.

[241] N. P. Akman, T. T. Sivri, A. Berkol, and H. Erdem, “Lip Reading Multiclass Classification by Using Dilated CNN with Turkish Dataset,” in *Proceedings of the 2nd International Conference on Electrical, Computer and Energy Technologies*. IEEE, July 20-22 2022, pp. 1–6.

[242] X. Yu, L. Wang, C. Chen, J. Tie, and S. Guo, “Multimodal Learning of Audio-Visual Speech Recognition with Liquid State Machine,” in *Proceedings of the 29th International Conference on Neural Information Processing*. Springer, November 22–26 2022, pp. 552–563.

[243] W. Maass, T. Natschläger, and H. Markram, "Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations," *Neural Computation*, vol. 14, no. 11, pp. 2531–2560, November 2002.

[244] K. Roy, A. Jaiswal, and P. Panda, "Towards Spike-Based Machine Intelligence with Neuromorphic Computing," *Nature*, vol. 575, no. 7784, pp. 607–617, November 2019.

[245] C. Chen, S. Rosa, Y. Miao, C. X. Lu, W. Wu, A. Markham, and N. Trigoni, "Selective Sensor Fusion for Neural Visual-Inertial Odometry," in *Proceedings of the 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 15–20 2019, pp. 10 542–10 551.

[246] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent Models of Visual Attention," *Advances in Neural Information Processing Systems*, vol. 27, pp. 2204–2212, December 8–13 2014.

[247] M. Kim, H. Kim, and Y. M. Ro, "Speaker-Adaptive Lip Reading with User-Dependent Padding," in *Proceedings of the 17th European Conference on Computer Vision*. Springer, October 23–27 2022, pp. 576–593.

[248] P. Nemani, G. S. Krishna, N. Ramisetty, B. D. S. Sai, and S. Kumar, "Deep Learning Based Holistic Speaker Independent Visual Speech Recognition," *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 6, pp. 1705–1713, November 2022.

[249] S. Ji, W. Xu, M. Yang, and K. Yu, "3D Convolutional Neural Networks for Human Action Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, March 2012.

[250] C. Wang, J. Du, H. Chen, R. Wang, C.-H. H. Yang, J. Zhao, Y. Ren, Q. Li, and C.-H. Lee, "Enhancing Privacy Preservation with Quantum Computing for Word-Level Audio-Visual Speech Recognition," in *Proceedings of the 15th Asia Pacific Signal and Information Processing Association Annual Summit and Conference*. IEEE, October 31–November 3 2023, pp. 635–642.

[251] M. Kim, H.-I. Kim, and Y. M. Ro, "Prompt Tuning of Deep Neural Networks for Speaker-Adaptive Visual Speech Recognition," 2023. [Online]. Available: https://arxiv.org/abs/2302.08102

[252] B. Pouthier, L. Pilati, G. Valenti, C. Bouveyron, and F. Precioso, "Another Point of View on Visual Speech Recognition," in *Proceedings of the INTERSPEECH*. ISCA, August 20–24 2023, pp. 4089–4093.

[253] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-Based Action Recognition with Multi-Stream Adaptive Graph Convolutional Networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 9532–9545, October 2020.

[254] Q. Li, Z. Han, and X.-M. Wu, "Deeper Insights Into Graph Convolutional Networks for Semi-Supervised Learning," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, February 2–7 2018.

[255] X. Ai and B. Fang, "Cross-Modal Language Modeling in Multi-Motion-Informed Context for Lip Reading," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2220–2232, June 2023.

[256] A. Sriram, H. Jun, S. Satheesh, and A. Coates, "Cold Fusion: Training Seq2Seq Models Together with Language Models," in *Proceedings of the INTERSPEECH*. ISCA, September 2–6 2018, pp. 387–391.

[257] E.-C. Mănescu, R.-A. Smădu, A.-M. Avram, D.-C. Cercel, and F. Pop, "End-to-End Lip Reading in Romanian with Cross-Lingual Domain Adaptation and Lateral Inhibition," in *Proceedings of the 22nd IEEE International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, IEEE. IEEE, October 26–29 2023, pp. 287–293.

[258] Q. Liu, M. Ge, and H. Li, "Intelligent Event-Based Lip Reading Word Classification with Spiking Neural Networks Using Spatio-Temporal Attention Features and Triplet Loss," *Information Sciences*, vol. 675, p. 120660, May 2024.

[259] Q. Liu, D. Xing, H. Tang, D. Ma, and G. Pan, "Event-Based Action Recognition Using Motion Information and Spiking Neural Networks," in *Proceedings of the 30th International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, August 19–26 2021, pp. 1743–1749.

[260] X. Dong and J. Shen, "Triplet Loss in Siamese Network for Object Tracking," in *Proceedings of the 15th European Conference on Computer Vision*. Springer, September 8–14 2018, pp. 472–488.

[261] H. Chen, C. Wang, J. Du, C.-H. H. Yang, and J. Qi, "Projection Valued-Based Quantum Machine Learning Adapting to Differential Privacy Algorithm for Word-Level Lipreading," in *Proceedings of the 50th IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, April 6–11 2025, pp. 1–5.

[262] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep Learning with Differential Privacy," in *Proceedings of the 23rd ACM SIGSAC Conference on Computer and Communications Security*, ser. CC '16. Association for Computing Machinery, October 24–28 2016, pp. 308–318.

[263] J. Qi, C.-H. Yang, and P.-Y. Chen, "QTN-VQC: an End-to-End Learning Framework for Quantum Neural Networks," *Physica Scripta*, vol. 99, no. 1, p. 015111, December 2023.

[264] L. Banchi, J. Pereira, and S. Pirandola, "Generalization in Quantum Machine Learning: a Quantum Information Standpoint," *PRX Quantum*, vol. 2, p. 040321, November 2021.

[265] A. Mesbah, A. Berrahou, H. Hammouchi, H. Berbia, H. Qjidaa, and M. Daoudi, "Lip Reading with Hahn Convolutional Neural Networks," *Image and Vision Computing*, vol. 88, pp. 76–83, April 2019.

[266] N. Shrivastava, A. Saxena, Y. Kumar, R. R. Shah, D. Mahata, and A. Stent, "MobiVSR: A Visual Speech Recognition Solution for Mobile Devices," 2019. [Online]. Available: https://arxiv.org/abs/1905.03968

[267] T. Saitoh and M. Kubokawa, "LiP25w: Word-Level Lip Reading Web Application for Smart Device," in *Proceedings of the 15th International Conference on Auditory-Visual Speech Processing*. ISCA, August 10–11 2019, pp. 84–88.

[268] P. Ma, B. Martinez, S. Petridis, and M. Pantic, "Towards Practical Lipreading with Distilled and Efficient Models," in *Proceedings of the 46th IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, June 6–11 2021, pp. 7608–7612.

[269] Y. Voutos, G. Drakopoulos, G. Chrysovitsiotis, Z. Zachou, D. Kikidis, E. Kyrodimos, and T. Exarchos, "Multimodal Lip-Reading for Tracheostomy Patients in the Greek Language," *Computers*, vol. 11, no. 3, p. 34, February 2022.

[270] I. W. W. Wisesa and S.-J. Ruan, "Developing a Lightweight Model for Lip-Reading," in *Proceedings of the 10th International Conference on Consumer Electronics-Taiwan*. IEEE, July 17–19 2023, pp. 627–628.

[271] H. Bulzomi, M. Schweiker, A. Gruel, and J. Martinet, "End-to-End Neuromorphic Lip-Reading," in *Proceedings of the 36th IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, June 18–22 2023, pp. 4101–4108.

[272] W. Maass, "Networks of Spiking Neurons: The Third Generation of Neural Network Models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, December 1997.

[273] W. Fang, Z. Yu, Y. Chen, T. Huang, T. Masquelier, and Y. Tian, "Deep Residual Learning in Spiking Neural Networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 21 056–21 069, 2021.

[274] G. Zhang and Y. Lu, "Research on a Lip Reading Algorithm Based on Efficient-GhostNet," *Electronics*, vol. 12, no. 5, p. 1151, February 2023.

[275] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More Features from Cheap Operations," in *Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 13–19 2020, pp. 1580–1589.

[276] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks," in *Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 13–19 2020, pp. 11 534–11 542.

[277] T. Arakane and T. Saitoh, "Efficient DNN Model for Word Lip-Reading," *Algorithms*, vol. 16(6), no. 269, 2023.

[278] S. Zagoruyko and N. Komodakis, "Wide Residual Networks," in *Proceedings of the 27th British Machine Vision Conference*. British Machine Vision Association, September 19–22 2016, pp. 87.1–87.12.

[279] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of the 36th International Conference on Machine

Learning. PMLR, June 9–15 2019, pp. 6105–6114. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html

[280] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "ViViT: A Video Vision Transformer," in *Proceedings of the 18th IEEE/CVF International Conference on Computer Vision*. IEEE, October 11–17 2021, pp. 6836–6846.

[281] E. D. Cubuk, B. Zoph, J. Shlens, and Q. Le, "RandAugment: Practical Automated Data Augmentation with a Reduced Search Space," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 613–18 624, 2020.

[282] H. Wang, B. Cui, Q. Yuan, G. Pu, X. Liu, and J. Zhu, "Mini-3DCvT: a Lightweight Lip-Reading Method Based on 3D Convolution Visual Transformer," *The Visual Computer*, pp. 1–13, June 2024.

[283] J. Zhang, H. Peng, K. Wu, M. Liu, B. Xiao, J. Fu, and L. Yuan, "MiniViT: Compressing Vision Transformers with Weight Multiplexing," in *Proceedings of the 35th IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 18–24 2022, pp. 12 135–12 144.

[284] Y.-H. Park, R.-H. Park, and H.-M. Park, "SwinLip: An Efficient Visual Speech Encoder for Lip Reading Using Swin Transformer," *Neurocomputing*, vol. 639, p. 130289, 2025.

[285] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-Augmented Transformer for Speech Recognition," in *Proceedings of the INTERSPEECH*. ISCA, October 25–29 2020.

[286] D. Estival, S. Cassidy, F. Cox, and D. Burnham, "AusTalk: an Audio-Visual Corpus of Australian English," in *Proceedings of the 9th International Conference on Language Resources and Evaluation*. European Language Resources Association, May 26–31 2014, pp. 3105–3109.

[287] A. Czyzewski, B. Kostek, P. Bratoszewski, J. Kotus, and M. Szykulski, "An Audio-Visual Corpus for Multimodal Automatic Speech Recognition," *Journal of Intelligent Information Systems*, vol. 49, pp. 167–192, January 2017.

[288] A. Rekik, A. Ben-Hamadou, and W. Mahdi, "A New Visual Speech Recognition Approach for RGB-D Cameras," in *Proceedings of the 11th International Conference on Image Analysis and Recognition*. Springer, October 22–24 2014, pp. 21–28.

[289] A. Rios-Navarro, E. Piñero-Fuentes, S. Canas-Moreno, A. Javed, J. Harkin, and A. Linares-Barranco, "LIPSFUS: a Neuromorphic Dataset for Audio-Visual Sensory Fusion of Lip Reading," in *Proceedings of the 58th International Symposium on Circuits and Systems*. IEEE, May 21–25 2023, pp. 1–5.

[290] A. Jiménez-Fernández, E. Cerezuela-Escudero, L. Miró-Amarante, M. J. Domínguez-Morales, F. de Asís Gómez-Rodríguez, A. Linares-Barranco, and G. Jiménez-Moreno, "A Binaural Neuromorphic Auditory Sensor for FPGA: A Spike Signal Processing Approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 4, pp. 804–818, July 2016.

[291] T. Serrano-Gotarredona and B. Linares-Barranco, "A 128 × 128 1.5Contrast Sensitivity 0.9Dynamic Vision Sensor Using Transimpedance Preamplifiers," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 3, pp. 827–838, January 2013.

[292] F. Tao and C. Busso, "Gating Neural Network for Large Vocabulary Audiovisual Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 7, pp. 1290–1302, March 2018.

[293] J. S. Chung and A. Zisserman, "Out of Time: Automated Lip Sync in the Wild," in *Proceedings of the 13th Asian Conference on Computer Vision*. Springer, November 20–24 2016, pp. 251–263.

[294] G. Schwiebert, C. Weber, L. Qu, H. Siqueira, and S. Wermter, "A Multimodal German Dataset for Automatic Lip Reading Systems and Transfer Learning," in *Proceedings of the 13th Language Resources and Evaluation Conference*. European Language Resources Association, June 20–25 2022, pp. 6829–6836.

[295] A. Pondit, M. E. A. Rukon, A. Das, and M. A. Kabir, "BenAV: a Bengali Audio-Visual Corpus for Visual Speech Recognition," in *Proceedings of the 28th International Conference on Neural Information Processing*. Springer, December 8–12 2021, pp. 526–535.

[296] M. M. Rahman, M. R. Tanjim, S. S. Hasan, S. M. Shaiban, and M. A. Khan, "Lip Reading Bengali Words," in *Proceedings of the 5th International Conference*

on Algorithms, Computing and Artificial Intelligence, ser. ACAI '22. Association for Computing Machinery, December 23–25 2022, pp. 1–6.

[297] A. Berkol, T. Tümer-Sivri, N. Pervan-Akman, M. Çolak, and H. Erdem, "Visual Lip Reading Dataset in Turkish," *Data*, vol. 8, no. 1, p. 15, January 2023.

[298] S. Antar, A. Sagheer, S. Aly, and M. F. Tolba, "Avas: Speech Database for Multimodal Recognition Applications," in *Proceedings of the 13th International Conference on Hybrid Intelligent Systems*. IEEE, December 4–6 2013, pp. 123–128.

[299] L. A. Elrefaei, T. Q. Alhassan, and S. S. Omar, "An Arabic Visual Dataset for Visual Speech Recognition," *Procedia Computer Science*, vol. 163, no. C, pp. 400–409, 2019.

[300] Z. Jabr, S. Etemadi, and N. Mozayani, "Arabic Lip Reading With Limited Data Using Deep Learning," *IEEE Access*, vol. 12, pp. 111 611–111 626, August 2024.

[301] J. Peymanfard, A. Lashini, S. Heydarian, H. Zeinali, and N. Mozayani, "Word-Level Persian Lipreading Dataset," in *Proceedings of the 12th International Conference on Computer and Knowledge Engineering*. IEEE, November 17–18 2022, pp. 225–230.

[302] Z. Gan, H. Zeng, H. Yang, and S. Zhou, "Construction of Word Level Tibetan Lip Reading Dataset," in *Proceedings of the 3rd IEEE International Conference on Information Communication and Signal Processing*. IEEE, September 12–15 2020, pp. 497–501.

[303] Aripin and A. Setiawan, "Indonesian Lip-Reading Detection and Recognition Based on Lip Shape Using Face Mesh and Long-Term Recurrent Convolutional Network," *Applied Computational Intelligence and Soft Computing*, vol. 2024, no. 1, 2024. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1155/2024/6479124

[304] P. Borde, R. Manza, B. Gawali, and P. Yannawar, "vVISWa–A Multilingual Multi-Pose Audio Visual Database for Robust Human Computer Interaction," *International Journal of Computer Applications*, vol. 137, no. 4, pp. 25–31, March 2016.

[305] A. Waibel, T. Hanazawa, G. E. Hinton, K. Shikano, and K. J. Lang, "Phoneme Recognition Using Time-Delay Neural Networks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 3, pp. 328–339, Mar 1989.

[306] A. Zhang, Y. Tay, S. Zhang, A. Chan, A. T. Luu, S. C. Hui, and J. Fu, "Beyond Fully-Connected Layers with Quaternions: Parameterization of Hypercomplex Multiplications with 1/n Parameters," in *Proceedings of the 9th International Conference on Learning Representations*, May 3–7 2021. [Online]. Available: https://openreview.net/forum?id=rcQdycl0zyk

[307] F. Rosenblatt, "The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[308] E. Grassucci, A. Zhang, and D. Comminiello, "PHNNs: Lightweight Neural Networks via Parameterized Hypercomplex Convolutions," 2021. [Online]. Available: https://arxiv.org/abs/2110.04176

[309] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *Proceedings of the 7th International Conference on Learning Representations*, May 6–9 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7

[310] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017. [Online]. Available: https://arxiv.org/abs/1704.04861

[311] Y. Tang, K. Han, J. Guo, C. Xu, C. Xu, and Y. Wang, "GhostNetv2: Enhance Cheap Operation with Long-Range Attention," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9969–9982, 2022.

[312] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A New Backbone That Can Enhance Learning Capability of CNN," in *Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, June 14–19 2020, pp. 1571–1580.

[313] J. Chen, S.-h. Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee, and S.-H. G. Chan, "Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks," in *Proceedings of the 36th IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 17-24 June 2023, pp. 12 021–12 031.

[314] D. Qin, C. Leichner, M. Delakis, M. Fornoni, S. Luo, F. Yang, W. Wang, C. Banbury, C. Ye, B. Akin, V. Aggarwal, T. Zhu, D. Moro, and A. Howard, "MobileNetV4: Universal Models for the Mobile Ecosystem," in *Proceedings of the 18th European Conference of Computer Vision*, 2024, pp. 78–96.

[315] J. Zhang, X. Li, J. Li, L. Liu, Z. Xue, B. Zhang, Z. Jiang, T. Huang, Y. Wang, and C. Wang, "Rethinking Mobile Block for Efficient Attention-Based Models," in *Proceedings of the 19th IEEE/CVF International Conference on Computer Vision*. IEEE, Oct 01–06 2023, pp. 1389–1400.

[316] W. Yu, P. Zhou, S. Yan, and X. Wang, "InceptionNeXt: When Inception Meets ConvNeXt," in *Proceedings of the 37th IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 16–22 2024, pp. 5672–5683.

[317] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proceedings of the 35th IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 18–24 2022, pp. 11 966–11 976.

[318] X. Ma, X. Dai, Y. Bai, Y. Wang, and Y. Fu, "Rewrite the Stars," in *Proceedings of the 37th IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 16–22 2024, pp. 5694–5703.

[319] K. Xu, Y. Li, H. Zhang, R. Lai, and L. Gu, "EtinyNet: Extremely Tiny Network for TinyML," in *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, no. 4, Feb 22–Mar 1 2022, pp. 4628–4636.

[320] M. Tan and Q. Le, "EfficientNetV2: Smaller Models and Faster Training," in *Proceedings of the 38th International Conference on Machine Learning*, vol. 139. PMLR, 18–24 Jul 2021, pp. 10 096–10 106. [Online]. Available: https://proceedings.mlr.press/v139/tan21a.html

[321] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han *et al.*, "Yolov10: Real-Time End-to-End Object Detection," *Advances in Neural Information Processing Systems*, vol. 37, pp. 107 984–108 011, Dec 10–16 2024.

[322] A. Krizhevsky, G. Hinton *et al.*, "Convolutional Deep Belief Networks on CIFAR-10," pp. 1–9, 2010. [Online]. Available: https://www.cs.toronto.edu/~kriz/conv-cifar10-aug2010.pdf

[323] C. Luo, J. Zhan, T. Hao, L. Wang, and W. Gao, "Shift-and-Balance Attention," 2021. [Online]. Available: https://arxiv.org/abs/2103.13080

[324] A. Goyal, A. Bochkovskiy, J. Deng, and V. Koltun, "Non-Deep Networks," *Advances in Neural Information Processing Systems*, pp. 6789–6801, November 28–December 9 2022.

[325] Z. Yang, L. Zhu, Y. Wu, and Y. Yang, "Gated Channel Transformation for Visual Recognition," in *Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 13–19 2020, pp. 11 791–11 800.

[326] Z. Liu, Z. Xu, J. Jin, Z. Shen, and T. Darrell, "Dropout Reduces Underfitting," in *Proceedings of the 40th International Conference on Machine Learning*. PMLR, July 23–29 2023, pp. 22 233–22 248. [Online]. Available: https://proceedings.mlr.press/v202/liu23aq.html

Author's Publications

[A1] I. I. Panagos, A. P. Giotis and C. Nikou, "Tracking Multiple Instances of Retail Consumers from RGB and Thermal Images" (Abstract), Engineering Proceedings, Vol. 21, no. 1: 17, 2022, https://doi.org/10.3390/engproc2022021017.

[C1] I. I. Panagos, A. P. Giotis and C. Nikou, "Multi-object Visual Tracking for Indoor Images of Retail Consumers", in Proceedings of the 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), June 2022, https://doi.org/10.1109/IVMSP54334.2022.9816269.

[C2] A. Katsaliros, I. I. Panagos, G. Sfikas, C. Nikou, "Road Crack Detection Using Quaternion Neural Networks", in Proceedings of the 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), June 2022, https://doi.org/10.1109/IVMSP54334.2022.9816292.

[C3] I. I. Panagos, G. Sfikas and C. Nikou, "Compressing Audio Visual Speech Recognition Models With Parameterized Hypercomplex Layers", in Proceedings of the 12th Hellenic Conference on Artificial Intelligence, September 2022, https://doi.org/10.1145/3549737.3549785.

[C4] I. I. Panagos, G. Sfikas and C. Nikou, "Improving Lightweight Isolated Word VSR Performance", *to be submitted*, 2025.

[J1] I. I. Panagos, A. P. Giotis, S. Sofianopoulos and C. Nikou, "A New Benchmark for Consumer Visual Tracking and Apparent Demographic Estimation from RGB and Thermal Images", Sensors, vol. 23, no. 23: 9510, 2023, https://doi.org/10.3390/s23239510.

[J2] I. I. Panagos, G. Sfikas and C. Nikou, "Visual Speech Recognition Using Compact Hypercomplex Neural Networks", Pattern Recognition Letters, vol. 186, pp. 1 – 7, 2024, https://doi.org/10.1016/j.patrec.2024.09.002.

[J3] I. I. Panagos, G. Sfikas and C. Nikou, "Lightweight Operations for Visual Speech Recognition", Preprint, *under review*, 2025.

[J4] I. I. Panagos, G. Sfikas and C. Nikou, “Designing Practical Models for Isolated Word Visual Speech Recognition”, Preprint, *under review*, 2025.

Short Biography

Iason Ioannis Panagos received his B.Sc. degree in Computer Science and Engineering from the Department of Computer Science and Engineering, University of Ioannina, Greece, in 2019. His graduate thesis involved the estimation of hand pose from images using deep learning techniques. He is a Ph.D. candidate at the same department, researching methods to make visual speech recognition more efficient, enabling more practical applications. Since 2020, he has worked as a research associate for several projects in the computer vision domain that include applications of deep learning to multiple target tracking in indoor environments as well as estimation of the gender and age attributes of individuals using body images. His main interests cover the fields of machine learning, computer vision and image processing with a focus on applied deep learning solutions for practical problems.