Keyword spotting in handwritten document images using supervised and unsupervised representations

A Dissertation

submitted to the designated

by the Assembly

of the Department of Computer Science and Engineering

Examination Committee

by

Angelos Giotis

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

University of Ioannina

School of Engineering

Ioannina November 2021

Advisory Committee:

- **Christophoros Nikou**, Professor, Department of Computer Science and Engineering, University of Ioannina
- **Konstantinos Blekas**, Professor, Department of Computer Science and Engineering, University of Ioannina
- **Vassilis Katsouros**, Researcher A′, Institute of Language and Speech Processing, Athena Research Center

Examining Committee:

- **Christophoros Nikou**, Professor, Department of Computer Science and Engineering, University of Ioannina
- **Konstantinos Blekas**, Professor, Department of Computer Science and Engineering, University of Ioannina
- **Vassilis Katsouros**, Researcher A′, Institute of Language and Speech Processing, Athena Research Center
- **Basilis Gatos**, Researcher A′, Institute of Informatics & Telecommunications, National Centre for Scientific Research "Demokritos"
- **Michalis Vrigkas**, Assis. Professor, Department of Communication and Digital Media, University of Western Macedonia
- **Aristeidis Lykas**, Professor, Department of Computer Science and Engineering, University of Ioannina
- **Lysimachos Pavlos Kontis**, Professor, Department of Computer Science and Engineering, University of Ioannina

Dedication

This thesis is dedicated to my beloved family who supported me throughout this long journey.

Acknowledgements

With the completion of this thesis, a large chapter of my life is coming to a close.

Firstly, I would like to thank my supervisor Professor Christophoros Nikou for his seamless cooperation, his continuous support, his guidance, and most importantly, his caring nature for me not only as his student, but also as an individual. I would also like to deeply thank Dr. Basilis Gatos for the opportunity he gave me to work at his lab. His understanding of my abilities and flaws, significantly affected me to improve my general mindset. Moreover, I would like to thank Dr. Vasilis Katsouros for his excellent cooperation from the beginning of this journey. His crucial interventions made me realize that early prioritization gets actual results. I can not forget the flawless communication I had with Professor Konstantinos Blekas, regrading my academic obligations, as well as his sincere interest for my improvement as a researcher, sharing important ideas and thoughts that I respect and wish to follow.

Furthermore, I would like to thank Professor Aristeidis Lykas for his priceless academic advice that helped me realize my goal. Also, I would like to thank Assistant Professor Michalis Vrigkas and Professor Lysicmachos Kontis for their cooperation as well as the necessary distractions their company offered me during extracurricular activities. I could also not forget to thank Dr. Giorgos Sfikas for our exceptional cooperation during all these years, which I wish it will continue.

Finally, I would like to thank my family who above all, were there for me and gave me the courage to carry on, no matter what the circumstances. Especially, I would like to thank my brother George whom I deeply love and who emotionally lifted much of my burdens which helped me get up every time I fell and keep going. I would like to thank my friends Christos, Thanasis and Dimitris for their priceless extracurricular advice that helped me understand my limits and capabilities. Finally, I would like to thank a very close to me person who helped me believe in myself and do what needs to be done.

Table of Contents

List of Figures

List of Tables

Glossary

ALCM	Adaptive Local Connectivity Map
BLSTM	Bidirectional long-short term memory
BoF-HMM	Bag of features hidden Markov model
BoVW	Bag of Visual Words
CC	Connected component
CDP	Continuous Dynamic Programming
CDBN	Convolutional deep belief network
CNN	Convolutional neural network
CRF	Conditional random field
CSR	Common Subspace Regression
DCToW	Discrete cosine transform of words
DNN	Deep neural networks
DTW	Dynamic time warping
GMM	Gaussian mixture model
GSC	Gradient, structural and concavity
HMM	Hidden Markov model
HoG	Histogram of gradients
HTR	Handwritten Text Recognition
HKS	Heat Kernel Signature
HKWS	Handwritten keyword spotting
IoU	Intersection over union
KWS	Keyword spotting
LBP	Local Binary Patterns
LGH	Local Gradient Histogram
LLC	Locality-constrained Linear Coding
LSI	Latent Semantic Indexing

MAP	Mean Average Precision
MMR	Multimedia retrieval
MRF	Markov random field
NDCG	Normalized Discounted Cumulative Gain
NN	Neural networks
OCR	Optical Character Recognition
OOV	Out of vocabulary
PAS	Pairs of adjacent segments
PAW	Pieces of Arabic Words
PDA	Personal digital assistant
PHOC	Pyramidal Histogram of Characters
QBE	Query-by-example
QBS	Query-by-string
QCNN	Quaternion convolutional neural network
RLSA	Run Length Smoothing Algorithm
RNN	Recurrent neural network
ROI	Region of interest
RPN	Region Proposal Network
SIFT	Scale invariant Fourier transform
SC-HMM	Semi-continuous hidden Markov model
SPM	Spatial Pyramid Matching
SPOC	Spatial Pyramid of Characters
SPP	Spatial Pyramid Pooling
SVM	Support vector machine
WFST	Weighted Finite State Transducers
WSC	Word Shape Coding
WS	Word spotting

Abstract

Angelos Giotis, Ph.D., Department of Computer Science and Engineering, School of Engineering, University of Ioannina, Greece, November 2021.
Keyword spotting in handwritten document images using supervised and unsupervised representations.
Advisor: Christophoros Nikou, Professor.

Vast collections of documents available in image format need to be efficiently digitized for information retrieval purposes. Many approaches from the document analysis and recognition research community have been proposed to alleviate the search process. However, the automatic recognition of degraded manuscripts using traditional Optical Character Recognition (OCR) is impractical, due to inherent challenges of these documents such as unknown layouts and fonts, the variability of handwriting and the open vocabulary. For this reason, the recent attraction for large-scale document indexing is based on a recognition-free image retrieval technique, known as keyword spotting (KWS).

The main focus of this PhD thesis lies on the systematical study and development of handwritten KWS methods as a practical solution, contrary to a costly and error-prone full text transcription. KWS methods aim to retrieve all instances of a user query in a set of document images. In an attempt to denote which parts of a KWS system require most attention to achieve high accuracy, we present a comprehensive survey of KWS techniques. To this end, each fundamental step of the respective pipeline, including layout analysis and preprocessing, feature selection and extraction, representation learning, alignment and matching is thoroughly explored. Several aspects that need to be taken into account such as robustness to writing style variabilities, the availability of training data, the evaluation protocols and measures as well as enhancement techniques which further boost the performance are highlighted

and composed to a structured methodology. By these means, we suggest a theoretical foundation to be adopted by future works for unbiased evaluation and comparison.

Of most importance is the appropriate selection of features to form discriminative word image representations which can yield accurate and fast retrieval. In this thesis, we developed two template-based methods using translation and scale-invariant handcrafted features for KWS on modern and historical manuscripts. In the first work, supervised local contour features are used to train a representative shape of a word-class to address intra-class writing style variations. Its only limitation is related to out of vocabulary queries. The second method comprises an adaptation of the initial system into an unsupervised scheme for efficient and accurate script independent KWS.

Nevertheless, both former approaches are based on variable length image representations which are not fast to compare. Hence, a methodology which adapts a family of supervised, fixed-length representations that encode attribute-like features of the word image transcription is proposed for fast word retrieval. Attributes are properties that reflect the occurrence or absence of textual components (e.g. characters) at specific positions of the word. The proposed method extends this binary word image representation to include language-dependent features present in polytonic Greek text. Following recent trends with respect to the deep learning era, in order to improve the representational power of word images, we propose a deep learning-based framework as an extraction model of deep features which are used to adapt KWS on weakly supervised diverse manuscripts with high distribution shift between source and target datasets. To this end, spatial transformations of the convolutional feature space aim to deter the KWS model so as to adversarially improve its robustness to unknown writing styles and word-classes. Finally, a technique to spot text regions in challenging historical natural images is proposed relying on adversarial learning of quaternion image descriptors which are far less resource demanding than vanilla neural network representations.

Εκτεταμενη Περιληψη

Άγγελος Γιώτης, Δ.Δ., Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πολυτεχνική Σχολή, Πανεπιστήμιο Ιωαννίνων, Νοέμβριος 2021.
Εντοπισμός λέξεων σε εικόνες χειρόγραφων κειμένων με επιβλεπόμενες και μη επιβλεπόμενες αναπαραστάσεις τους.
Επιβλέπων: Χριστόφορος Νίκου, Καθηγητής.

Ένας αρκετά μεγάλος όγκος δεδομένων από συλλογές εγγράφων χρειάζεται να ψηφιοποιηθεί για την δημιουργία ψηφιακών βιβλιοθηκών με στόχο τη διατήρηση του υλικού και την εύχρηστη αναζήτησή του. Οι παραδοσιακές τεχνικές ψηφιακής επεξεργασίας εικόνας που βασίζονται στην πλήρη οπτική αναγνώριση χαρακτήρων των εγγράφων με σκοπό τη δεικτοδότησή τους, δεν παρουσιάζουν ικανοποιητικά αποτελέσματα εξαιτίας εγγενών παραγόντων των εγγράφων. Οι παράγοντες αυτοί σχετίζονται με τις διαφορετικές μορφές δομής σελίδας των εγγράφων, με τις άγνωστες, κατά την εκπαίδευση των μοντέλων αναγνώρισης, γραμματοσειρές των κειμένων, τη διαφορετικότητα ως προς τον τρόπο γραφής και τον άγνωστο, δυνατό αριθμό όρων αναζήτησης που χρειάζεται να έχει ένα λεξικό μοντέλο αναγνώρισης χαρακτήρων. Για τους λόγους αυτούς, το ενδιαφέρον της ερευνητικής κοινότητας στην περιοχή των μεθόδων δεικτοδότησης εγγράφων σε μεγάλη κλίμακα στρέφεται σε εναλλακτικές τεχνικές, απαλλαγμένες από τη διαδικασία αναγνώρισης, γνωστές ως τεχνικές εντοπισμού λέξεων.

Το αντικείμενο της διδακτορικής διατριβής αφορά στον εντοπισμό λέξεων (ΕΛ) σε εικόνες χειρόγραφων κειμένων. Προς αυτή την κατεύθυνση, η διατριβή αυτή περιλαμβάνει τη συστηματική μελέτη και ανάπτυξη μεθόδων ΕΛ, ως μιας πρακτικής προσέγγισης στην ανάκτηση πληροφορίας από χειρόγραφα κείμενα, σε αντίθεση με τις παραδοσιακές τεχνικές πλήρους αναγνώρισης οι οποίες αρκετά συχνά παράγουν εσφαλμένες εκτιμήσεις. Ένα σύστημα εντοπισμού λέξεων αποσκοπεί στην εύρεση όλων των στιγμιότυπων μιας ζητούμενης, από ένα χρήστη, λέξης, μέσα στις

συλλογές κειμένων. Σε μια προσπάθεια να τονίσουμε τα σημεία που χρειάζονται προσοχή κατά την ανάπτυξη τεχνικών ΕΛ που επιτυγχάνουν υψηλή απόδοση, παρουσιάζουμε μια εκτενή μελέτη της βιβλιογραφίας, μέσα από την οποία αναλύεται σε βάθος κάθε πρωταρχική συνιστώσα της αρχιτεκτονικής ενός συστήματος ΕΛ. Οι συνιστώσες αυτές, μεταξύ άλλων, περιλαμβάνουν την ανάλυση δομής σελίδας και την προ-επεξεργασία των εγγράφων, την επιλογή και διαδικασία εξαγωγής χαρακτηριστικών που αναπαριστούν λέξεις, την εκμάθηση κατάλληλων αναπαραστάσεων των λέξεων από περιγραφικά χαρακτηριστικά και την ευθυγράμμιση των αναπαραστάσεων για το τελικό ταίριασμα των εικόνων των λέξεων. Παράγοντες, όπως η ανθεκτικότητα στη διαφοροποίηση του γραφικού χαρακτήρα, η διαθεσιμότητα δεδομένων εκπαίδευσης, οι δείκτες και τα πρωτόκολλα αξιολόγησης των μεθόδων καθώς και μια σειρά από τεχνικές μετα-βελτίωσης του τελικού αποτελέσματος του ΕΛ, αναδεικνύονται μέσα από μια δομημένη μεθοδολογία σχεδιασμού τεχνικών ΕΛ. Με αυτόν τον τρόπο, προτείνουμε ένα θεωρητικό υπόβαθρο, κατάλληλο να υιοθετηθεί από τις μελλοντικές εργασίες, επιτρέποντας την αμερόληπτη αξιολόγηση και σύγκρισή τους.

Ιδιαίτερης σημασίας στη δημιουργία διακριτικών αναπαραστάσεων, ικανών να επιτύχουν υψηλή απόδοση και ταχύτητα ταιριάσματος εικόνων, είναι η επιλογή των κατάλληλων χαρακτηριστικών που περιγράφουν τις εικόνες των κειμένων. Κατά τα πρώιμα στάδια εκπόνησης της διδακτορικής διατριβής, αναπτύχθηκαν δυο τεχνικές που βασίζονται στην αναζήτηση λέξεων με παράδειγμα (δηλαδή, επιλέγοντας μια λέξη που εκκινεί τη διαδικασία ΕΛ) χρησιμοποιώντας τοπικά χαρακτηριστικά περιγράμματος των εικόνων, αμετάβλητα σε μετασχηματισμούς μετατόπισης και κλιμάκωσης. Η πρώτη εργασία, αξιοποιώντας δεδομένα μάθησης για κάθε κατηγορία λέξης, προτείνει ένα μοντέλο ΕΛ αντιπροσωπευτικό της μέσης διαφοροποίησης του σχήματος των λέξεων της κατηγορίας, αντιμετωπίζοντας έτσι τις πιθανές αλλαγές στον τρόπο γραφής κάθε λέξης. Ο μόνος περιορισμός της μεθόδου αφορά στη δυνατότητα αναζήτησης των λέξεων εκείνων για τις οποίες υπάρχουν στιγμιότυπα εικόνων στο σύνολο εκπαίδευσης. Η δεύτερη μεθοδολογία αξιοποιεί τα ίδια διακριτικά χαρακτηριστικά αναπαράστασης λέξεων, απαλλαγμένης όμως από δεδομένα μάθησης, για τον αποδοτικό ΕΛ σε εικόνες ετερογενών, ως προς το αλφάβητο και τη γλώσσα, χειρόγραφων κειμένων.

Βασικό μειονέκτημα των χαρακτηριστικών που προτάθηκαν για τον ΕΛ στις προηγούμενες μεθοδολογίες, είναι οι μεταβλητού μήκους αναπαραστάσεις (διανύ-

σματα) των λέξεων, για τις οποίες οι προτεινόμενοι αλγόριθμοι ταιριάσματος δεν οδηγούν πάντοτε σε ικανοποιητική απόδοση. Κατά συνέπεια, στην πορεία της διατριβής, προτάθηκε μια μέθοδος που βασίζεται σε αναπαραστάσεις λέξεων σταθερού μήκους, οι οποίες μπορούν άμεσα να συγκριθούν με μια αναζήτηση κοντινότερου γείτονα (π.χ. Ευκλείδια απόσταση) οδηγώντας έτσι σε πολύ ταχύτερη ανάκτηση. Επιπρόσθετα, τα χαρακτηριστικά αυτά, αξιοποιώντας δεδομένα μάθησης, έχουν τη δυνατότητα να ενσωματώσουν αρκετά μεγάλο ποσοστό της συνολικής διαφοροποίησης ως προς το γραφικό χαρακτήρα, εφόσον κωδικοποιούν ιδιότητες πρωτογενών τμημάτων των λέξεων (π.χ. χαρακτήρων) που επαναλαμβάνονται τακτικά σε συγκεκριμένες θέσεις μέσα στις λέξεις, ανεξάρτητα από τον τρόπο γραφής. Οι ιδιότητες αυτές σχετίζονται με την παρουσία ή όχι ενός χαρακτήρα σε μια δεδομένη θέση της λέξης. Η προτεινόμενη τεχνική επεκτείνει το μοντέλο δυαδικής αναπαράστασης λέξης ώστε να συμπεριλάβει χαρακτηριστικά που σχετίζονται με τις ιδιομορφίες του πολυτονικού συστήματος γραφής για ΕΛ σε Ελληνικά πολυτονικά κείμενα.

Ακολουθώντας την τρέχουσα τάση της ερευνητικής κοινότητας που συνοδεύεται από τη ραγδαία αύξηση των μεθόδων ΕΛ οι οποίες βασίζονται σε βαθιά μάθηση από την πληθώρα δεδομένων εκπαίδευσης που είναι πλέον διαθέσιμα, προτείνουμε μια ακόμη μέθοδο, ώστε να βελτιστοποιήσουμε την αναπαραστατική ισχύ των διανυσμάτων λέξεων. Στην προτεινόμενη τεχνική, χρησιμοποιούμε συνελικτικά νευρωνικά δίκτυα για την εξαγωγή βαθιών χαρακτηριστικών. Τα χαρακτηριστικά αυτά επιτρέπουν την προσαρμογή του προτεινόμενου μοντέλου ΕΛ, όταν αυτό εκπαιδεύεται σε χαμηλής στάθμης, ως προς τις διαφοροποιήσεις γραφικού χαρακτήρα και την ποσότητα, δεδομένα μάθησης, σε συλλογής κειμένων των οποίων η κατανομή διαφοροποιήσεων διαφέρει αισθητά σε σχέση με το αρχικό σύνολο εκπαίδευσης. Επιπλέον, θεωρούμε ότι η υπό εξέταση συλλογή κειμένων περιέχει ελάχιστα δεδομένα εκπαίδευσης για την προσαρμογή του μοντέλου ΕΛ, το οποίο καθιστά το πρόβλημα ακόμη πιο δύσκολο. Για την αντιμετώπιση των προκλήσεων αυτών προτείνουμε ένα ανταγωνιστικό πλαίσιο βαθιάς μάθησης, όπου το βασικό μοντέλο ΕΛ ανταγωνίζεται ένα δεύτερο νευρωνικό δίκτυο που στοχεύει στην αλλοίωση των εικόνων με μια σειρά από γεωμετρικούς μετασχηματισμούς στον υπόχωρο των βαθιών χαρακτηριστικών. Η αλλοίωση αυτή λειτουργεί σαν εμπόδιο στην διαδικασία εκπαίδευσης για την εξαγωγή διακριτικών αναπαραστάσεων από το μοντέλο ΕΛ, βελτιώνοντας έτσι επαναληπτικά, την ανθεκτικότητα της μεθόδου στους διαφορετικούς τρόπους γραφής και τον εντοπισμό άγνωστων (κατά την εκμάθηση) λέξεων της υπό εξέταση

συλλογής κειμένων.

Τέλος, στο πλαίσιο αξιοποίησης μεθόδων βαθιάς μάθησης, με χρήση παραγωγικών ανταγωνιστικών νευρωνικών δικτύων, προτείνουμε μια τεχνική εντοπισμού περιοχών κειμένου σε φυσικές εικόνες ιστορικών Βυζαντινών επιγραφών. Βασική καινοτομία της μεθόδου είναι η χρήση τετραδονιακών (επέκταση μιγαδικών) αναπαραστάσεων που κωδικοποιούν αποδοτικά την πληροφορία όλων των χρωματικών συνιστωσών των εικόνων, απαιτώντας πολύ λιγότερους υπολογιστικούς πόρους από ισοδύναμες βαθιές αναπαραστάσεις πραγματικών τιμών των εικόνων.

Chapter 1

Introduction

1.1 Problem at hand

A great amount of information in libraries and cultural institutions exist all over the world and need to be digitized so as to preserve it and protect it from frequent handling. During the 1960s, Handwritten Text Recognition (HTR) emerged [6], aiming to make such information more accessible to the public. Over the last decades, many methodologies from the field of document analysis and recognition have been developed to alleviate the search process for digitized manuscripts available online. Early approaches to HTR relied on Optical Character Recognition (OCR) [7], which was a very popular area of research during the 90s. Nevertheless, OCR is most effective when a character segmentation step is employed beforehand, which is typically feasible for machine-printed text.

In order to create digital libraries which allow efficient searching and browsing for future users, thousands of digitized documents have to be transcribed or at least indexed at a certain degree. However, automatic full text transcription is not always a practical solution, especially for the case of historical documents. Moreover, the automatic recognition of poor quality printed text and even more, handwritten text,

is not possible by traditional OCR approaches which mainly suffice for modern printed documents with simple layouts and known fonts. Most of the constraints encountered by recognition systems stem from difficulties in segmenting characters or words, the variability of the handwriting and the open vocabulary. For this reason, more flexible information retrieval and image analysis techniques are required.

1.2 Document indexing using image retrieval methods

The actual problem behind building digital libraries lies on the retrieval of digitized documents in terms of reliable extraction and access to specific information. While a document image processing system analyzes different text regions so as to convert them to machine-readable text using OCR, a document image retrieval system searches whether a document image contains particular words of interest, without the need for correct character recognition, but by directly characterizing image features at character, word, line or even document level.

1.2.1 Text recognition

Recognition-based retrieval relies on the complete recognition of documents either at character level using OCR, or at word level using *word recognition* methods. In the latter case, the goal is to correctly classify a query word into a labeled class, or else, obtain its transcription. Most methods of this type require prior transcription of text-lines, words or characters to train character or word models. During the search phase, a text dictionary or lexicon is used and only words from that lexicon can be used as candidate transcriptions in the recognition task. These methods usually exploit the sequential nature of text, relying on hidden Markov models (HMMs) [8,9], conditional random fields (CRFs) [10] or neural networks (NNs) [11–13] to encode character sequences which are then aligned by a decoding process between query and target keywords. In the final step of recognition, they might follow a hybrid approach by combining different classifiers, such as support vector machines (SVMs) with HMMs [14, 15] or HMMs with NNs [16]. An obvious drawback of these approaches is that they have to deal with the inherent handwriting variability and handle a large number of word and character models. Nevertheless, the scope of this work does not focus on recognition-based retrieval methods and thus, we only briefly refer to them.

1.2.2 Keyword spotting (KWS)

The *recognition-free* retrieval paradigm which is also known in the literature as *word spotting* (WS) or *keyword spotting* (KWS) is the main subject of this thesis. The goal here is to retrieve all instances of user queries in a set of document images which may be segmented at text lines or words. Actually, the user formulates a query and the system evaluates its similarity with the stored documents and returns as output a ranked list of results which are most similar to the query. The process is totally based on matching between common representations of features, such as color, texture, geometric shape or textual features, while conversion of whole documents into machine readable format and recognition do not take place at all. Therefore, the selection and use of proper features and robust matching techniques are the most important aspects of a word spotting system.

Word spotting methods may be divided into multiple categories according to various factors. Depending on how the input is specified by the user we can distinguish *query-by-example* (QBE) from *query-by-string* (QBS) methods. In the QBE scenario, the user selects an image of the word to be searched in the document collection, whereas in the QBS paradigm, the user provides an arbitrary text string as input to the system. Another way to categorize word spotting methods depends on whether training data, namely, annotations at character, word or text-line level, are used offline, either to learn character and word models or tune the parameters of the KWS system. This way we can distinguish *learning-based* from *learning-free* approaches. Finally, word spotting methods which can be directly applied to whole document pages are considered as *segmentation-free*, in contrast with *segmentation-based* methods, where a segmentation step has to be applied at line or word level during image preprocessing.

Word spotting was initially proposed in the speech recognition community [17]. Its application was adopted later on for printed [18,19] and handwritten [20] document indexing. While early approaches were based on raw features extracted directly from image pixels [20, 21], the standard trend is to characterize document images with more complex features based on gradient information, shape structure, texture, etc. (see Section 2.4.1).

1.2.3 Applications of KWS

There are a variety of applications of word spotting for document indexing and retrieval including the following:

- retrieval of documents with a given word in company files,
- searching online in cultural heritage collections stored in libraries all over the world,
- automatic sorting of handwritten mail containing significant words (e.g. "urgent", "cancelation", "complain") [22],
- identification of figures and their corresponding captions [23],
- keyword retrieval in pre-hospital care reports (PCR forms) [24],
- word spotting in graphical documents such as maps [25],
- retrieval of cuneiform structures from ancient clay tablets [26, 27],
- assisting human transcribers in identifying words in degraded documents, especially those appearing for the first time [28].

Although word spotting and word recognition belong to two separate retrieval paradigms, they sometimes interact by assisting one another. For instance, the authors in [29] propose a keyword spotting approach relying on a NN-based recognition system. On the contrary, in [30], word spotting contributes as a means of bootstrapping a handwriting recognition system, in terms of selecting new elements from the retrieved results. These elements can be used to augment the training set through a semi-supervised procedure, thus increasing the final recognition accuracy while at the same time avoiding the costly manual annotation process. Actually, there has been a growing interest over the last years in designing KWS systems that jointly solve keyword recognition and spotting as a classification as well as a retrieval task, respectively [31–37]. Typically, a lexicon of the languages examined is used for recognizing the query image, whereas the proposed methods are able to perform both QBE and QBS for KWS, usually by adopting attribute-like features [31]. Such features encode information from textual and visual elements into compact holistic representations. These methods perform well assuming a large subset of the document collection is transcribed at word level.

Finally, apart from this type of application, word spotting and recognition are also used in more generic computer vision tasks such as recognizing text in natural scene images. Unlike document analysis, different difficulties such as huge variations in illumination, point of view, typography, text slant or skew and others, are encountered. In fact, text detection in natural images is a challenging task due to the variety of text appearance, the unconstrained locations of text within the natural image, degradations of text components present in historical material, as well as the complexity of each scene. To address these challenges, the recent interest has shifted towards machine learning models, i.e. convolutional neural networks (CNNs) [38, 39]. Although text spotting is not the main attraction of this thesis, a specific application for detecting text components in historical natural images is discussed in Chapter 5.

1.2.4 Evolution of the related works

In order to track the recent literature, we present some statistics related to the evolution of word spotting methods over the last 15 years. The research community concentrates on indexing historical documents on a large scale using word spotting, hoping that key elements which capture most of the underlying manuscript information would be an alternative solution to a costly full text transcription. Thus, we consider that the recognition-free retrieval task remains an open problem. To the best of our knowledge, Figure 1.1 provides a concise view of the various word spotting approaches for offline, handwritten or printed documents, which were published in conferences and journals or archive repositories since 2007. As it can be seen in Figure 1.1, there is an increased number of papers over the last decade which confirms the growing interest of the community in word spotting. It is also interesting to note that during $2015 - 2016$, a significant boost of approaches can be observed, mainly due to the rapid growth of the deep learning research community at that time, as we also discuss in Chapter 2.

1.3 Contributions and structure of the thesis

The contribution of this thesis can be summarized as follows. Initially, a survey of KWS techniques is presented. In this work, each key step of a generic KWS system (document image layout analysis and preprocessing, feature selection and extraction,

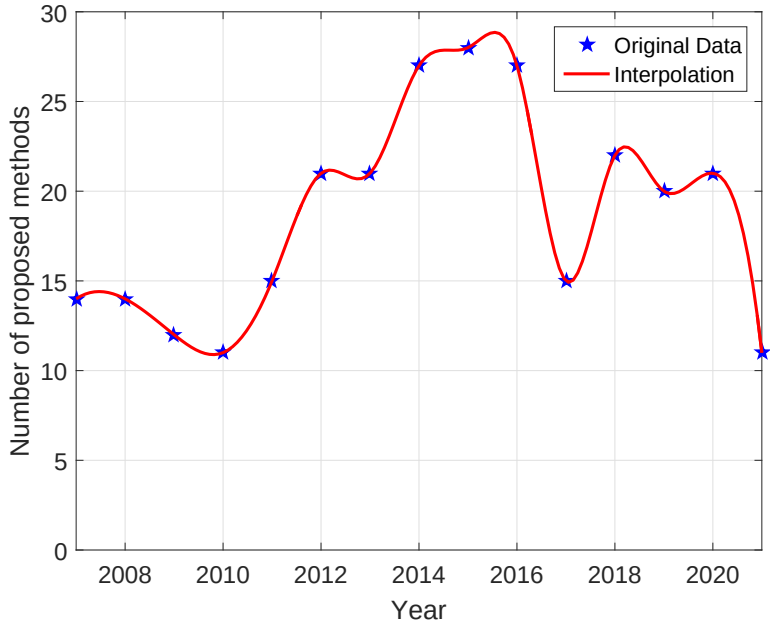


Figure 1.1: Word spotting approaches published over the last 15 years.

representation alignment and matching) is thoroughly examined, whereas several aspects that need to be taken into consideration when developing a KWS technique (writing style variability, availability of image annotations for training, data preparation, evaluation protocols and measures, enhancement of retrieval results) are underlined and composed to a structured methodology that is proposed to be followed as a standard approach by future works for unbiased evaluation and comparison. Subsequently, two main approaches based on translation and scale-invariant handcrafted features, which were developed for handwritten keyword spotting (HKWS) on modern and historical documents at an early stage of the thesis are presented. In addition, a methodology which adapts a family of learning-based representations which encode attribute-like features of the word image transcription for KWS in polytonic Greek text is proposed. Progressing accordingly to the deep learning era, our main effort is shifted towards attribute-based and deep features for the adaptation of a KWS system on diverse manuscripts where data distributions differ substantially between source and target datasets. Finally, in the same spirit with deep features, a technique to spot text regions in the wild was recently developed based on adversarial learning of quaternion image descriptors.

The structure of the thesis is briefly presented for each chapter, along with its key contributions, as follows:

- In **Chapter 2**, we present an extensive review of document image word spotting

techniques [40]. We analyze the nature of text sources along with the inherent difficulties addressed by word spotting methods. Among the main steps of the word spotting system, namely, feature extraction, representation and similarity computation, we also investigate the preprocessing stage with respect to binarization, segmentation and normalization techniques. Furthermore, we present the benefits accrued from relevance feedback methods employed in the retrieval phase of a word spotting task, either by involving the user to select true query instances or in a completely unsupervised way. Afterwards, we examine whether direct comparison among different methods is straightforward or not, since the evaluation measures and protocols applied for assessing the performance may differ substantially. Finally, we present the most commonly used datasets along with the experimental results published by the state-of-the-art methods and discuss about the performance obtained in each case with a view to what should be the next step in the development of KWS systems.

- **Chapter 3** is composed of two parts. A learning-based method [41] developed for word spotting in modern Greek handwritten text written by multiple authors. To this end, we make use of translation and scale-invariant local contour features, previously employed for object detection. The query is performed under a *query-by-word-class scenario*, which is a variant of the QBE paradigm. The proposed trained model is able to deform to unknown writing style variations for which no training data was used. The second part describes the transformation of the initial system into a learning-free scheme for script independent word spotting in historical handwritten text, written by a few authors [42]. An improvement of this learning-free method is also developed with respect to the feature similarity measure yielding faster and more accurate performance.

- **Chapter 4** includes an attribute-based framework for word spotting in typewritten documents. This method suggests a modification of previously proposed attribute representations (PHOC) [31], successfully applied for multi-writer KWS. The proposed descriptor [43] extends the binary encoding of a word image (which reflects the occurrence or absence of an attribute at a specific position in the word), so as to include language-dependent features present in polytonic Greek documents. The next part focuses on the use of spatial transformations of deep features in the convolutional feature space, as a means to augment weakly

supervised document collections. Transformed deep features are utilized in an adversarial learning scheme, aiming to hamper the ability of the employed convolutional neural network to accurately predict attribute-based labels for KWS. In this framework, following a transfer learning approach, deep features are able to adapt to writing style variabilities and unknown word-classes, not seen during training, for efficient KWS in target manuscripts with limited annotations.

- **Chapter 5** includes an application of adversarial features represented by quaternion descriptors for text spotting in the wild [44]. In this respect, KWS is not explicitly addressed as the main problem. Instead, deep quaternionic representations are proposed to solve a relaxed problem of text detection in natural images obtained from scanned Byzantine inscriptions.

- In **Chapter 6** conclusions are drown with respect to the benefits obtained from each developed technique and future directions of our research are discussed.

Chapter 2

Families of approaches with respect to KWS pipeline

In this Chapter, we present a comprehensive study on keyword spotting methods developed over the last 15 years, along with the key aspects which affect the performance of a KWS system. The initial work, which was published during the third year of the thesis, includes an extensive review of KWS approaches that were proposed from 2007 to 2016 in an attempt to alleviate indexing a wide variety of documents written in various scripts or fonts. Therein, we examined the text nature of the documents used by the literature, we described the intermediate steps of a word spotting system, namely, preprocessing, feature extraction, representation and similarity measures which are used to retrieve instances of user inserted queries. Subsequently, we overviewed a number of boosting techniques which enhance the outcome of the image matching

step. Evaluation standards applied for the performance assessment of a word spotting system were also investigated along with the need for a commonly established protocol to allow straightforward comparison with the state of the art. Finally, we presented the results reported by the state of the art (at that time) in the most commonly used databases. In that sense, we aimed to provide a solid background for new researchers of the document analysis and text understanding community, while highlighting KWS key mechanisms that make it work as an efficient information mining and retrieval system for document image collections.

As a final contribution of this thesis to the former study, we further updated each basic component of the initial work, regarding the KWS pipeline and several factors that affect its performance (image preprocessing, feature extraction, representation, etc.) with key methods which were proposed from 2016 to date. We also update the results reported by the current state of the art. Most of the main effort focuses on the recent bloom of adopting deep learning techniques to increase the performance on one or more distinct components of a KWS system.

2.1 Preliminary studies on KWS

Apart from the KWS methods proposed over the years, there also exist a number of surveys for word spotting, either for a specific script, or a particular domain (machine-printed, handwritten), or even for a variety of applications. Murugappan et al. [45] present a study for word spotting in printed documents. The authors divide the word spotting methods according to a character-based and a word-based representation depending on the features used in each case. Their work implies that character-based approaches provide satisfactory results if character segmentation is easy to obtain, whereas word-based approaches can deal with touching characters efficiently and analyze the shapes of the words without explicit character recognition. In addition, a comparative study for segmentation and word spotting methods is presented in [46] for handwritten and printed text in Arabic documents. The segmentation techniques rely on horizontal and vertical profile features and scale space segmentation. The features under comparison are geometrical moments and word profiles, whereas the similarity computation is carried out using the cosine metric and dynamic time warping (DTW). An explicit view of the various aspects of a word

spotting system is presented by Marinai et al. [47]. In their work, the different features used for each technique are categorized according to the layer at which the similarity computation is performed (pixel/column features, connected components, word level features etc.). Image representations (i.e. feature vectors) with respect to the specific feature types are also analyzed along with the respective similarity measures. Another significant work of Tan et al. [48] underlines the necessity for content-based image retrieval as an economical alternative to OCR, relying on proper selection of features, representation and similarity measures. Word spotting is defined under a framework of categories with respect to the word image representation component.

Nevertheless, a considerable number of word spotting approaches proposed over the last years as well as several techniques involved for the improvement of the performance yet remain unexplored. In addition to this, after the recent explosion of the number of deep learning approaches proposed from the field of computer vision and machine learning for object detection and classification, their adaptation for document image word spotting actually began back in 2015 [49]. Our work herein also aims to review the recently proposed methods and complete the missing parts of other studies in the word spotting literature.

2.2 Challenges in document image word spotting

Keyword spotting in document images presents several challenges which are related to the nature of the original documents. In this section, we first investigate the various text sources used by KWS methods and subsequently overview the corresponding challenges.

2.2.1 Nature of text addressed in word spotting

Regarding the nature of documents which have been addressed so far by the research community for word spotting, we can distinguish various categories depending on factors such as the age of the text, its alphabet, the underlying language and the source which created the text (e.g. human or machine). Table 2.1 illustrates the various scripts addressed by most of the key methods for word spotting, during the period considered in this thesis.

Historical documents typically contain text written in a language that is no longer

Table 2.1: Text sources addressed by word spotting methods.

Publications	Context	Language	Script	Type
Aldavert et al. [50–52], Zagoris et al. [53]	Historical	English	Latin	Handwritten
Bogacz et al. [27], En et al. [54]	Historical	English	Latin	Handwritten
Zhang and Tan [55], Fornés et al. [56]	Historical	English	Latin	Handwritten
Roy et al. [57], Rothacker et al. [58,59]	Historical	English	Latin	Handwritten
Mondal et al. [60], Dovgalecs et al. [61]	Historical	English	Latin	Handwritten
Rath et al. [62], Zhong et al. [63]	Historical	English	Latin	Handwritten
Cao et al. [24], Wagan et al. [64]	Modern	English	Latin	Handwritten
Kumar et al. [65], Toseli et al. [66]	Modern	English	Latin	Handwritten
Retsinas et al. [67], Krishnan et al. [32,34]	Historical, modern	English	Latin	Handwritten
Almazán et al. [31], Liang et al. [68]	Historical, modern	English	Latin	Handwritten
Wilkinson et al. [69,70], Fischer et al. [71]	Historical, modern	English	Latin	Handwritten
Ghosh et al. [72,73], Retsinas et al. [74]	Historical, modern	English	Latin	Handwritten
Kessentini et al. [75,76], Choisy [77]	Modern	French	Latin	Handwritten
Howe [78,79], Frinken et al. [29]	Historical	English, German	Latin	Handwritten
Wolf et al. [80], Daraee et al. [81]	Historical	English, German	Latin	Handwritten
Puigcerver et al. [82,83], Riba et al. [84]	Historical	Spanish	Latin	Handwritten
Hast et al. [85,86], Villegas et al. [87]	Historical	Spanish	Latin	Handwritten
Fink et al. [88], Romero et al. [28]	Historical	German	Latin	Handwritten
Cheikhrouhou et al. [89], Chatbri et al. [90]	Historical, Modern	Arabic, French	Arabic, Latin	Handwritten, machine-printed
Lladós et al. [91], Wang et. al. [92]	Historical	English, Spanish	Latin	Handwritten
Oosten et al. [93], Der Zant et al. [94]	Historical	Dutch	Latin	Handwritten
Kovalchuk et al. [95], Almazán et al. [96]	Historical	English	Latin	Handwritten, machine-printed
Mondal et al. [97,98]	Historical	English, French	Latin	Handwritten, machine-printed
Sfikas et al. [43]	Historical	Greek	Greek	Handwritten, machine-printed
Rodríguez-Serrano and Perronnin [99]	Historical, modern	English, French, Arabic	Latin, Arabic	Handwritten
Sudholt et al. [4,100], Al-Rawi et al. [101]	Historical, modern	English, German Arabic	Latin, Arabic	Handwritten
Leydier et al. [102]	Historical	Middle English, Semitic, Chinese	Latin, Arabic, Chinese	Handwritten
Terasawa and Tanaka [103]	Historical	English, Japanese	Latin, Chinese	Handwritten
Sugawara et al. [104]	Historical	Japanese	Chinese	Handwritten
Abidi et al. [105], Sagheer et al. [106]	Historical	Urdu	Arabic	Handwritten
Khayyat et al. [107], Li et al. [108]	Modern	Farsi	Arabic	Handwritten
Kumar et al. [109], Wshah et al. [110]	Modern	English, Urdu, Hindi	Latin, Arabic, Devanagari	Handwritten
Srihari and Ball [111]	Modern	English, Urdu, Hindi	Latin, Arabic, Devanagari	Handwritten
Bhunia et al. [33]	Modern	Indian	Bangla, Gurumukhi, Devanagari	Handwritten
Huang et al. [112]	Modern	Chinese	Chinese	Handwritten
Giotis et al. [41]	Modern	Greek	Greek	Handwritten
Saabni et al. [113]	Modern	Arabic	Arabic	Handwritten
Shah et al. [114]	Modern	Pashto	Arabic	Handwritten
Can and Duygulu [115], Rusiñol et al. [116]	Historical	English, Ottoman	Latin, Arabic	Handwritten, machine-printed
Wei et al. [117–121]	Historical	Kanjur	Mongolian	Woodblock-printed
Ranjan et al. [122], Li et al. [123]	Modern	English	Latin	Machine-printed
Zagoris et al. [124], Bai et al. [125]	Modern	English	Latin	Machine-printed
Louloudis et al. [126], Roy et al. [127]	Historical	French	Latin	Machine-printed
Papandreou et al. [128]	Historical	French	Latin	Machine-printed
Gatos and Pratikakis [129]	Historical	German	Latin	Machine-printed
Sousa et al. [130]	Historical	Portuguese	Latin	Machine-printed
Marinai [131]	Historical	Latin	Latin	Machine-printed
Konidaris et al. [132], Kesidis et al. [133]	Historical	Greek	Greek	Machine-printed
Xia et al. [134]	Historical	Chinese	Chinese	Machine-printed
Hassan et al. [135], Krishnan et al. [136]	Modern	English, Indian, Gujarati	Latin, Bangla, Devanagari	Machine-printed
Shekhar et al. [137], Yalniz et al. [138]	Modern	English, Indian	Latin, Telugu	Machine-printed
Meshesha and Jawahar [139]	Modern	English, Amharic, Hindi	Latin, Amharic, Devanagari	Machine-printed

in use. Contrary to *modern* documents, the alphabet, the writing style or the accents are different. Historical documents usually suffer from degradations such as stained paper, faded ink or ink bleed through, winkles and unknown graphical symbols, as opposed to modern text, thus hampering the readability and in turn the word spotting process.

So far, word spotting has been applied to various scripts, such as Arabic, Chinese, Devanagari, Greek and Latin. These scripts differ from each other owing to factors such as the writing direction, the size of the alphabet (number of characters), possible diacritic marks (polytonic Greek text) and cursiveness. For example, documents in Arabic scripts are written from right to left, in horizontal direction and are fully cursive. On the contrary, text in Latin script is written from left to right in horizontal direction only, cursively in some cases. Chinese scripts contain thousands of characters and are written in two dimensions, either from left to right horizontally, or from top to bottom vertically. Devanagari scripts are written horizontally, from left to right in a complex cursive way, whereas Greek scripts are written from left to right without cursiveness. Furthermore, each separate character of the Chinese scripts has specific meanings or semantics, in contrast with the isolated characters of other scripts.

Many of the proposed techniques for word spotting in a specific language may be directly applied to a different language on the ground that it is written in a relevant script. However, the application of a word spotting method in different scripts is not straightforward, since it heavily depends on the features which are extracted before image matching takes place. For instance, profile or pixel-based features [62,129] are suitable for obtaining representations which allow for word spotting in heterogeneous documents regardless of the underlying language. This is contrasted with structural features and shape codes [125,140] which are defined to capture the specific shapes of the writing symbols of a language.

One other aspect of the documents addressed by word spotting techniques is related to the creation of the respective text. *Handwritten* documents, either historical or modern, always suffer from variability in writing style, not only among different authors but also for documents of the same writer. This is not the case though for *machine-printed* text where variations mainly concern the font type. An exception is the case of *woodblock-printed* documents of Chinese and Mongolian scripts which present intra-writer variability for the same author. Word spotting in handwritten text is generally considered more challenging than spotting printed text, as apart from

variations in writing style, handwriting is also unconstrained. For instance, words may be skewed, characters may be slanted, non-text content such as symbols may be present and letters may be broken or connected in a cursive manner. Nevertheless, historical printed documents also present challenges for word spotting because of degradations such as missing data, non-stationary noise due to illumination changes during the scanning process, low contrast, show through or warping effects etc.

Indexing documents contained in large databases around the globe is not the only area of application for image retrieval methods. Online handwritten text presents a growing significance due to the increasing use of PDAs, tablet PCs, and digital pens. Understanding such documents may be useful, for instance, in the case of a smart meeting room which allows participants to search, browse or organize handwritten notes taken with digital pens during a meeting. However, an important difference between online and offline text lies on the features which are extracted from each of the respective sources. Instead of focusing on color, texture or geometric shape, features related to the pen tip trace and the stroke's characteristics are extracted, such as its width, height, the pen's pressure and others. Example works for online text can be found in the literature, regarding either word spotting [141,142] or recognition [143]. Another interesting work from the online domain is proposed by Sudholt et al. [3]. Herein, offline query images from online trajectories obtained by a natural interface which accepts handwritten input are used to train convolutional neural networks (CNN). Initial variable sequences of online aforementioned features are pooled to compact representations which are embedded to a common subspace along with compact offline representations of handwritten document images. This common space allows for the querys' relevant images to be retrieved as a result of a nearest neighbor search between pairs of fixed-length representations. In this work though, we only consider offline documents.

2.2.2 Challenges addressed by existing methods

Degradations involved in historical documents, pre-hospital care reports and other text sources hinder the overall performance of a word spotting system. For example, low image quality directly affects the following segmentation and feature extraction stages of a word spotting system.

Apart from possible degradations, handwritten documents usually present high

variability in writing style, meaning, the same query word may differ substantially among its instances. This calls for features which are distinctive enough to be detected inside the query instances, yet not too dependent on a specific writing style. Most methods that deal with multi-writer word spotting rely on annotated data to learn a model able to capture the basic structure or semantic information of a word, regardless of the writing style.

The need for adequate training data poses another challenge for word spotting, since they are not always easy to obtain. For instance, handwritten documents are unconstrained and thus often render the transcription process difficult. To make it a tad harder for learning-based KWS systems, manual annotation of historical manuscripts at word or character level usually requires more effort than their modern equivalent, even for paleographers who have a hard time recognizing primary text components for the first time. Methods that do not require training using annotated images present a solid advantage in this respect.

Text cursiveness found in handwritten documents, overlapping sub-word components existing in Arabic scripts and many punctuation marks or graphical symbols lying in historical documents may lead to inaccurate segmentations. In that sense, methods that avoid potential error-prone segmentations tackle this challenge.

It is often expected that the user has to find a particular instance of a query in order to initiate the search for similar instances. In some cases though, it is more preferable for the user to insert an arbitrary string to be searched for. However, the amount of vocabulary in training sets is often far less than the complete vocabulary of a certain language. Therefore, unseen vocabularies in training samples might be taken as query keywords at the retrieval stage or appear in documents that will be retrieved. In this respect there are QBS methods which are not able to perform out of vocabulary (OOV) word spotting, namely, only a limited number of keywords, which are known during training, can be used as queries.

In Section 2.2.1, we mentioned the dependence of a word spotting method on a specific language, let alone a particular alphabet. A learning-based method able to perform well in different languages for a relevant script is not essentially suitable for a different script, unless new training data are used. In fact, with respect to the recently employed deep neural networks (DNN) as feature extraction models for handwritten KWS [4, 32, 63, 69] the language limitations still hold. This means that the corresponding deep features are able to perform well during retrieval, given their

representations are obtained from training data within the language-specific domain. In case that the domain involves historical documents written in English from a few authors or even artificially synthesized handwritten words in multiple writing-styles [144], the adaptation to a target script of the same language, but from the modern multi-writer text domain, is not ensured. Usually, deep features need to be trained from scratch or at least fine-tuned so as to encode the inherent writing style variations of the target domain. On the contrary, script-independent approaches deal more efficiently with this matter.

Chinese and Japanese documents have a large number of character classes (almost over 5000) and they present no explicit differences between inter-character and inter-word spaces. To cope with this challenge some keyword spotting methods follow the strategy of over-segmenting the text lines into primitive segments and adopt a character classifier to assign a small number of high-confidence classes to the input character pattern.

Word spotting methods need to be accurate enough for successful indexing while at the same time fast enough for high scalability. One way to achieve computationally efficient retrieval is to use fixed-length feature representations, since they are faster to compare than variable length sequences, as we will discuss in Section 2.4.2. Table 2.2 summarizes the aforementioned challenges along with the respective key methods which address them.

2.3 Basic document image analysis technologies involved

Although the intermediate stages of a word spotting system may vary across different methods, we can distinguish some common steps. Document images are initially preprocessed in order to enhance the subsequent feature extraction step. After appropriate features have been extracted, a common representation is selected to describe both the documents at a specific level (word, line or page) and the query, which in most cases is a single word provided either as an image or a text string. The next part lies on the matching algorithm applied between the representations of the query and the documents. This matching outcome is used at a later stage for retrieving the desired information. In the following, we will discuss some basic technologies involved during the preprocessing step.

Table 2.2: Challenges addressed by word spotting methods.

Challenges	Publications
Robustness to degradations	[4, 24, 27–29, 31, 32, 34, 43, 50–63, 67–74, 78, 79, 81–86, 86–89, 91–106] [3, 115–121, 126–134, 144–177]
Multi-writer language conditions	[24, 29, 31, 32, 34, 37, 51, 63, 65–67, 69–72, 74–77, 79, 81, 83, 84, 86–89] [4, 41, 91, 92, 99–101, 105–115, 117, 118, 144, 145, 147, 148, 150, 155, 159] [36, 160, 163, 165, 166, 168–170, 172–177, 177–180]
Learning-free or annotation-free methods	[51–56, 58, 61, 62, 64, 67, 78, 84, 85, 88, 90–92, 95–98, 102, 103, 105] [114–116, 118, 123–129, 132, 133, 136, 138, 147, 148, 150, 151, 154, 157, 159] [161, 163, 164, 168, 173, 174, 176, 177]
Segmentation-free methods	[27, 54, 55, 58, 59, 61, 70, 72, 73, 84–86, 88, 90, 95, 96, 102, 116, 129, 148, 154] [156, 168, 170, 172, 175–177]
Out-of-vocabulary (OOV) KWS for QBS scenario	[29, 31, 32, 34, 37, 43, 50, 57, 59, 63, 66, 68, 69, 71, 72, 75, 76, 79, 82, 83, 87] [3, 4, 100–102, 104, 120, 125, 144, 150, 153, 156, 159, 160, 165, 166, 169, 175] [36, 172, 178]
Script-independence	[33, 51, 67, 89, 99, 101–103, 109–111, 115, 116, 135–139, 148, 163, 169]
Chinese-like script	[102–104, 112, 117–119, 121, 134, 162]
Scalability of image representations at word/line/page lvl	[31–34, 37, 50–54, 56, 57, 59–64, 67–70, 72–75, 79, 81, 82, 84, 86, 88, 89, 91] [3, 4, 43, 95, 96, 100, 101, 104, 116, 119–121, 126, 127, 129, 136, 144, 152, 153] [36, 155, 156, 159, 160, 162, 163, 165–175, 177–180]

2.3.1 Binarization

Binarization is the starting step of most word spotting systems and refers to the conversion of the original input to a binary black-and-white image. It can provide a good starting point for segmentation as well as feature extraction. For instance, some methods which perform text-line segmentation using connected components analysis require the documents to be properly binarized. Similarly, contour-based features extracted from skeletons or outer contours are heavily dependent on the binarization outcome.

Otsu's global thresholding [181] is one of the most commonly used binarization techniques in the literature [25, 54, 124, 140, 147, 164, 182–184]. This method selects a global threshold value from all possible thresholds as the one minimizing the intra-class variance of the thresholded black and white pixels. Can et al. [115] obtained similar results to Otsu's method using another global thresholding technique in which the threshold is based on the mean intensity value of the gray-scale image. Other global thresholding approaches can be found in [72, 85, 95, 104, 130, 150, 151, 158, 161, 185]. Often, images are initially enhanced using Gaussian filtering [85] and similar edge enhancement and smoothing operations [150, 151, 161] before applying global thresholding.

In the case though of degraded document collections which usually suffer from non-uniform illumination, image contrast variation, bleeding-through or smear effects, more efficient local thresholding techniques are required. For instance, Sauvola's technique [186] calculates a local threshold which is adapted to the neighborhood of each pixel according to the local mean value and the local standard deviation inside the neighborhood which is defined by a sliding window. Methods based on local thresholding can be found in [65, 68, 187–189]. Some methods of this family also include an image enhancement step. Fink et al. [88] preprocess images to improve the overall contrast between the script and the document background. To this end, they employ histogram equalization to the intensity channel in an YCrCb color space and subsequently use a 9x9 median filter to reduce the background noise. Kumar et al. [65] normalize the background light intensity using an adaptive linear or non-linear function [190] that best fits the background. The background normalized image is further enhanced by histogram normalization. Finally, the normalized image is binarized using an adaptive thresholding algorithm. Cao et al. [24] follow a probabilistic approach [191] to binarize documents and remove inherent grid lines. They model degraded images with Markov random fields (MRFs) where the prior is learnt from a training set of high quality binarized images, whereas the probabilistic density is learnt on-the-fly from the gray-level histogram of input images. A soft assignment variant of Sauvola's local threshold is employed in [155].

Several state-of-the-art approaches for binarizing degraded documents rely on hybrid schemes which combine global and local thresholding. The authors in [98, 129, 132, 133] use the technique proposed in [192] which consists of five steps: a preprocessing procedure using a low-pass Wiener filter, a rough estimation of foreground regions, a background surface calculation by interpolating neighboring background intensities, a thresholding by combining the calculated background surface with the original image and finally a post-processing step that improves the quality of text regions and preserves stroke connectivity. Wei et al. [117] make use of three global thresholding methods to extract regions of interest (ROI) from gray-level images. Each ROI is then processed by a modified Sauvola's algorithm with variant sizes of the small windows. Howe [78] employs the method proposed in [193] which optimizes a global energy function based on the Laplacian operator upon the local likelihood of foreground and background labels, the Canny edge detection to identify likely discontinuities and a graph cut implementation to find the minimum energy

solution of the objective function. Hast, as well as Vats et al. [86,168] propose a background noise removal using a simple two band-pass filtering approach, as proposed in [194]. A high frequency band-pass filter is used to separate the fine detailed text from the background, whereas a low frequency band-pass filter is used for masking and noise removal. The background removal is performed in such a way that the gray-level information is not affected. This renders the following keypoint detector and the employed descriptor more informative.

However, there is often a tradeoff between the amount of missing data and accurate data after binarization is applied and therefore some works [55, 58, 62, 102, 103, 114, 116, 145, 148, 157, 195–198] prefer to perform directly on the gray-scale image. For example, Zhang et al. [55] propose an illumination invariant descriptor of gray-scale document images using features extracted from keypoints. If the images suffer from low resolution, the authors report a low number of detected keypoints which in turn yields a reduced number of retrieved query instances, despite the high accuracy of those retrieved. Leydier et al. [102, 196] prefer to separate the text from the background using a gradient norm threshold instead of binarizing the document image. This renders their proposed gradient-based features more informative in high magnitude zones computed on the gray-scale image. Similarly, Terasawa and Tanaka [103] deem a background removal more suitable for their method. The background is removed using a simple thresholding technique such that the graylevel information which is important for the proposed gradient-based features is not affected. To exploit fast matching algorithms which can only be applied to binary images, Shah and Suen [114] extract features directly from gray-scale images and then convert the resulting feature vectors into their binary equivalents using an encoding scheme with four bits per feature value. A drawback of this approach is that the final feature vector has high dimensionality. Cao et al. [195] consider the gray-scale images more preferable when dealing with heavily degraded documents, such as carbon medical forms, where the binarized version is not even readable by a human.

Convolutional neural networks recently achieved high performance to document image binarization [199, 200]. Since the output of binarization is of the same size as the input image, well-established deep learning models were successfully applied. An example is the winner of the recent DIBCO competition [201] which uses the U-Net convolutional network architecture [202] for accurate pixel classification. In [203], the fully convolutional neural network is applied at multiple image scales. Deep

encoder-decoder architectures are also employed for document image binarization in [204, 205]. A hierarchical deep supervised network is proposed by Vo et al. [200] and achieves seminal performance on state-of-the-art benchmark data sets. In [206], the Grid Long Short-Term Memory (Grid LSTM) network is used for binarization though performing a tad lower than Vo's method [200].

In the same spirit with traditional global thresholding methods, which prefer an image enhancement step, prior to binarization, He et al. [207] propose a novel deep-Otsu framework. Therein, instead of training the neural network to learn the labels of each pixel so as to estimate the binarized output, the output of their model is a latent uniform and clean version of the input image, which represents an internally enhanced version of the image, rather than binary maps. This allows the network to learn the degradations, namely, the differences between the degraded and clean images. In other words, the neural network is trained to correct degradations by iteratively using the enhanced image output as a new input image, whereas the final binarization is performed using Otsu's global-thresholding technique.

Although the recent explosion of utilizing deep learning models for binarization has achieved seminal performance, most of the proposed deep learning methods for document image KWS [4, 32, 34, 36, 63, 69, 70, 74, 101, 144, 159, 160, 165, 166, 169, 170, 172–175, 178] prefer features extracted directly from the unprocessed input images without relying on a binarization step [161] at all. Most of these works argue that deep features detected on preprocessed images might miss distinctive information found in original input images [80].

Finally, another family of methods [105, 113] work on both gray-scale and binary images aiming to combine the advantages of each type. Abidi et al. [105] employ a set of profile-based features which can be extracted from either gray-scale or binary images to match partial words in Arabic script. To examine the discriminative power of each independent feature, they evaluate the retrieval percentage of five features obtained from binary images and one feature extracted from the gray-scale version, which proved to outperform the other five features. Nevertheless, the authors report that the combined information from all features improves the word spotting performance. Saabni and Bronstein [113] propose a multi angular descriptor of either binary or gray-scale word images. The descriptor is based on multiple view points obtained from rings out of the shape of the word and therefore is not significantly affected by the binarization step.

2.3.2 Segmentation

Segmentation-based word spotting methods involve a segmentation preprocessing stage in order to segment the document pages at word or line level. Although segmentation can be considered as a simple task for modern machine-printed documents, segmentation of handwritten or historical documents is still an open research problem due to the significant challenges that are involved. These include variations in inter-line or inter-word gaps, overlapping and touching text parts, existence of accents, punctuation marks and decorative letters, local text skew and slant.

In the following, we present a categorization of the general text line techniques together with one representative reference per each category. (a) Projection-based methods: the horizontal image projections are analyzed in order to detect hills (correspond to text lines) and valleys (correspond to white spaces between text lines). Although these methods are usually applied to machine-printed documents, they can also be used for handwritten documents [208]. (b) Smearing methods: the white runs in a certain direction are analyzed and eliminated under several conditions [209]. (c) Grouping methods: low-level elements such as pixels or related components are grouped together based on several rules [210]. (d) Methods based on Hough transform: a set of points is projected to the Hough space in order to detect lines [211].

Concerning word segmentation, the proposed techniques usually first calculate the distances of adjacent components using the bounding box, the Euclidean, the run-length or the convex hull distance [212]. At a next step, these distances are classified as inter-word or intra-word [213].

Some segmentation-based word spotting methods assume that datasets are already segmented to text lines or words while others perform a respective segmentation step. Example word spotting methods based on horizontal projection profiles for text line separation, followed by vertical profiles for word segmentation can be found in [68,135,182,214–216]. Rodriguez-Serrano and Perronin [22] use horizontal projection profiles to obtain text lines. For each line they compute the convex hulls of connected components and define a distance between neighboring components as the minimum distance between their convex hulls. Distances larger than a threshold are likely to correspond to word gaps. Kumar et al. [65] extract text lines using the algorithm proposed by Shi et al. [217] which uses a steerable filter to convert a down-sampled version of the input document image into an Adaptive Local Connec-

tivity Map (ALCM). Connected component based grouping is done to extract each text line. Word segmentation is then done by finding convex hulls for each connected component and learning the distribution over the distances between the centroids of the convex hulls for within and between word gaps.

A combination of vertical and horizontal projection profiles, as well as zoning techniques of upper, middle and lower zones of word images are proposed in [33] for character segmentation of Indic scripts. Similar vertical and horizontal profiles are employed by Stauffer et al. [161], for word segmentation in historical Latin scripts.

The Run Length Smoothing Algorithm (RLSA) [218] is a common smearing technique for segmenting document pages into text lines and words. RLSA examines the white runs existing in the horizontal and vertical directions. For each direction, white runs with length less than a threshold are eliminated. The horizontal and vertical length thresholds are usually defined proportionally to the average character height. The application of RLSA results in a binary image where characters of the same word become connected to a single connected component. Then, a connected component analysis is applied in order to extract the final word segmentation result. Example works using RLSA can be found in [97,98,132,133]. RLSA works well for printed documents but usually presents poor results in handwritten historical documents where inter-word spaces are variable. Mondal et al. [98] evaluate a number of DTW-based sequence alignment techniques under conditions of perfect (manual) and error-prone (RLSA-based) word segmentations and confirm that DTW works well only in the first case. Otherwise, they propose a Continuous Dynamic Programming (CDP) method which performs robust partial matching at line (or piece of line) level.

Most works in Arabic scripts [106, 107, 183, 219] are only able to perform on partial word level. Pieces of Arabic Words (PAW) are obtained either manually or from connected component analysis on the segmented words. Each word in the Arabic script consists of one or more PAW, each of which contains only one major connected component (CC) and some or none minor CCs. These minor CCs are often called diacritics and dots. Major and minor CCs can be distinguished by their size and location. Khayyat et al. [107] smear the documents with a morphological dilation using a binary dynamic adaptive mask [220] to extract text lines. Then they extract major and minor components from PAW.

Chinese scripts also show variations between inter-character and inter-word spaces. Most keyword spotting methods follow the strategy of over-segmenting the text lines

into primitive segments. For instance, Huang et al. [112] segment the document image into text lines using a graph-based clustering algorithm [221]. Each line is then over-segmented into primitive segments using the algorithm of [222]. Candidate characters generated by concatenating consecutive segments form a candidate segmentation lattice.

In a language independent scenario, Srihari et al. [111] perform text line segmentation using a clustering method. For word segmentation, the problem is formulated as a classification problem as to whether or not the gap between two adjacent CCs in a line is word gap or not. An artificial neural network with features characterizing the CCs was used for this classification task.

Traditional segmentation techniques which have been successful in modern document images, relying on projection profiles or connected components, are likely to fail for historical documents. In such cases, these techniques have to be manually tuned to the document collection's specificities. For this reason, several deep learning-based approaches have emerged to deal with the large number of intrinsic layout properties of a particular document collection, using labelled data. For instance, Wilkinson et al. [223] use a CNN for classifying segmentation word hypotheses where the visual word appearance is learned from annotated sample data. This method also requires a prior detection step (CNN) of candidate word bounding-boxes. In addition, Chen et al. [224] propose an unsupervised feature learning method for page segmentation of historical handwritten documents available as color images. They perform page segmentation at pixel level, classifying it as either periphery, background, text block, or decoration. To this end, convolutional autoencoders are used to learn features directly from pixel intensity values without any assumption of specific topologies and shapes of the underlying text.

Krishnan et al. [144] make use of a simple multi-stage bottom-up approach similar to that of Louloudis et al. [211] by forming three sets of connected components (CCs) on the binarized image based on its sizes. Given the bounding boxes of a set of CCs and its line associations, they analyse the inter CC spacing and derive multiple thresholds to group it into words. This results in multiple word bounding box hypotheses with a high recall. Possible erroneous segmentations are alleviated by the following proposed matching step.

However, most of the recently proposed state-of-art methods for word-based handwritten KWS assume perfect word segmentations, which is a usually a hindrance for

historical documents. Intrigued by this observation, Dey et al. [225] analyze the robustness of several state-of-the-art KWS methods [4,31,197] under improper segmentation. The authors suggest that the best performing word spotting method actually depends on the quality of the segmentation.

2.3.3 Normalization

The segmentation is usually followed by a normalization step in which several variabilities are removed. For instance, handwritten documents present challenges such as text skew and slant or warping effects accrued during the scan process. Wang et al. [226] handle text skew by combining projection profiles with Hough transform to separate the text according to the skew angle of each line. To cope with different writing styles, most approaches based on line segmentation [29, 71, 227], as well as word-based methods, such as the works of Rodríguez-Serrano and Perronnin [22,99], determine the skew angle by a regression analysis based on the bottom-most black pixel of each image column extracted via a sliding window. Then, the skew of the text line is corrected by rotation. After estimating the slant angle based on a histogram analysis, a shear transformation is applied to the image. Moreover, a vertical scaling procedure is applied to normalize the height with respect to the lower and upper baseline and finally, horizontal scaling normalizes the width of the text line with respect to the estimated number of letters. Scale normalization at word level is also applied for handwritten and printed documents. In [132, 133], the segmented words are resized to fit in a fixed bounding-box while preserving their aspect ratio, whereas in [95] each candidate word is resized to fit a fixed-size rectangle regardless of its size and aspect ratio. In [161] the inclination of the document (text skew), is estimated on the lower baseline of a line of text and then corrected on single word images. Retsinas et al. [155] propose a main-zone normalization by detecting the main-zone which best describes a word image as a text-line. This is done using a line fitting regression model [67] based on iteratively re-weighted least squares. Then, skew correction is performed using the slope of the detected main-zone, as well as a vertical normalization of the image by moving the main zone at the center of the generated normalized image.

With the recent advent of deep learning-based KWS methods, a standard solution for example model architectures is to normalize input images to fixed size [69, 104,

121, 166]. For instance, Wei and co-workers propose a normalization by resizing all input images to a standard size of 310 pixel width and 50 height in [121], whereas in [120], they resize all input images so that they have the same width (either pure or by padding white pixels) and aspect ratio. Wicht et al. [150, 151] normalize the word images to remove the skew and slant of the text using [228]. The word images are then resized to a third of their height. In these normalization strategies though, the majority of dataset word images has either to be scaled or cropped to a fixed size. This leads to possible distortions or deletions of important parts of a word image. This is usually the case for CNNs which are fed with images of the same width and height. This resizing might distort similar semantic aspects in the visual domain. In order to tackle this, Sudholt et al. [4] make use of a Spatial Pyramid Pooling (SPP) layer [229]. This type of layer allows CNNs to accept arbitrarily sized input images and still produce a constant output size which is essential for training the NN. Such pooling techniques are also adopted by recently proposed seminal works for KWS [100, 165].

2.4 Keyword spotting system architecture

In this section, we examine the main steps of the word spotting pipeline. Figure 2.1 illustrates a general purpose word spotting system where the whole procedure is divided in an offline and an online phase. In the offline stage, features are extracted from word images, text lines or whole pages which are then represented by feature vectors. In the case where training data are used, traditional feature vectors are usually modelled with statistical models (e.g. HMMs). More recent learning-based techniques exploit well-established deep architectures (CNNs) as feature extraction models for KWS, leading to compact fixed-length feature vectors. Typically, in the online phase, a user formulates a query either by selecting an actual example from the document collection (QBE), or by typing an ASCII text word (QBS). Depending on the query type, a common representation with that of the offline phase is used to describe the query and then a matching process is applied between these representations in order to obtain a similarity or relevance score which in turn yields a ranking list of results according to their similarity with the query.

The most common distinction of word spotting approaches depends on how the

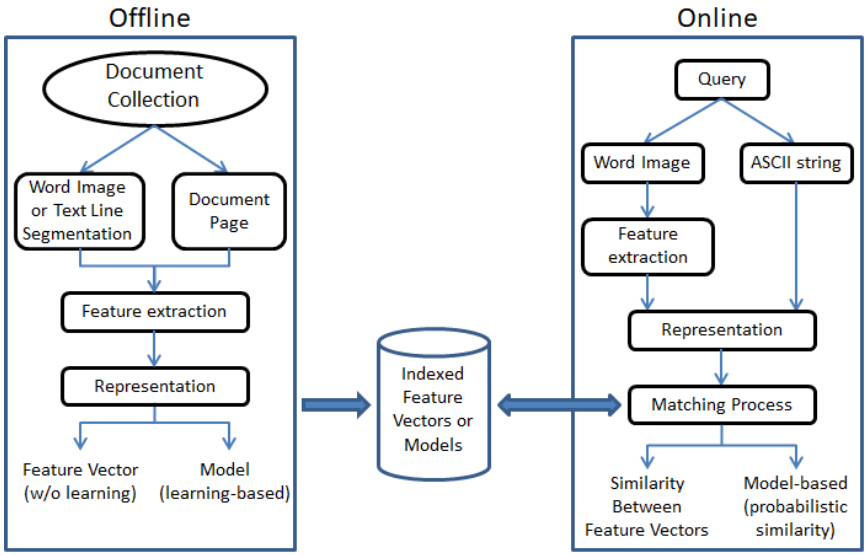


Figure 2.1: General word spotting system architecture.

input is specified. Each type (QBE or QBS) has its own merits and handicaps. One obvious drawback of QBE methods is that the search is constrained for words that appear at least once in a document collection since an actual instance of the query word is required to trigger it. QBS approaches on the other hand allow arbitrary textual queries without the need to find a particular query occurrence. Herein, the keyword representation is usually accrued from trained character models. However, in the case where labeled data are not available or inadequate, an alternative solution is to artificially generate the query input in ASCII text from character images selected either manually or in a semi-supervised way. In this context, the word spotting task is also referred to as *word retrieval* [68,102,104,123,132,133,146,230,231]. For instance, Sugawara et al. [104] generate images of query texts using generative semi-supervised models [232]. The conditional generative model infers the latent variables which represent the underlying structure of input character images, like character shape, glyph, and handwriting, and generates new images that have characteristics of input images from combination of the latent variables and several class labels.

In the following, we review the main steps of a word spotting system with respect to the extracted features, the representation defined to describe both documents and queries at a specific level and the similarity measures used to compare them. A concise view of some key approaches considered in this work is presented in Table 2.3. Since most word spotting approaches belong to various distinct categories, we mainly divide them according to the representation used in each case.

Table 2.3: Overview of key techniques according to the core steps of the KWS pipeline.

Publications	Query	Features	Representation	Similarity
[51, 61, 91, 116, 197]	QBE	SIFT	BoVW	Cosine, Euclidean
[182]	QBE	SIFT	BoVW	Symmetric KL-divergence
[138]	QBE	SIFT	BoVW	Longest Common Subsequence
[119, 162]	QBE	SIFT	BoVW/RNN	Euclidean
[58, 88, 177]	QBE	SIFT	BoF-HMM	Viterbi decoding probability
[59]	QBS	SIFT	BoF-HMM	Viterbi decoding probability
[66, 71, 83, 87, 227]	QBS	Geometrical	HMM	Viterbi decoding probability
[233]	QBE	Local Gradient Histogram (LGH)	HMM	Viterbi decoding probability
[234]	QBE	Geometrical, pixel counts	SC-HMM, HMM	Viterbi decoding probability
[22]	QBE	Geometrical, pixel counts, LGH	SC-HMM, HMM	Viterbi decoding probability
[63]	QBE	CNN pixel values	NN internal representation	NN learned similarity
[4]	QBE/QBS	CNN pixel values/PHOC/SPP	Fixed-length vector	Bray-Curtis dissimilarity
[100, 159, 173, 174, 178]	QBE/QBS	CNN pixel values/PHOC/TPP	Fixed-length vector	Cosine distance
[70]	QBS	CNN-ResNet pixel values/PHOC/DCToW	Fixed-length vector	Cosine distance
[81, 175]	QBS	CNN-ResNet pixel values/DCToW	Fixed-length vector	Cosine distance
[170]	QBS	CNN-ResNet pixel values	Fixed-length vector	Cosine distance
[121]	QBE	Hybrid DNN/CNN (pixels/activations)	Fixed-length vector	Euclidean, Cosine distance
[166]	QBE/QBS	CNN (visual), RNN/GRU (text)	Fixed-length vector	Euclidean
[153]	QBS	CNN (visual), LSDE (text)	Fixed-length vector	Euclidean, Levenshtein (string)
[67]	QBE	Gradient-based (POG)	Fixed-length vector	Euclidean
[235]	QBE	Zoning/NN layer activations	Fixed-length vector	Euclidean
[69]	QBE/QBS	CNN pixel values/PHOC/DCToW	Fixed-length vector	Euclidean
[32, 34, 165]	QBE/QBS	NN layer activations	Fixed-length vector	Cosine distance
[179, 180]	QBE	Zoning/NN layer activations	Fixed-length vector	Cosine distance
[31, 43, 236]	QBE/QBS	SIFT/PHOC	Fixed-length vector	Euclidean
[72, 237]	QBE	SIFT/PHOC	Fixed-length vector	Euclidean
[95]	QBE	HoG, LBP	Fixed-length vector	Euclidean
[53, 106, 118]	QBE	Gradient/profile-based	Fixed-length vector	Euclidean
[135]	QBE	Shape Context	Fixed-length vector	Euclidean
[126, 238]	QBE	Adaptive Zoning	Fixed-length vector	Euclidean
[56]	QBE	Blurred Shape Model	Fixed-length vector	Euclidean
[239]	QBE	Characteristic Loci	Fixed-length vector	Euclidean
[64]	QBS	Gradient-based	Fixed-length vector	Euclidean
[132, 133]	QBS	Standard Zoning	Fixed-length vector	Euclidean
[195]	QBS	Gabor (gray-scale)	Fixed-length vector	Euclidean
[230]	QBS	Global, Profiles	Fixed-length vector	Dot Product
[124]	QBE	Global, Profiles	Fixed-length vector	Minkowski distance
[129]	QBE	Standard Zoning	Fixed-length vector	Square distance-based
[114]	QBE	Zoning/Profile-based	Fixed-length vector	Correlation-based
[96, 240]	QBE	HoG	Fixed-length vector	Cosine Distance
[148]	QBE	HoG, Scale space pyramid	Fixed-length vector	Euclidean-based
[215]	QBS	Moment-based	Fixed-length vector	Cosine Distance
[55, 142]	QBE	Dali, SIFT	Heat Kernel Signature	Euclidean-based
[90]	QBE	Point Distribution Histogram	Variable-length vector	Histogram Intersection
[150, 151]	QBE	NN layer activations	Variable-length vector	DTW-based
[60, 97, 241]	QBE	Profiles, Moments, Gabor	Variable-length vector	DTW-based
[62, 117, 242]	QBE	Word profiles	Variable-length vector	DTW-based
[105, 183]	QBE	Global, profile-based	Variable-length vector	DTW-based
[113]	QBE	Multi Angular Descriptor	Variable-length vector	DTW-based
[128]	QBE	Adaptive Zoning	Variable-length vector	DTW-based
[103]	QBE	Slit Style HoG	Variable-length vector	DTW-based
[139]	QBS	Profiles, Moments, DFT	Variable-length vector	DTW-based
[243]	QBE	Wavelet coefficients	Variable-length vector	Earth Movers Distance
[102, 196]	QBS	Gradient-based (ZoI)	Variable-length vector	Cohesive Elastic Matching
[154]	QBE	Gradient-based (CCs)	Variable-length vector	Euclidean-based
[163]	QBE	PoG/Zoning	Variable-length vector	Selective Matching
[125]	QBS	Column-based	Word Shape Coding	Sequence alignment
[123]	QBS	Column-based	Word Shape Coding	Edit Distance
[140]	QBS	Character shape features	Word Shape Coding	Edit Distance
[189]	QBE	Profile-based	Graph-based	Edit Distance-based
[127]	QBE	Character primitives	Graph-based	Edit Distance-based
[92, 216, 226]	QBE	Structural, Shape Context	Graph-based	Edit Distance-based
[84]	QBE	Graphemes of convex groups	Graph-based	Edit Distance-based
[161]	QBE	Keypoints, Projections	Graph-based	Edit Distance-based
[80]	QBE/QBS	Keypoints, Projections, CNN	Graph/CNN PHOC-fixed	Cosine Distance

2.4.1 Feature extraction

The appropriate selection of features has a great impact on the performance of a word spotting system as well as numerous other computer vision applications. Girshick et al. [244] state that progress made on various visual recognition tasks in the last decade relied considerably on the use of SIFT [245] and HoG [246] features. Particularly in word spotting applications, Rodríguez-Serrano and Perronnin evaluate the performance of different feature types using DTW [233] and HMMs [22]. In both cases, the authors show that their proposed local gradient histogram features outperform other profile-based or geometrical features. Other word spotting approaches [91, 184, 247] also confirm the effect of features on the final performance.

In general, we can distinguish two broad categories of features. *Global* features are extracted from the object of interest which can be either a word image or a document region as a whole. Examples of such features are the width, height, or the aspect ratio of the word image, the number of foreground pixels, moments of background pixels and others. On the contrary, *local* features may be detected independently at different regions of the input image, which may be a text line, word or primitive word parts. For instance, the pixel densities, the position or the number of holes, valleys, dots and crosses at keypoints or regions are local features. We should note here that approaches based only on global features are obsolete in the recent literature.

Local features from the other hand are very common and are used either solely or in combination with global features. Local features extracted from raw pixels to directly represent document images were outperformed throughout the years by higher level features. A typical example of higher level features comprise the upper and lower word profiles, the number of foreground pixels and the number of transitions from background to foreground. These column features, also known as word profiles, were popularized by Rath and Manmatha [62, 248] and adopted by many other researchers. They are extracted from each column of the word image or the text line and concatenated to variable-length sequences of features which describe text regions (e.g. words) as a whole.

Geometrical column features are also widely used with the sliding window approach in [29, 66, 71, 83, 87, 227, 249, 250]. These typically contain three global and six local features. The global features are the moments of the black pixels distribution within the window. The local features are the position of the top-most and that of

the bottom-most black pixel, the inclination of the top and bottom contour of the word at the actual window position, the number of vertical black/white transitions and the average gray scale value between the top-most and bottom-most black pixel. These features also form a variable-length sequence of features, usually modelled with HMMs or NNs, which can adapt better to writing style variations.

Zoning features [126,128,132,133] have also been proved quite efficient statistical features which provide high speed and low complexity word matching. They are usually calculated by the density of pixels or other pattern characteristics in the zones that the pattern frame is divided. Their application to printed documents yields satisfactory results which is not always the case for handwritten documents.

Neural network-based models typically use raw pixel intensity information as their input [4, 235]. From a theoretical stand-point, using image information with little or no preprocessing is a valid practice in the case of NNs, as intermediate net layer activations can be considered as the image features, dynamically learned during network training. Following this rationale a step further, in a number of works the NN is used purely as a feature extractor [32,100,159,173,174,178,235]. The image is fed-forward through the NN, and the activations of one or more layers are used to form feature vectors.

A recent trend in deep NN-based (DNN) KWS follows the combination of zoning along with NN layer activations in particular (vertical or horizontal) regions of the input as features, extracted from the segmented NN intermediate layer feature maps into such zones [179, 180, 235]. In this case, separate DNN features extracted from each zone are fed after the last feature extraction layer (e.g. last convolutional layer), to a fully connected layer, so as to form a compact feature vector which encodes both image features from each layer of abstraction as well as their spatial information.

Gradient-based features are also widely used as higher level local features. This family of features tends to be superior over the word profiles for multi-writer word spotting since it can also capture the directions of the strokes, which are discriminative for distinguishing different words. Typical examples of this type are the Histograms of Gradients (HoG) [246] as well as the features extracted using the Scale Invariant Feature Transform (SIFT) [245]. Similar to SIFT, HoG computes a histogram of gradient orientations in a certain local region. One of the main differences between SIFT and HoG is that HoG normalizes such histograms in overlapping local blocks and makes a redundant expression. HoG features are computed in a rigid grid while

SIFT features are either densely sampled in local patches of the image or extracted from keypoints (e.g. corners). Several variants of HoG and SIFT features have been successfully used for word spotting [103, 233, 247].

Pattern features are computed by placing primitives in local image regions and analyzing the relative differences. Pattern analysis is quite useful in texture information representations. Examples of this type are the Local Binary Patterns (LBP) [251] and Gabor features [252]. LBP features mainly focus on the gradient information about the local pattern and they can preserve more local information than the features extracted from only one pixel wide column. They are usually combined with gradient-based features to yield a more discriminative representation for word spotting [95]. Gabor features are related with Gabor wavelets for human perception simulation, which are computed by convolving images with Gabor filters. Application of this type of features can be found in [60, 195].

Apart from statistical features (e.g. HoG), structural features, such as graphemes from connected components, adjacent line segments or graphs arranged into tree structures have also found their way in word spotting. The main motivation behind selecting such features is that the structure of the handwriting is more stable than the pure appearance of its strokes. This is especially important when dealing with the elastic deformations of different handwriting styles. Such structural features may be extracted from the contour [75, 92, 113] or the skeleton [41, 68, 78, 80, 84, 90–92, 161] of an image. Usually, these features assume a binarization step of the input images prior to their extraction.

Advanced gradient, structural and concavity (GSC) features [253] are a good choice for Arabic scripts [109–111]. They are multi-resolution features that combine three different attributes of the character shape, the gradient (representing the local orientation of strokes), the structural features (which extend the gradient to longer distances and provide information about stroke trajectories) and the concavity features (which capture stroke relationships at long distances).

Finally, a recently proposed technique introduced the idea of using attributes as features for word spotting [236]. Attributes are semantic properties that can be used to describe images and categories since they can transfer information from different training words and lead to compact signatures. The selection of these attributes is usually a task-dependent process, so for their application to word spotting they are defined as word-discriminative and appearance-independent properties. In a nutshell,

they combine visual (features) and textual (labels) information to encode a word image representation, which actually transforms images to a string embedding space, thereby enabling both QBE and QBS scenarios. Moreover, attribute-based features are robust to writing styles and fast to compare.

2.4.2 Representation

After a set of features has been extracted, a suitable representation of their values has to be defined in order to allow efficient comparison between the query image and the documents at a specific level. *Variable-length* representations describe word images or text lines as a time series, usually using a window that slides over the image in the writing direction. In contrast, *fixed-length* representations extract a single feature vector of fixed size which characterizes the document region as a whole.

Variable-length representations adopt the sequential nature of handwritten words formed by the concatenation of individual characters. Nevertheless, since two words may have different numbers of characters or widths, defining a distance between feature vectors is not straightforward. In this case, a standard practise is to use sequence alignment techniques such as the DTW.

Probabilistic representations are also very popular and have proven to be more effective than variable-length vectors obtained directly from image features. These typically consist of character or word models which represent the sequential features and are trained from annotated data usually based on hidden Markov models [58, 66, 71, 77, 83, 87, 99] as well as neural networks [29, 30, 117].

Word Shape Coding (WSC) [123,125,140] is also another way to represent sequential features on stroke level. Particularly, each word image is encoded as a sequence of symbols roughly corresponding to characters. In most cases the symbol set has a lower cardinality with respect to the character set in the original language and it is easier to recognize. Each word is represented by a symbol string. Due to the reduced number of symbol classes, one-to-one correspondences between a symbol and a character are uncertain and therefore a symbol string can be mapped to several words.

A growing interest in *graph-based* representations [68, 84, 92, 161, 189, 216, 226] is also reported by the research community. Such representations are defined on structural features extracted from connected components or strokes, along with their spatial arrangements. Although structural features are considered language dependent

as they capture the specific shapes of the writing symbols of a language, graph-based representations of such features may perform well in terms of speed and accuracy under large variability in writing style.

Fixed-length representations present a clear advantage over sequential representations, as the fixed-size feature vectors can be compared using standard distances such as the Euclidean distance, or any statistical pattern recognition technique. This way image matching is reduced to a much faster nearest neighbor search problem. In some cases, fixed-length descriptions are formed directly from the extracted features without involving some learning step.

There are cases though where variable-length representations are pooled to fixed-length feature vectors using an encoding scheme. In this spirit, many researches from the document analysis community deem the word spotting problem as an object detection task based on matching techniques between features extracted from keypoints. However, the keypoint matching framework presents the same drawbacks as the sequential methods since an alignment between the keypoint sets has to be computed. In order to avoid exhaustively matching all keypoint pairs, the bag-of-features paradigm from the information retrieval field was adopted as the *Bag-of-Visual-Words* (BoVW) [254]. This consists in a holistic and fixed-length image representation while keeping the discriminative power of local features such as SIFT. The BoVW representation relies on the following steps:

1. Keypoints are extracted from the document images at a specific level using an appropriate detector.

2. Keypoints or shape descriptors evaluated upon them, are clustered and similar descriptors are assigned to the same cluster. Each cluster corresponds to a visual word that is a representation of the features shared (in terms of average or median value) by the descriptors belonging to that cluster.

3. Each image region is described by a vector containing the occurrences of each visual word in that image.

Instead of using keypoints to build the visual codebook, recent approaches prefer to densely sample features over regular fixed-size grids [52, 116, 119, 152, 162, 177, 197] since the larger amount of descriptors extracted from an image, the better the performance of the BoVW model is. Descriptors having a low gradient magnitude are

directly discarded. One main drawback of BoVW models is that they do not take into account the spatial distribution of the features. In order to add spatial information to the orderless BoVW model, Lazebnik et al. [255] proposed the Spatial Pyramid Matching (SPM) method which takes into account the visual word distribution over the fixed-size patch by creating a pyramid of spatial bins. This SPM technique is adopted by most of the recent BoVW-based methods.

Another way to form fixed-length descriptions from variable-length representations is the *Fisher vector* [256]. Assuming that a set of features such as SIFT are extracted from a dense grid, corresponding for instance to a word image, the next step is to train a Gaussian mixture model (GMM) using SIFT descriptors from all input images of the document collection. Subsequently, Fisher vectors are calculated for each image as a function of their SIFT description and the gradients of the GMM with respect to its parameters. This yields a fixed-length, highly discriminative representation, that can be seen as an augmented BoVW description which encodes higher order statistics. Fisher vectors have previously been used with success in various fields of computer vision [257, 258].

Relevant examples of pooling features to fixed-length vectors can be found in [113]. The authors employ the Boostmap algorithm described in [259] to embed the feature space of variable-length representations which are matched with DTW into a Euclidean space for faster comparisons. In the same direction, Wei et al. [118] use DFT on variable-length word profiles to create fixed-length vectors.

Regarding more recent approaches for handwritten KWS based on deep learning, there are two standard practices for image representation. On one hand, approaches such as the work of Sudholt et al. [4] accept arbitrarily-sized images as input to the convolutional layers of the proposed CNN architecture. Then, a Spatial Pyramid Pooling (SPP) Layer [229] is stacked between the convolution layers and the fully connected (FC) layers. This SPP layer allows CNNs to produce a constant output size of the feature maps which is essential for training the upcoming FC layers. Similar techniques of pooling image features of arbitrary size into standard-size vectorial representations can also be found in [100, 165]. On the other hand, input images are typically resized to a fixed-size before being fed to the convolutional part of the network, to ensure a constant-size output from the concatenation of the feature maps as a final representation [104, 166]. Most methods argue that the former approach is more suitable since image resizing during preprocessing might miss important

information.

Finally, of note is the NN-based model proposed in [63]. In this work, a triplet convolutional neural network accepts pairs of word images as inputs and returns a similarity score in the output. Image description is not explicitly expressed as neither a variable nor a fixed-length vector. Hence, there is no image descriptor in the classical sense, and images are processed and represented internally throughout the NN layer pipeline. The output similarities guide the learning process by adjusting the final fixed-length representations in a word embedding space of images and strings, with the aim of minimizing the distance between similar images and maximizing the distance between different images.

2.4.3 Matching process

The matching task is composed of the similarity computation between the feature representations of the query, which may be a feature vector, a graph, or a statistical model and the document image at word, line or page level. The system performance is greatly affected by the suitable selection of the matching technique. Actually, an improper choice of a matching algorithm may lead to lower performance despite the potentially good choices of features and representations for a particular case.

Word to word matching

This family of approaches requires the document images to be segmented at word level and the matching is carried out directly between the representations of the query and each word image. Apart from the query type (template image or string), we can further distinguish *learning-free* from *learning-based* techniques. We should note here that *learning-based* methods typically rely on annotated dataset images for training the respective models. However, in the case that no labelled data (annotations of the transcribed text of the underlying images) are required and the models are trained in an unsupervised fashion, a further distinction, that was recently suggested in [173,174] discriminates *annotation-free* KWS methods from *annotation-based* approaches requiring fully or weakly supervised data [159].

Many of the proposed methods follow the learning-free paradigm under the QBE scenario. For instance, Rath and Manmatha [62] compare variable-length sequences of features extracted from word profiles using DTW for word spotting in historical

handwritten documents. In the same direction, many variants of DTW-based word spotting methods have been proposed. Adamek et al. [260] employ DTW to align convexity and concavity features extracted from single closed contours for spotting words in historical handwritten documents. In historical printed text, Khurshid et al. [189] propose an approach to initially align features (S-characters) extracted from connected components at character level by DTW and subsequently compare the resulting character prototypes at word level using a segmentation-driven edit distance. Rodríguez-Serrano and Perronnin [233] confirm the superiority of local gradient histogram features over the word profiles for multi-writer handwritten word spotting using DTW. Papandreou et al. [128] propose an adaptive zoning description that can also be matched by DTW for printed documents. An interesting unsupervised example-based approach is proposed by Wicht et al. [151], where the authors make use of stacked convolutional deep belief networks (CDBN) for extracting features from image patches and DTW as the matching technique for the variable-length sequences of deep features. Retsinas et al. [163] propose an efficient word-based and learning-free approach for handwritten KWS based on a modified version of their previously employed projections of oriented gradients [67], which is combined with horizontal zoning. A selective sequence matching algorithm is used to determine the similarity between sequences of descriptors. The proposed representation achieved the highest performance with respect to the state of the art at the time it was published, both for historical and modern benchmark datasets under multi-writer conditions. The robustness to multiple writing styles is partly attributed to a substantial preprocessing step of the images and is further improved by augmenting the query image set under different preprocessing configurations. Multiple query instances are then retrieved by an extended multi-instance selective matching technique.

Fixed-length representations are also very common in the QBE learning-free case. Gatos et al. [238] introduce the idea of adaptive zoning features for QBE word spotting in a historical printed document dataset. These features are extracted after adjusting the position of every zone based on local pattern information. The adjustment is performed by moving every zone towards the pattern body according to the maximization of the local pixel density around each zone. In the same dataset, a size-normalization technique along with zoning and profile features to compute the dissimilarity between two word images is proposed in [261]. The distance is based on a combination of a windowed Hausdorff measure and a robust curvature estimation

using integral invariants. Another learning-free fixed-length representation which is based on zoning characteristics is proposed in [132] and uses the L_1 distance metric. Moreover, characteristic Loci features [262], which are a particular case of the shape context descriptor, have been used by Fernandez et al. [239]. They are extracted from keypoints, represented by histograms of Locu numbers in a fixed-length vector and compared using the Euclidean distance. As it was mentioned in the previous paragraph, Retsinas et al. [67], combine projections of image gradients in a Radon transform-like procedure to form fixed-length vectors which are compared using the Euclidean distance.

Aldavert et al. [51] propose an unsupervised QBE method based on the BoVW framework which is enhanced by several improvements recently proposed in computer vision, though not exploited by the document analysis community. Particularly, they encode descriptors using the sparse coding technique proposed in [263], known as Locality-constrained Linear Coding (LLC). To make visual words more discriminative, they add spatial information using a Spatial Pyramid Matching (SPM) [255] as well as a power normalization technique during the pooling process. The query fixed-length vector is then matched with the vectors of the dataset word images using the Euclidean distance. Unsupervised word spotting methods based on the Bag-of-Visual-Words paradigm can attain a high retrieval performance when the methods used at each step are selected carefully. Although a representative codebook can be created on small datasets without high cost, this is not the case for datasets where millions of words are written by multiple writers. A straightforward solution is to randomly sample a subset of word snippets to generate the codebook. However, this approach has the drawback that certain characters and writing styles may be under-represented by the codebook. Therefore, Aldavert et al. [52] propose a codebook trained from synthetic data which incorporates semantic information in the generation process to automatically determine the optimal size and cardinality of the codewords. The use of synthetic data ensures that all characters are properly represented while at the same time allowing to simulate the script variability present in documents written by multiple writers.

A combination of four novel graph-based representations for QBE learning-free KWS is proposed by Stauffer et al. [161]. A first approach is based on the representation of characteristic keypoints by nodes, while edges represent strokes between these points. Another approach is based on a grid-wise segmentation of word images,

where each segment is eventually represented by a node. The next two representation configurations are based on vertical and horizontal segmentations of word images by means of projection profiles. For matching between the query graph and the dataset word image graphs the authors utilize an approximation of inexact graph matching based on the bipartite graph edit distance algorithm. This work is an extension of their former approach proposed in [158].

With respect to the learning-free paradigm, there are also methods which allow textual queries as inputs (QBS). Bhardwaj et al. [215] extract high order geometrical moments from binary word images as global features to form fixed-length feature vectors which are compared using the cosine similarity. A template is used to generate the query word image corresponding to the query text inserted as input by the user. Another type of techniques that falls in this category relies on representations using word shape coding (WSC). Image matching is usually performed among code strings by means of the minimum *Edit Distance* or by some sequence alignment procedure. The Edit Distance between two strings is given by the minimum number of operations needed to transform one string to the other, where the operation is an insertion, a deletion, or a substitution of a single character. For example, Bai et al. [125] extract features such as character ascenders, descenders, deep eastward and westward concavity, holes, i-dot connectors and horizontal-line intersection. These features are represented using word shape coding and the resulting vectors are compared by a sequence alignment technique. The main advantage of WSC approaches is that arbitrary textual queries can be used without involving training on labeled images. However, such approaches have become obsolete over the years as they are language dependent and feasible mainly in printed documents with already known fonts.

A probabilistic representation for learning-based QBE word spotting in multi-writer text is proposed in [22]. The query and the dataset word images are represented as sequences of feature vectors extracted using a sliding window in the writing direction and they are modeled using statistical models. Particularly, the authors use multiple instances of a potential query for training a HMM. During the matching process, the similarity between the query and a word image is obtained by the posterior probability of the candidate image, given the model. This probability is calculated using either a continuous HMM (c-HMM) or a semi-continuous HMM (sc-HMM) and a GMM as a universal vocabulary for score normalization. Among different types of fea-

tures, they report the best performance when local gradient histogram features (LGH) are chosen. However, the method is constrained to queries for which at least one instance appears in the training set. This issue is tackled in their extended work [99] for spotting out-of-vocabulary (OOV) words using a sc-HMM. In fact, the model's parameters are estimated on a pool of unsupervised samples which allow the model to adapt online to the query image. Moreover, the similarity computation between two sc-HMMs is simplified to a DTW between their Gaussian mixture weight vectors which reduces the computational cost.

A first attempt of QBS language-independent KWS for different indic scripts is proposed in [33], where a character zone-wise segmentation splits the word image into three zones, namely a middle zone and upper/lower zones, corresponding to the main part of a word and its modifiers, respectively. Pyramids of Histogram of Gradients (PHOG) are used as features to describe these zones from each word image. The middle zone is modeled using HMM whereas the upper/lower zones are used to train a support vector machine (SVM) classifier of modifiers of the middle-zone word. The HMM is trained only from source script characters and tested in the target script word image, in a sliding-window based fashion, where for each position of the image PHOG features are extracted. The probability of the character model of the text line (line transcription used as training data from source script) is maximized by Baum-Welch algorithm assuming an initial output and transitional probabilities, while a character filler model [71] represents isolated character models. At the recognition stage of the HMM, the trained characters HMMs are connected to the keyword text model in order to calculate the likelihood score of the input keyword image. This likelihood score is finally normalized with respect to a general filler model before it is compared to a threshold for final ranking. The output of the SVM model for the upper/lower zones is then combined with the HMM-computed word score in a re-ranking step to improve the final KWS performance.

It is interesting to notice that there exist learning-based methods which can deal with both types of query formulation (QBE and QBS). In the seminal work of Almazán et al. [31, 236], the authors have proposed a model to learn projections from an image space and a text-string space to a common latent subspace using Kernel Common Subspace Regression (CSR). Vectors in the latent subspace correspond to a common fixed-length representation, computable both for word images and text strings. Dense SIFT descriptors are extracted from word images, encoded to

Fisher vectors, while their labels are used to create Pyramidal Histogram of Characters (PHOC) descriptors. PHOC encode textual information in the form of a spatial pyramid of character histograms, treating the absence or presence of unigrams and bigrams as text attributes. At test time, dataset word images and the query are projected to a Euclidean common latent subspace and compared using nearest neighbour search. A number of recent works have been inspired by the work of Almazán and co-workers, further extending or adapting the base model [4, 32, 43, 69]. In Sudholt et al. [4], PHOC descriptors are computed by a sigmoid activation function at the last layer of an end-to-end CNN, dubbed *PHOCNet*, while in Krishnan et al. [32] deep CNN penultimate layer activations are used to create word image descriptions from an end-to-end network (HWNet) which is initially pre-trained on a large dataset of synthetically generated images (HW-SYNTH) [144]. In Wilkinson et al. [69], a triplet CNN is used, accepting pairs of positive word matches plus a negative pair. Therein, a new text descriptor is also proposed, dubbed Discrete Cosine Transform of Words (DCToW). A similar in spirit work of Aldavert et al. [50] combines visual (SIFT) and textual information obtained from character n-gram models to allow example-based or textual queries. Word images are represented by fixed-length vectors and matched using the cosine similarity.

Since 2016, there has been much interest in using neural networks for keyword spotting. Concerning deep learning-based methods, when dealing with word image description and word to word matching, convolutional neural networks (CNN) or similar feed-forward networks that include convolutional layers in their architecture have been used. These networks work typically either by producing in their output a suitable descriptor of the input word image [4, 100, 101, 159, 173, 174], or by using network layer activations to create input word image descriptors [32, 34, 37, 74, 120, 153, 180, 235]. Again, a typical distance that is used is the Euclidean. In the case of Sudholt et al. [4] the Bray-Curtis dissimilarity is employed since it is a metric that has been shown to work well with spatial pyramid representations [264]. In Zhong et al. [63], a neural network that accepts pairs of word images has been proposed. This model directly outputs similarity scores for the input pair. The pairwise similarity comparison of two word images is considered as a classification as well as a regression problem by collecting a dataset with positive and negative word image pairs using supervised learning. In order to deal with the fact that neural networks require a comparably large training set, the technique of *jittering* or data augmentation has

been used to augment training sets. Following this technique, a number of simple affine transformations can be applied on the training word images to create new training images and boost NN performance [4, 32, 69]. Pretraining on a generic set and then refining the network on a different second set, which typically should be qualitatively closer to the test set, is another standard practice [32, 69]. In Sfikas et al. [235], only pretraining with a generic set is performed, skipping refining altogether. They proceed using combinations of intermediate layer activations as features, aiming to capture more abstract textual features in this manner.

In recently proposed seminal works that employ NNs to compute a standard attribute-based descriptor enabling both QBE and QBS KWS, Sudholt et al. [100] further explore the effect of the word image embedding, adopting three different representations. Apart from PHOC, DCToW is employed, as well as a variation of PHOC, dubbed Spatial Pyramid of Characters (SPOC), which can be seen as a multinomial generalization of the PHOC. Hence, instead of expressing binary presence or absence of each character in each split, the corresponding characters are counted. The authors also improve the KWS performance by adopting a Temporal Pyramid Pooling layer (TPP) as a modification of their former Spatial Pyramid Pooling (SPP) layer [4] to accept input images of variable size. The TPP layer subdivides its input feature maps coming from the last convolutional layer, on horizontal bins only, contrary to horizontal and vertical bin splits of the SPP. This way it encodes features at each spatial position of the word in the writing direction, following the sequential nature of handwritten inputs of arbitrary size. In addition, the KWS performance is evaluated under different loss functions for NN training (cosine and binary cross entropy loss). Cosine distance is also used for final ranking. Gurjar et al. [159] were the first to introduce a weakly supervised KWS method, in an attempt to drastically reduce the requirement of having thousands of training images manually transcribed, to only requiring a few hundred image annotations, without significantly affecting the KWS performance. The proposed method utilizes the *HW-SYNTH/IIIT-HWS* synthetic dataset [32, 144] to pre-train the PHOCNet CNN ([4]), which is then fine-tuned in two target benchmark datasets. KWS performance is on par with the state-of-the-art, only using 86% and 98% less training data from each target dataset, respectively. Al-Rawi et al. [101] introduced a multi-script PHOC word image representation learnt from a ResNet-152 deep CNN [265] for script-independent word spotting on English, French, German, Arabic and Bangla (Indian) languages. In this work, a unified multi-PHOC repre-

sentation for multiple scripts is built by concatenating all symbols from all examined languages into a single set of symbols. This way, a single model can learn words from any script. In the same spirit with Gurjar et al [159], Wolf et al. [173,174] propose a completely annotation-free KWS system. To this end, they make use of two initial (ResNet50) CNNs to classify the font (each font used for synthetic data is a class) and the slant angle (among five classes) of synthetically generated word images [165]. These networks also predict the font and slant for each word image of the target dataset so as to adapt the synthesis process to unknown samples of a collection without any annotations. The TPP variation of the PHOCNet [100] is employed for predicting candidate pseudo-labels for the target dataset, as a word recognition task, provided a lexicon for each target language. In [174], a confidence measure is also proposed to further assess this word recognition process for label creation. These pseudo-labels are then used to fine-tune the TPP-PHOCNet for final ranking.

In the case where intermediate layer activations are used to describe word images, Gomez et al. [153] first learn a string embedding space in which distances between projected points are correlated with the Levenshtein edit distance between the original strings. This means that in this embedding space, the Euclidean distance between two points is equivalent to the Levenshtein distance of the words they represent. The proposed network is trained with a siamese setup, presented with arbitrary pairs of text strings. Once the string embedding model is trained it is used to teach an image embedding model [266] so that, given a word image as input, it regresses at its output the LSDE representation of the corresponding string provided by the teacher model. Finally, the authors construct a deep image-string embedding model by jointly fine-tuning the pre-trained image and string models. LSDE image representation is compared against PHOC and DCToW and shows superior performance when correlated to the edit distance, in contrast with any other nearest neighbor search. Retsinas et al. [180] suggest a compact deep descriptor based on the max-pooled outputs of the segmented (into horizontal zones) feature maps from the last convolutional layer of the proposed PHOCNet-based network for QBE KWS. The proposed CNN is used for predicting unigram and bi-gram models where unlike PHOC, their spatial information is encoded to the final fixed-length representation from the pooling combination of the multiple convolutional outputs of each zone. Krishnan et al. [34] utilize an extended version of the HWNet system [144], dubbed *HWNet v2* [165], which is based on a multi-scale training of multiple ResNet34 network architectures for image

representation of real data. Specifically, ResNet34 network includes four blocks where each block contains multiple ResNet modules, an ROI pooling layer to pool the visual representation from the penultimate layer of each ResNet, to a fixed-size embedding, and two fully connected networks. Apart from the real stream, the proposed system allows for textual embeddings which are fed by a label stream containing a synthetic image and its PHOC string embedding. The synthetic image representation and the PHOC attributes are projected to a common subspace using CSR, to capture the correlation between the attributes present in both modalities. This way the proposed method enables both word spotting and recognition in a novel end2end embedding framework which learns a common subspace of image and text representation in a multi-task learning fashion. Final attribute representations are matched using cosine distance achieving seminal KWS performance in well-established benchmark datasets. Finally, Retsinas et al. [37] propose a unified model that can handle both KWS and word recognition with the same network architecture. The network is comprised of a non-recurrent CTC branch and a sequence-to-sequence (Seq2Seq) branch [120] which is used to create efficient word representations. The deep features are further augmented with an autoencoding module which translates query strings to the Seq2Seq intermediate representation space, or by forced aligning the query to the decoder. The proposed joint loss leads to a boost in recognition performance whereas intermediate Seq2Seq-based representations are further refined with binarization, using an efficient straight-through estimator-based (STE) retraining scheme [267] to provide compact and highly efficient descriptors for KWS. Their method is on par with the state of the art for QBE KWS in multi-writer benchmarks, while it sets a new baseline performance for QBS KWS.

Word to line matching

This family of methods requires the documents to be segmented at text lines. Normally, a window slides over the text lines in order to extract column-based features. We can distinguish two main types of approaches.

In the first category, there are learning-free QBE methods that represent the query and the text lines with sequences of feature vectors and word spotting is applied as a subsequence matching task. In this framework, Terasawa and Tanaka [103] extract Slit Style HoG features from the query image and the text lines using a sliding window. These features are a modification of HoG which is based on gradient dis-

tribution. Variable length sequences representing the query and the text lines are then matched using a DTW-based technique which uses Continuous Dynamic Programming (CDP). CDP computes similarities between the query sequence and all the possible subsequences of a text line. Similarly, Mondal et al. [97] make use of word profiles and propose a flexible sequence matching technique which is based on DTW and has the ability to find subsequences in a sliding window-oriented approach, permits one-to-many and many-to-one correspondences while at the same time skipping outliers.

The second category is mainly composed of learning-based QBS methods. Therein, representations of features extracted via a sliding window are modeled using statistical models, such as HMMs [71, 227, 250] and recurrent neural networks [29, 249].

For instance, a HMM-based method which learns character models for word spotting in handwritten text is proposed in [227]. Initially, text line images are normalized to reduce variability in writing style. Each text line image is represented by a sequence of feature vectors which is obtained by a sliding window of one pixel width moving from left to right over the image. At each window position, 9 geometrical (Section 2.4.1) features are extracted. A character HMM has a standard number of states, each emitting observable feature vectors with output probability distributions given by a GMM. Character models are trained offline using labeled text line images. Then, a text line model is created as a sequence of letter models according to the transcription. The probability of this text line model to emit the observed feature vector sequence of the line image is maximized by iteratively adapting the initial output probability distributions and the transition probabilities between states with the Baum-Welch algorithm [268].

A HMM-Filler model which can generate any sequence of characters is created using all trained letter HMMs. For a given text line image which is modeled by the Filler model, the likelihood of the observed feature vector sequence is computed using the Viterbi algorithm [268]. This way the Filler model can be used once to compute offline the Viterbi decoding for all given text line images. In the online phase, a textual query is represented by a keyword model which is build from character and Filler HMMs. A Viterbi score is also computed for this keyword model and a given text line image and the final matching score between the query and the specific text line is a likelihood ratio of the keyword and Filler text line models, normalized by the length of the query word. This work is improved in [71] by integrating character n-gram

language models into the spotting task.

An important drawback of this approach is the large computational cost of the keyword-specific HMM Viterbi decoding process needed to obtain the confidence scores of each word to be spotted. To counter this issue, Toselli et al. [250] propose a technique to compute such confidence scores, directly from character lattices produced during a single Viterbi decoding process using only the Filler model, meaning that no explicit keyword-specific decoding is needed.

Another learning-based QBS method which makes use of the same features, the same representation and employs bidirectional long-short term memory (BLSTM) recurrent NNs is presented in [29]. The input layer contains one node for each of the 9 geometrical features extracted at each position of the sliding window, the hidden layer consists of the long short-term memory (LSTM) cells and the output layer contains one node for each possible character along with a special node to indicate "no character". The output activation of the nodes in the output layer are normalized to sum up to 1. This way they can be seen as a vector indicating the probability for each letter to occur at a particular position. The output of the network is therefore a matrix of probabilities for each letter and each position. A score is assigned to each path through the matrix by multiplying all probability values along the path. The letters visited along the optimal path, meaning, the one with maximum score, give the spotted letter sequence. To spot a query keyword inside a text line, the character probability sequence is extended by an additional entry with a constant value of 1. By adding this symbol at the beginning and at the end of the keyword, the algorithm finds the best path through the output matrix that passes through the symbol added at the beginning, then through all the characters of the keyword, and then through the symbol added at the end. In other words, the path traverses through the letters of the keyword at their most likely position while the rest of the text line has no influence. This way, the keyword spotting score reflects the product of all character probabilities of the optimal subsequence that starts with the space before the first character of the keyword and ends with the space after its last character. This score is also normalized by the length of the query word.

With the advent of the deep learning era, a number of methods have been also proposed for line-level KWS. Retsinas et al. [178] propose a novel method for word to line KWS which can perform both QBS and QBE paradigms. This is achieved by three basic components. A character width estimator NN which serves as a scale

detector in order to assert scale invariance. A feature extraction model based on the TPP-PHOCNet architecture [100] which extracts CNN features from the convolutional output of the last convolutional layer using ROI pooling for text lines and finally, a common space encoder NN which projects both visual features and attributes into a common space, enabling both QbE and QbS scenarios. With respect to line image normalization according to horizontal zones, deep features are extracted and fast compared since instances of the same word have similar width at the normalized line images. As for the convolutional layers, they are not constrained by the size of the input image and therefore the keyword search is addressed as a feature matching procedure between query and text-line feature representations. In [89] Cheikhrouhou and colleagues propose a unified CNN-BLSTM deep neural network architecture where script identification and line-level QBS keyword spotting tasks are jointly trained through a multi-task learning framework. The script identification result is injected in the KWS module to eliminate characters of irrelevant scripts and perform the decoding step using a single-script mode. While traditional HTR-based decoding methods use the HMM or the CTC token passing for keyword detection, the authors rely on a generalized decoding technique, based on Weighted Finite State Transducers (WFST) representation [269]. In short, a CTC-based network produces the probabilities of the characters with an extra label known as blank character. Then, the sequence of the probabilities is decoded using the CTC algorithm to produce the label sequence [11]. This WFST representation allows CTC decoding to be combined to a grammar that encodes the keyword line model. The grammar encodes the permissible character sequences that enable the detection of the keyword at the beginning, the middle or the end of a text line. Non-keyword characters are absorbed by a filler model [71], which is a background graph concatenating all the character models.

Word to page matching

One of the major issues of the preprocessing stage is that possible segmentation errors are regularly conveyed in the spotting phase. Particularly, accurate word segmentations are difficult to obtain in handwritten and degraded documents. For this reason, several *segmentation-free* word spotting techniques have emerged.

Leydier et al. [102, 196] compute local keypoints over a document page in order to detect regions of interest. Gradient features are then extracted from these zones of interest and the query image. The user actually inserts a textual query which is

artificially generated from manually selected character images. The query image feature vector is then matched with that of each zone using an elastic matching method between different pixel-wise gradient matchings. In a similar fashion, Zhang and et al. [55, 198] detect regions of interest by computing local keypoints over the document pages. Features based on the Heat Kernel Signature (HKS) [270] are extracted from these regions and used through a costly distance computation in a language independent manner, though not scalable in large datasets.

The most common approach is to use a patch-based framework [58, 61, 96, 116, 129] in which a window slides over the whole document. In this framework, perfect segmentations are not expected and elements from surrounding words will appear within a patch. Gatos et al. [129] detect salient text regions on a document page using a RLSA-based smoothing. A block-based extraction of pixel densities is then applied for the query image and the salient regions which are matched using a template matching process satisfying invariance in terms of translation, rotation and scaling. Rusiñol et al. [116] represent document regions with a fixed-length descriptor based on the BoW representation of SIFT features extracted via a sliding window over the whole page. In this case, comparison of regions is much faster since a dot-product or Euclidean distance can be used. In addition, Latent Semantic Indexing (LSI) is used to learn a latent space where the distance between word representations is more meaningful than that in the original space. Rothacker et al. [58] also make use of the BoVW to feed a HMM obtaining a robust representation of the query in a patch-based framework. The HMM is trained on-the-fly from the specific query.

Hast and Fornés also follow a combination of keypoint detectors using SIFT/SURF for blobs, Harris corner detector and an edge detector to form a variable length feature vector for learning-free QBE KWS. For each keypoint, a local disc neighbourhood (with radius 16) is sampled into a square matrix, from which the amplitude of the Fast Fourier Transform (FFT) is computed. Image matching is reduced to a nearest neighbor between these four subgroups of corresponding keypoints which are computed for the whole document and the query in a sliding window fashion. The resulting correspondences from this matching process between the query and the sliding window needs to be further processed to discard many false positives (outliers). This is done by a relaxed version of the RANSAC algorithm, namely, by a Putative Match Analysis (PUMA) technique [271].

However, when following a sliding-window approach there are too many possible

targets to consider, depending on the number of scales and the stride length. This leads to an increase in the number of false matches and the computational demands. To this end, Kovalchuk et al. [95] propose the extraction of a set of overlapping candidate targets as groups of connected components that satisfy location and scale constraints to fit a standard-size bounding box. Subsequently, they combine HoG and LBP features to form a fixed-size feature vector representing the query and the candidate target images which are matched using the Euclidean distance. In a recent deep learning-based QBE KWS technique, Ghosh and Valveny [73] detect candidate word regions from grouping connected components (CC) with horizontal co-line constraints. This is done by the generation of an over-complete set of line separation hypotheses by simply finding local minima in the horizontal projection of the image, after applying an average filter in order to smooth the projection profile. Each CC is assigned to one or more text lines and then candidate word regions are generated as combinations of CCs within the same line while rejecting non-word and non-text areas. These candidate word regions are fed to a region proposal CNN, similar in spirit with fast R-CNN [272], to aggregate deep features from different salient regions of the image using standard PHOCNet architecture. To predict PHOC vectors from different word candidates the SPP layer of PHOCNet [4] is replaced using a Region of Interest (ROI) pooling layer. Euclidean or cosine distance is then used for the retrieval of similar instances with the query image. Wilkinson et al. [70] propose a novel end-to-end CNN for segmentation-free QBS KWS, dubbed as *Ctrl-F-Net*. Dilated text proposals are extracted using a Region Proposal Network (RPN) [273] in order to predict bounding boxes. Furthermore, the predicted bounding boxes are augmented with a set of heuristically generated region proposals based on CCs with non-maximum suppression. Particularly, a CNN-ResNet34 is used for feature extraction, a RPN on 34 feature maps is used to regress 15 anchor boxes at each spatial position of a sliding window. Finally, 256 positive and negative sampled boxes are fed to a triplet CNN for extracting the final string embedding vector. These regions of interest are then represented with word string embeddings (PHOC/DCToW) in an integrated manner and retrieved according to cosine distance with the query.

Usually, CC-based methods for text detection or segmentation into word regions present two important drawbacks. First, the detectors are dependent on document image binarization. Secondly, since connected components can represent parts of words, single words or multiple words, heuristic strategies for combining CCs are required.

Rothacker et al. [156] propose to generate word hypotheses based on higher-level feature representations that indicate word occurrences. Inspired by text detection in natural images [274], they predict text/non-text scores for certain document image regions. The uncertainty of these scores is then explicitly modeled with extremal regions (ERs) [275]. The ER approach generates hypotheses of word bounding boxes. For these, PHOCs are predicted using a TPP-PHOCNet [100]. This is essentially a Region-based CNN (R-CNN) [272] framework. After predicting the PHOC descriptors, word spotting can be performed through a nearest neighbor search. Generating the local text scores is the critical part of the proposed method. Hence, the authors consider three different approaches: SIFT contrast scores, local region classification scores generated with a CNN and local word region scores obtained with an extension of CNN class activation maps [276].

Zagoris et al. [154] propose an unsupervised connected component analysis on local features extracted from keypoints. More specifically, the detection of CCs center of gravity is used as keypoints. Connected componets are detected on each level of quantized orientations of thresholded gradient vectors of the input image on horizontal and vertical direction, and local gradient-based features are extracted from such keypoints. For image matching, an average Euclidean distance from all normalized Euclidean distances between corresponding keypoints of the query and the test word images is used, where no corresponding points with the query are ignored. The method performs on par with a number of learning-based KWS systems for the QBE segmentation-free scenario.

Almazán et al. [96] use HoG features to describe the query image and the document pages in a fixed grid using a sliding window. In order to speed up the sliding-window approach, both Almazán et al. [96] and Rusiñol et al. [197] make use of the product quantization method [277] to compress the descriptor size. In the same direction, Ghosh et al. [237] perform QBS word spotting by avoiding the costly computation of the attribute-based representation over a sliding-window at query-time, which is previously employed in [31] for segmentation-based word spotting. This is achieved by pre-computing an integral representation of the attributes at the cost of discrimination. Within the grid based KWS category, Rabaev et al. [148] propose a scale space pyramid representation where the input image is iteratively convolved with a Gaussian filter and the result is then downsampled by a factor of 2. At each pyramid level HoG features are extracted in a $8x8$ cell and fixed-length representa-

tions are formed by a X^2 kernel mapping function to a Hamming space as well as k-means clustering of HoG descriptors. Compression of the final descriptor is further carried out using product quantization whereas image matching is performed using Euclidean distance for final ranking.

Moreover, Riba et al. [84] employ a graph representation relying on a codebook of graphemes which are extracted from shape convexities upon the vectorial approximation of the skeleton graph. These graphemes are used as stable units of handwriting, along with their spatial relationships. Segmentation-free word spotting is achieved by localizing the query word graph as a subgraph of the entire graph representing the whole document. The image matching is performed using an approximate graph Edit Distance method based on a bipartite-graph matching [278] between the two graphs. This time-consuming graph matching is improved by a graph indexing approach that makes use of binary embeddings during preprocessing.

Of note is also the work of Zhao et al. [175] for learning-based QBS segmentation-free KWS. The proposed system utilizes a pre-trained ResNet50 CNN on ImageNet dataset [279] for feature exctraction as well as a novel Feature Pyramid Network (FPN) for multi-scale features. In a following step, feature fusion is performed by a FPN concatenation where a feature sharing mechanism is employed to pass fused features to a multi-task training network. The first task corresponds to pixel classification which predicts word-image pixels, whereas the second task regresses bounding boxes by predicting offsets of word-image pixels from bounding box boundaries. A final third task learns a mapping from the word area to the word string embedding space (DCToW). Cosine distance is used for final ranking.

2.5 Retrieval enhancement

In this section, we present a number of methods which are used to improve the retrieved results of a word spotting system in terms of incorporating the information of the ranked lists obtained from user queries. This is done either by involving the user to select positive query instances in a supervised process, or in a purely unsupervised manner.

2.5.1 Supervised relevance feedback

The ranked lists of the images which are most similar to the query usually contain many false positive instances. In order to improve the performance of content-based image retrieval systems, several boosting mechanisms have been proposed over the years. *Relevance feedback* is a common technique of this type of approaches. The idea is to examine the results that are initially returned from a given query and to use information about whether or not those results are relevant. This feedback about relevance allows to provide an enhanced result list in the subsequent iterations. Relevance feedback is also used in more general information retrieval applications such as multimedia retrieval (MMR) [280], aiming to refine the multimedia data representation. The proper extraction of semantic information from multimedia data sources is a challenging task since these sources include directly perceivable media such as audio, image and video, indirectly perceivable sources such as text, bio-signals as well as not perceivable sources like bio-information, stock prices, etc. Particularly for word spotting, we can distinguish two main families of approaches, namely, *supervised* and *unsupervised* methods.

In the case of supervised relevance feedback, also known as *explicit feedback*, the user provides relevance judgements using either a binary or a graded relevance system. Herein, we can further notice two more categories. On one hand, relevance feedback methods may follow the idea of *query reformulation*. Its goal is to search, given the relevance assessments, a new query point in the vector domain that is closer to the positive instances and farther to the negative ones than the original query point. On the other hand, *re-ranking* methods attempt to reorganize the initial ranked list by means of the relevance judgements, without casting any new query.

For instance, the works of Bhardwaj et al. [215] and Cao et al. [188] adopt the query reformulation idea to improve the retrieved results based on the widely-used Rocchio's formula [281]. At each relevance feedback iteration the Rocchio's algorithm reformulates the query feature vector by adjusting the values of its individual features according to the relevance information. In a similar way, Konidaris et al. [132] and Kesidis et al. [133] propose to include the user in the retrieval phase by selecting positive instances from the initial ranked list obtained from synthetic query strings. Since the initial results are based on a heterogeneous comparison between synthetic keywords and real images, the accuracy might not be adequate. Consequently, the

transition from synthetic to real data is made feasible by exploiting relevant judgements and use them to perform new queries thus leading to an increased performance. Of great interest is also the work of Rusiñol et al. [184] where relevance feedback is tested both under the query reformulation scenario and the re-ranking scheme. Particularly, Rocchio's method [281] and a related variant are compared with a relevance score [282] (re-ranking). This score is assigned for each word image of the initial ranked list as the ratio between the nearest relevant and the nearest non-relevant word images for this particular image. These relevance scores are then used to form the final ranked list.

Another interesting approach that could fall into the supervised feedback techniques for improving the retrieved results by pruning false positive matches from the final ranked list is presented by Wolf et al. [283]. The authors investigate four different metrics for quantifying the confidence of a CNN in its predictions for KWS. The first confidence measure is derived from a probabilistic retrieval model that is a probabilistic variant of the well-established TPP-PHOCNet [100] and by treating sigmoid activation as a pseudo-probability. This metric quantifies the quality of a binary PHOC attribute prediction of a word (which can be seen as a word posterior probability), by evaluating the probabilistic model w.r.t. the ground-truth attribute embedding. The second confidence measure is based on dropout technique in intermediate CNN layers during testing where the metric is defined as the average variance of the estimations for each attribute after multiple passes of selected sample training images. The third confidence measure makes use of a surrogate model as a task independent meta-classifier which predicts whether a sample comes from *In Distribution (ID)* or *Out of Distribution (OD)* domain. The last measure comes from a task-dependent meta-classifier which treats intermediate layer activations of the TPP-PHOCNet as deep features which are then concatenated and fed through a single neuron with sigmoid activation (again as a pseudo-probability). This way the method is able to determine for which part of a dataset the retrieval system gives reliable results and thus successfully prune false positives.

2.5.2 Unsupervised feedback and re-ranking

The obvious benefits of supervised relevance feedback lie on the fact that the user judgements are assigned for only a small portion of all possible candidate targets

of the query image inside the document collection. However, this manual process still remains costly and sometimes, even error-prone, i.e. for historical degraded and cursive documents where the visual information is not distinctive enough. This gives rise to unsupervised methods where it is more preferable to automatically select instances from the retrieved results. *Pseudo-relevance feedback* [284] is a characteristic example of this type of techniques. In this case, the top-N results from the ranked list are considered as relevant. Subsequently, an unsupervised re-ranking scheme is used on these top ranked results in order to select a number of elements from the reordered list. These elements are finally added into the query for *query expansion* to obtain a new improved ranked list. The process repeats iteratively until the desirable performance is reached.

Regarding the unsupervised re-ranking scheme, Almazán et al. [96] apply a second ranking step which considers only the best candidates retrieved by an initial efficient ranking step and uses more discriminative features encoded with the costly Fisher vector representation. Once the results retrieved by the sliding-window search are re-ranked using more informative features, a number of top-ranked window regions are used for query expansion. Then the expanded query set is used as the new positive samples of the query model. Although this set may also contain negative samples the accuracy seems to improve per each iteration. In the same spirit, Ghosh and Valveny [237] use a re-ranking step to compensate for the loss of accuracy accrued from an approximate solution of the powerful attribute-based representation in order to transit from segmentation-based to segmentation free word spotting. In other words, they use the top-N candidates from the ranked list given by the initial ranking obtained with the sliding window search and then re-rank them using the more discriminative original representation. Shekhar and Jawahar [284] follow a similar pseudo-relevance feedback paradigm. Therein, the top-N retrieved results are re-ranked according to a score which integrates information from SIFT descriptors and BoVW representation with spatial information, which was missing on the indexing stage. Concisely, the spatial pyramid is used to calibrate the score of each region of the word independently.

Vats and Fornés [168] propose a query expansion algorithm that is based on a sliding-window search performed on a list of words, where keypoint-based feature matching is performed that generates a list of words with a certain degree of confidence. The proposed query expansion aims at improving the word matching results

by re-ordering the final ranked list according to the confidence score. For this reason, the retrieved list of words are spread across two lists based on the confidence scores obtained from the matching algorithm. The first list consists of words with a confidence score higher than or equal to 0.7 while the second list consists of words with a confidence score lower than 0.7. Keypoint matching is performed on the list of words in the first list to validate the matching result. A local query expansion algorithm is then used where the list of words found on a page locally, are taken into account, instead of performing query expansion on the entire set of pages. Therefore, local query expansion is performed on the first page using a limited number of retrieved instances. This procedure is repeated for each page, and can be fully implemented in parallel. The result from each query expansion is accrued generating a new list of found words with a higher degree of confidence, thus generating a list of best candidate words, with significant improvement in the KWS performance.

2.5.3 Data fusion

Pseudo-relevance feedback methods may sometimes result into several ranked lists which need to be combined into a final ranked list. *Data fusion* methods in this respect, accept two or more ranked lists and merge them into a single ranked list thus providing a better effectiveness than any original ranked list. There are two main categories of data fusion techniques. Methodologies which use the similarity values from each ranked list in order to produce the final ranked list are known as *score-based*, while those which use the ranking information from each list in order to create the final ranking are defined as *rank-based*.

It is interesting to notice that the work of Rusiñol et al. [184] also proposes three different data fusion techniques. The idea is to deal with variability in writing style by casting multiple queries and combine the results. An *early fusion* method combines feature vectors accrued from different queries before the retrieval phase. This is done by averaging the query image descriptors and then normalizing by the L_2-norm. The second method is a *late fusion* score-based technique (CombMAX) which assigns to each word in the collection its maximum score across the different casted queries. The third fusion technique is a rank-based method, called Borda Count [285]. Herein, the top most image on each ranked list gets n votes, where n is the dataset size. Each subsequent rank gets one vote less than the previous rank. The final ranked list is

obtained by adding all the votes per image and re-sorting.

Louloudis et al. [126] also make use of three rank-based fusion methods in order to combine multiple lists obtained from different word spotting systems applied to the same query. Particularly, the authors consider the same preprocessing steps and matching algorithms and test two different feature types for the same query. This results into two different ranked lists. The first combination method (Rank Position) takes into account only the rank positions of the corresponding words. The second method is the Borda Count and the third method which seems to outperform the other two is a Minimum Ranking method. Therein, for each retrieved word the minimum rank position on all ranked lists is considered as the distance measure. In a similar spirit with rank-based fusion techniques, Retsinas et al. [179] make use of ensemble methods to combine multiple ranked lists obtained from differently trained CNNs for word-based QBE KWS. To this end, they compute 5 word spotting retrieval lists that correspond to the 5 convnets trained with different initializations. Each retrieval list consists of the distance of the query example to the database instances, ordered in terms of increasing distance. Subsequently, they construct a single retrieval list, where the position of each retrieved instance in the list is defined as the minimum over the corresponding distances found on the 5 separate retrieval lists. The authors show that this ensemble strategy even when using a relatively low number of models, can lead to significant accuracy improvement.

The authors in [118] present five scored-based and three rank-based fusion methods to merge multiple ranked lists obtained from each top-ranked instance on the initial ranking list. Since the similarity scores among separate ranked lists may differ both in range and distribution, they also suggest to normalize these scores using a number of score normalization techniques.

Finally, Bansal et al. [36] fuse the noisy output of a text recogniser [286] with a deep embeddings representation derived out of the entire word initially proposed in [34]. The authors then use average and max fusion to improve the ranked lists. More specifically, from the initial ranked list, they start with a word of zero edit distance to the query. A synthetic image corresponding to this word is then generated and subsequently fed to the real stream of the proposed end2end network [34]. That word's synthetic image embedding is then queried on the word image embeddings to obtain a re-ranked list. By using the complementary information of the text recogniser and word spotting methods, a word recognition and retrieval system is proposed,

capable of performing better than both of the individual systems.

2.6 Evaluation

The ranked list of results obtained from a word spotting system for a number of different queries is finally used to evaluate its accuracy. In this section, we introduce the databases which are publicly available and most widely used for word spotting. After describing the importance of having a common evaluation scheme for direct comparison between methods, we present the distinct measures used for assessing the performance. Finally, we present and discuss the results achieved by the state of the art in various word spotting applications. With respect to the results we have two sources of information. The first one contains results from three keyword spotting competitions, namely, H-KWS 2014 [287], KWS-2015 [288] and H-KWS 2016 [289], which were organized in conjunction with the ICFHR 2014, ICDAR 2015 and ICFHR 2016 conferences, respectively. The second source derives from the results reported by the recently published papers. We should note here that the initial survey article, published in 2017 [40], included results accrued from the first two competitions only for the state-of-the-art methods at that time. In this thesis, the work is extended with results obtained from the last competition organized for handwritten KWS, as well as the current trend established by the state of the art.

2.6.1 Databases

The *IAM*[1] database [290] consists of 1539 pages of modern handwritten English text, written by 657 writers. Pages are segmented and annotated, comprising 13353 text lines and 115320 words.

The *George Washington*[2] (GW) database [2] contains 20 pages of historical English text written by George Washington and his associates in 1755. The writing styles present only small variations and it can be considered a single-writer dataset. Pages are segmented and annotated, comprising 656 text lines and 4894 words. This is the most commonly used dataset for comparing different word spotting methods.

[1] http://www.iam.unibe.ch/fki/databases/iam-handwriting-database
[2] http://www.iam.unibe.ch/fki/databases/iam-historical-document-database

The *H-KWS 2014 Bentham* and *H-KWS 2014 Modern* datasets[3] were used in the H-KWS 2014 competition. In the recent literature, they are also dubbed as *Bentham14* and *Modern14*, respectively. The first one contains 50 pages from a document collection written by the English philosopher and reformer Jeremy Bentham (1748-1832) and his secretarial staff. It contains considerable variability in writing style and font size as well as noise. The second one is composed of 100 modern handwritten document pages written by 25 authors in four different languages (English, French, German and Greek).

The *KWS-2015 Bentham*[4] dataset contains 70 document pages containing 15419 segmented word images and was used in the KWS-2015 competition. Again, it is also mentioned as *Bentham15* dataset in some recent works.

The *Botany16* and *Konzilsprotokolle16* datasets[5] were used in the *H-KWS 2016* competition [289]. Both test datasets consists of 20 documents, written in English and German, respectively. *Botany16* is from the India Office Records and provided by the British Library. This collection covers the following topics: botanical gardens; botanical collecting; useful plants (economic and medicinal). *Alvermann Konzilsprotokolle* belongs to the University Archives Greifswald and involves around 18000 pages. This collection contains fair copies of the minutes, written during the formal meetings held by the central administration between 1794-1797. The documents belong to the University Archives and were digitized and provided by the University Library in Greifswald. Transcripts were provided by the University Archives (Dirk Alvermann).

Figure 2.2 shows an example page from each dataset mentioned above. To alleviate the process of cross-referencing the results among new word spotting methods and the ones considered in this work, Table 2.4 discriminates the proposed approaches for the aforementioned databases. We note here that to our knowledge, some of the submitted works in KWS-2015 and H-KWS 2016 competitions, are not necessarily published since the proposed KWS systems comply on several benchmark rules for each respective competition, and thus their KWS pipeline may vary from the initially published works. Therefore, we mention the respective groups and refer to the specific contest. As we will show in Section 2.6.3, apart from the competitions, we mainly focus on the results reported by the methods presenting the best comparison degree, in terms of the employed evaluation protocols and experimental setups.

[3] http://vc.ee.duth.gr/H-KWS2014/

[4] http://transcriptorium.eu/~icdar15kws/data.html

[5] https://www.prhlt.upv.es/contests/icfhr2016-kws/data.html

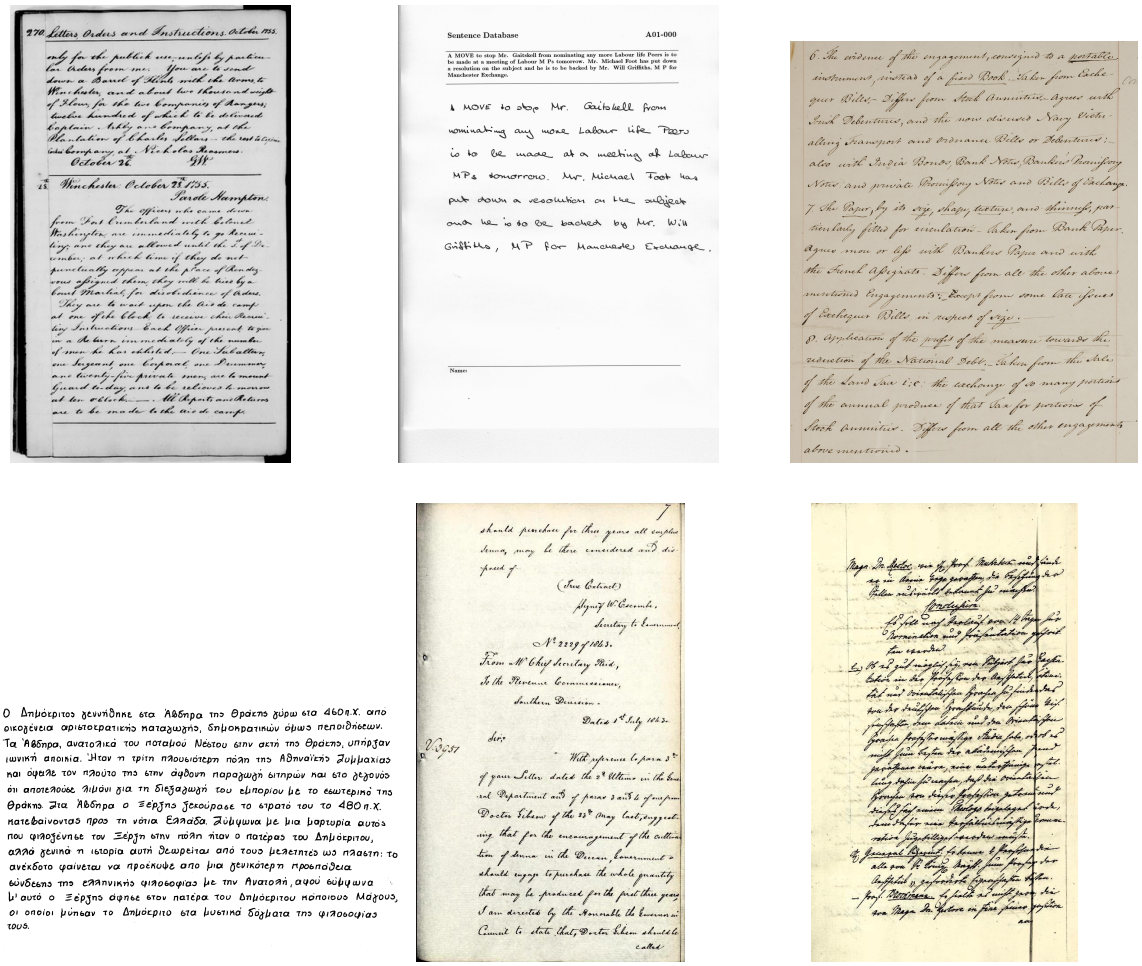


Figure 2.2: Example images of document pages from left to right for GW, IAM, Bentham (top row) and Modern14, Botany16, Konzilsprotokolle16 datasets (bottom row) considered in this work.

2.6.2 Evaluation protocols and measures

Many word spotting methods published in the recent years vary in assumptions and settings on which they depend. More specifically, some studies require the words to be segmented during preprocessing, while others require segmentation at line level or no segmentation at all. In addition, some methods are meant to perform well on a particular language, while others are able to deal with different languages and sometimes even heterogeneous scripts. There are also methods that target only printed text or specific writing styles, whereas others cope with handwriting variability. Moreover, some works rely on substantial prior learning using annotated data, while others are applied on unlabeled sets.

Apart from this wide variety of procedures and targets, there is also a huge dis-

Table 2.4: List of word spotting methods that use certain databases.

Databases	Methods
IAM	Bhardwaj et al. [215], Kumar et al. [65,109], Frinken et al. [29], Toselli et al. [66,250], Fischer et al. [71], Almazán et al. [31], Wshah et al. [110], Ghosh and Valveny [72,237], Sudholt et. al. [3,4,100], Wilkinson et al. [69,70,172], Krishnan et al. [32,34,144,165] Puigcerver et al. [83], Wicht et al. [150], Retsinas et al. [37,74,163,178–180], Rusakov et al. [160] Al-Rawi et al. [101,169], Mhiri et al. [166], Serdouk et al. [167], Wolf et al. [173,174] Jie et al. [170], Daraee et al. [81]
GW	Leydier et al. [196], Bhardwaj et al. [215], Rusiñol et al. [116,184,197], Lladós et al. [91], Rodríguez-Serrano and Perronnin [99], Frinken et al. [29], Almazán et al. [31,96,240,247], Liang et al. [68], Aldavert et al. [50–52], Howe [78,79], Dovgalecs et al. [61], Zhang et al. [55], Fischer et al. [71], Rothacker et al. [58,59,156,177], Kovalchuk et al. [95], Mondal et al. [97], Zagoris et al. [53,154], Wang et al. [92], Ghosh and Valveny [72,73,237] Sudholt et al. [3,4,100], Wilkinson et al. [69,70,172], Krishnan et al. [32,34,144,165] Zhong et al. [63], Bogacz et al. [27], Kulkarni et al. [147], Rabaev et al. [148], Wicht et al. [150,151] Wieprecht et al. [152], Gomez et al. [153], Retsinas et al. [74,155,163,180], Stauffer et al. [158,161] Bhunia et al. [33], Gurjar et al. [159], Rusakov et al. [160], Al-Rawi et al. [101] Mhiri et al. [166], Serdouk et al. [167], Jie et al. [170], Westphall et al. [171], Daraee et al. [81] Wolf et al. [80,173,174], Zhao et al. [175]
H-KWS 2014 Bentham	Kovalchuk et al. [95], Almazán et al. [236], Howe et al. [78], Leydier et al. [102], Pantke et al. [291], Aldavert et al. [51], Yao et al. [292], Santoro et al. [149,157] Retsinas et al. [155,163], Zagoris et al. [154], Hast and Vats [86], Wolf et al. [174] Amanatiadis et al. [176], Rothacker et al. [177]
H-KWS 2014 Modern	Kovalchuk et al. [95], Almazán et al. [236], Howe et al. [78], Leydier et al. [102], Pantke et al. [291], Aldavert et al. [51], Retsinas et al. [155,163]
KWS-2015 Bentham	Rothacker et al. (PRG) [288], Rusiñol et al. (CVC) [288], Leifert et al. (CITlab) [288], Sfikas et al. [235], En et al. [54], Santoro et al. [157], Zagoris et al. [154] Hast and Vats [86,168], Retsinas et al. [163], Benabdelaziz et al. [164], Wolf et al. [173,174] Rothacker et al. [177]
H-KWS 2016 Botany16 and Konzilsprotokolle16	Ghosh et al. (CVCDAG) [237,289], Sudholt et al. [4,100], Silberpfennig et al. (TAU) [289] Wilkinson et al. [69], Rothacker et al. [156,177], Krishan et al. [34,165], Stauffer et al. [161] Retsinas et al. [163], Westphal et al. [171], Daraee et al. [81], Wolf et al. [80], Vats et al. [168]

crepancy among methods that follow different evaluation protocols. This lack of homogeneity may lie on the distinct evaluation metrics, the sets of queries used for a specific dataset, the occurrence frequency of different queries, the number of pages or folds used for validation and testing for learning-based methods and others. The notable work of Rusiñol et al. [197] includes a review of the results obtained from various word spotting methods when tested on the English manuscript from the George Washington collection [2]. The inhomogeneity of these results somewhat confirms this discrepancy.

Consequently, one must take seriously into account the aforementioned aspects before evaluating a word spotting method, so as to make it directly comparable to as many approaches as possible. This way the results reported in the related literature will become more beneficial for new publications. Table 2.5 presents a clear view of the word spotting methods considered in this work with respect to the variable categories which are related to the evaluation issues mentioned above. Concisely, we consider the level at which segmentation is applied during preprocessing (word, line) and use the term "free" for methods that perform no segmentation at all. We then take into account whether annotated data is used for training or not. The variability of the

handwriting with respect to the number of authors is also taken into consideration (single author or multiple writers) except for printed documents. As we can see in Table 2.5, there are some distinct evaluation indices in the word spotting literature which are defined as follows. *Precision* is the fraction of retrieved words that are relevant to the query:

$$P = \frac{|\{\text{relevant instances}\} \cap \{\text{retrieved instances}\}|}{|\{\text{retrieved instances}\}|}$$

Recall is the fraction of relevant words that are successfully retrieved:

$$R = \frac{|\{\text{relevant instances}\} \cap \{\text{retrieved instances}\}|}{|\{\text{relevant instances}\}|}$$

The *F-measure* is defined as the harmonic mean of the precision and the recall:

$$F = 2 \cdot \frac{P \cdot R}{P + R}$$

The *R-Precision* index is defined as the Precision at a specific Recall value where $P = R$. In the case that precision should be determined for the k top retrieved words, $P@k$ is defined by:

$$P@k = \frac{|\{\text{relevant instances}\} \cap \{\text{k retrieved instances}\}|}{|\{\text{k retrieved instances}\}|}$$

This measure defines how successfully the methods produce relevant results to the first k positions of the ranked list. Finally, the *Average Precision* index (AP) is defined as the average of the precision value obtained after each relevant word is retrieved:

$$AP = \frac{\sum_{k=1}^{n} (P@k \times rel(k))}{|\{\text{relevant instances}\}|}$$

where $rel(k)$ is an indicator function equal to 1 if the word at rank k is relevant and 0 otherwise. The mean value of the Average Precision over all queries used in a word spotting task defines the *Mean Average Precision* (MAP). In Table 2.5 it is easy to observe that this index is the most dominant, thereby indicating its objectiveness and reliability.

In the case though where non-binary relevance assessments are provided beforehand, the Normalized Discounted Cumulative Gain (NDCG) index can be used in

order to handle small variations of the query word that can be found in the datasets. The NDCG measures the performance of a retrieval system based on the graded relevance of the retrieved entities. It varies from 0.0 to 1.0, with 1.0 representing the ideal ranking of the entities. For example, the words "fort" and "Fort" may have a relevance judgement equal to 0.9. It is defined by:

$$nDCG = \frac{DCG}{IDCG}$$

where:

$$DCG = rel_1 + \sum_{i=2}^{n} \frac{rel_i}{\log_2(i+1)}$$

where rel_i is the relevance judgement at position i, and $IDCG$ is the ideal DCG which is computed from the perfect retrieval result.

Finally, we would like to emphasize some crucial points of the performance evaluation of a word spotting system. As it is previously mentioned, the *relevance criterion* determines which query instances should be considered as retrieved and which of the retrieved as relevant. In the case of segmented words, the relevance criterion is a trivial choice as it states directly whether a retrieved word image is correctly classified as the word being searched for.

Actually, the larger the entity that is searched for occurrences of the query is, the less meaningful the relevance criterion becomes. In other words, when line-based methods are evaluated, this criterion only states if a retrieved line indeed contains the keyword, without any particular information of the relative location inside the line. Therefore, the evaluation measures could overestimate the performance. Not to mention that such a binary relevance assessment would yield a completely biased evaluation in a segmentation-free method where a retrieved word area would be considered as relevant if it just contained an actual instance in the document page. Due to this issue, the relevance criterion for segmentation-free word spotting systems should take into account the location information. A widely used measure in the literature considers the intersection over union (IoU) percentage between the retrieved word area and the ground-truth one. If this overlap ratio exceeds a specific threshold (usually 50%) the retrieved result is deemed as relevant. By these means, the system is evaluated in terms of how accurately the query instance is retrieved. We should also note here that in a segmentation-free method under the QBE scenario, the query itself should be taken into account in the final hit list, since it could be missing from

Table 2.5: Review of KWS methods for some of which, direct comparison is non-trivial according to the employed evaluation procedure.

Publications	Segmentation	Learning	Writing style	Evaluation index
[29, 71, 107–110, 250]	Line	Yes	Multi-writer	MAP
[55, 58, 95, 96, 116]	Free	No	Single	MAP
[56, 91, 92, 184, 231]	Word	No	Multi-writer	MAP
[53, 60, 64, 78, 182]	Word	No	Single	MAP
[125, 127, 132, 133]	Word	No	Printed	Precision/Recall
[84, 88, 197]	Free	No	Multi-writer	MAP
[24, 31, 99]	Word	Yes	Multi-writer	MAP
[136, 137, 242]	Word	Yes	Printed	MAP
[139, 189, 230]	Word	No	Printed	Precision/Recall, F-measure
[50, 79]	Word	Yes	Single	MAP
[61, 90]	Free	No	Single	Precision/Recall
[114, 239]	Word	No	Multi-writer	Precision/Recall
[75, 76]	Line	Yes	Multi-writer	Precision/Recall
[126, 128]	Word	No	Printed	Detection rate
[134, 185]	Character	Yes	Printed	Precision/Recall
[129]	Free	No	Printed	Precision/Recall, F-measure
[226]	Line	No	Multi-writer	MAP
[41]	Word	Yes	Multi-writer	F-measure
[219]	Word-part	No	Multi-writer	Detection rate
[97]	Line	No	Single	F-measure
[243]	Word	No	Single	Precision/Recall
[105]	Word	No	Single	Precision/Recall, F-measure
[68]	Word	Yes	Single	MAP at rank 10
[113]	Word	Yes	Multi-writer	Precision rate
[284]	Word	No	Printed	MAP
[117]	Word	Yes	Multi-writer	Precision/Recall, F-measure
[106]	Word-part	Yes	Multi-writer	Precision/Recall rates
[124]	Word	No	Printed	Mean Precision/Recall
[195]	Word	Yes	Single	Precision/Recall
[237]	Free	Yes	Multi-writer	MAP
[241]	Line	No	Printed	MAP
[59]	Free	Yes	Single	MAP
[118]	Word	No	Multi-writer	R-Precision
[102]	Free	No	Multi-writer	R-Precision
[153]	Word	Yes	Single	NDCG
[167]	Word	Yes	Single	Soft(edit distance) MAP

the retrieved regions.

2.6.3 Evaluation results

Regarding the first handwritten keyword spotting competition [287], an evaluation framework was established for assessing QBE keyword spotting approaches. The competition was divided in two distinct tracks. A segmentation-based track, where the location of word images inside the document pages was provided and a fully segmentation-free track. For each track, 50 document images of the H-KWS 2014 Bentham dataset and 100 document images of the H-KWS 2014 Modern dataset (25 pages per language) were used for testing at the competition, resulting in a total number of 300 document images for both tracks. The query set of each dataset contained word image queries of length greater than 6 letters appearing more than 5 times. The measures employed in the performance evaluation of the submitted word spotting algorithms are the Precision at Top 5 Retrieved words ($P@5$), the Mean Average Precision (MAP) and the Normalized Discounted Cumulative Gain (NDCG) for both binary and non-binary relevance judgements. In the segmentation-free track, an overlap percentage criterion was used to consider a retrieved result as relevant based on three overlap thresholds (0.6, 0.7, 0.8).

Five distinct research groups have participated in the competition with three methods for the segmentation-based track and four methods for the segmentation-free track. However, we present only the results achieved by the winners of each track. For more details about the methods participating and the results obtained in each case we refer the reader to [287]. The winner of the segmentation-based track is the learning-based method of Almazán et al. [236] which relies on the attribute representation of visual and textual features. We should note here that the authors adapt their system to the Bentham and the Modern benchmarks, by training attributes in the George Washington and the IAM datasets, respectively.

The winner of the segmentation-free track is the learning-free method of Kovalchuk et al. [95] based on the fixed-length representation of HoG and LBP features. The results obtained by these methods are presented in Table 2.6. The third row corresponds to the segmentation-based track, whereas the last row stands for the segmentation-free track. For the segmentation-free track we only present the results obtained on average for all the threshold values of the overlap percentage criterion.

Table 2.6: Experimental results achieved by the winners of each respective track.

Method	Bentham				Modern			
	P@5	MAP	NDCG (Binary)	NDCG	P@5	MAP	NDCG (Binary)	NDCG
Almazán et al. [236]	0.724	0.513	0.744	0.764	0.706	0.523	0.757	0.757
Kovalchuk et al. [95]	0.609	0.416	0.638	0.56	0.539	0.263	0.483	0.483

The second handwritten keyword spotting competition [288] was divided into two distinct tracks, namely, a learning-free (TRACK I) and a learning-based (TRACK II) track whereas each track included two optional assignments. A segmentation based assignment at word level and a segmentation-free assignment compose Track I. The training-based track was divided in QBE and QBS methods in a segmentation-free framework. Participants could submit to one or both of assignments, depending on the capabilities or restrictions of their systems. The evaluation set consisted of 70 document images from the KWS-2015 Bentham dataset, containing $15,419$ segmented words. The query set consists of 243 keywords of different lengths (6–15 characters). Each of these queries is represented by 6 or less different instances, comprising a total of 1421 query images. All queries occur at least 4 times in the evaluation set.

For each assignment, a baseline system was provided to the participants in order to compare their methods and tune the parameters of their systems, using a validation set of 10 document images, containing $3,234$ words. The query set for the validation partition included 95 images of 20 different keywords, extracted from the training page images as well. An additional set of 423 document images, manually segmented and transcribed into $11,144$ lines, was also handed to the participants competing in Track II as training data. No other training sets were allowed in this track.

Mean average precision (MAP) and $P@k$ were used to evaluate the solution of each participant corresponding to a particular assignment of each track. If a participant submitted solutions for both assignments, the MAP scores of each assignment were combined to produce a single ranking for each track. The combination rule was designed in order to favor participants with a flexible system without hampering those with a highly-specialized method. In segmentation-free scenarios, an overlap ratio of 0.7 between the retrieved area and the ground truth one was required to accept a result as a true positive.

Six research groups submitted final solutions to the evaluation system. Four of them participated in Track-I and the other two in Track-II. We will only present the results achieved by the winners of each track. To our knowledge, the proposed

methods of the winning systems were not published, since they might have been modified existing KWS systems to adapt to the specific needs of the competition. For this reason, we only mention the respective groups and refer the reader to [288] for more details about the baseline systems, the participant systems as well as the results achieved in each case.

The winner of the learning-free track was the Pattern Recognition Group (PRG - Leonard Rothacker, Sebastian Sudholt, Gernot A. Fink), from TU Dortmund University of Germany and submitted solutions for both assignments. The winner of the learning-based track was the Computational Intelligence Technology Lab. (CITlab - Gundram Leifert, Tobias Strauß, Tobias Grüning, Roger Labahn) from the University of Rostock, Germany who also submitted solutions in both assignments of Track II. Tables 2.7 and 2.8 illustrate the results for each case, respectively.

Table 2.7: Results for the winner of Track I.

Assignment	**Segm. based**		**Segm. free**	
Group	**MAP**	**P@5**	**MAP**	**P@5**
PRG	0.4244	0.4605	0.2761	0.3434

Table 2.8: Results for the winner of Track II.

Assignment	**QBS**		**QBE**	
Group	**MAP**	**P@5**	**MAP**	**P@5**
CITlab	0.8711	0.8737	0.8521	0.8552

The third and last to date, handwritten KWS competition was organized in conjunction with the ICFHR 2016 Conference, dubbed *H-KWS 2016* [289]. It was divided into two main tracks, namely, QBE (Track-I) and QBS (Track-II). Each track further discriminates word segmentation-based KWS systems (Challenge I.A, Challenge II.A) from purely segmentation-free approaches (Challenge I.B, Challenge II.B). Unlike previous contests, the aim of this competition was twofold. The first goal was to evaluate all the KWS methods using a unique evaluation protocol. In order to assess the performance of each system under different amounts of training data, the MAP obtained on each challenge was penalized depending on the amount of training data available at the time of the submission. Three distinct time periods (I:June 14–21, II:June 22–25, III:June 26–29) with increasing penalty order were used to this end. The second target was to compare the different participating methods with distributed performance on two different languages, i.e. historical German and English,

deriving from the *Botany16* ($D1$) and *Konzilsprotokolle16* ($D2$) datasets, respectively. The purpose of this distinction was to clearly understand the data requirements of each method and their concurrent applicability to different languages, without re-training the models for each language from scratch. Each test dataset comprised 20 pages wherein the bounding boxes of all words (3318 for Botany and 3891 for Konzilsprotokolle) were manually obtained. The query set of each dataset was provided in $UTF-8$ plain text format (QbS) and word image queries (QbE) of various length and frequency. 150 and 200 different words were manually selected for the Botany and the Konzilsprotokolle datasets, respectively. In the segmentation-free challenge of each track, an overlap ratio of 0.5 between the retrieved area and the ground truth one was required to accept a result as a true positive.

Following these rules, the score of a participant U in a given track was computed as follows. First, for each submission S, having access to the training data T (available in the corresponding period) on the challenge A, the average MAP over the two datasets ($D1$ and $D2$) was computed as follows:

$$MAP(U,A,T,S) \;=\; 0.5 \cdot MAP(U,A,T,S,D1) + 0.5 \cdot MAP(U,A,T,S,D2)$$

Then, the penalty factor $P(T)$ (corresponding values for time period I, II and III: $1, 1.5, 2$) for the training data T was applied to obtain a penalized MAP (PMAP):

$$PMAP(U,A,T,S) \;=\; \frac{MAP(U,A,T,S)}{P(T)}$$

Only the least penalized submission was considered for each challenge, e.g A, as the final score for the given user, U:

$$PMAP(U,A) \;=\; \max_{T,S} PMAP(U,A,T,S)$$

Finally, the score of the user in the given track combines the penalized MAP obtained in the two challenges as follows, in order to give extra credits to those teams that were able to participate in both challenges, without penalizing excessively those participants that decided to send solutions to only one of the two challenges in each track:

$$Score(U) \;=\; \max_{A} PMAP(U,A) + 0.2 \cdot \min_{A} PMAP(U,A)$$

Four teams participated in the track I (QBE) and three teams in the track II (QBS). Again, we will present the results achieved only by the winners of each track. The winner of the example-based methods of track I was the Computer Vision Center (CVCDAG), Universitat Autònoma de Barcelona, Spain (Suman Kumar Ghosh, Ernest Valveny, Marcal Rusinol) which submitted solutions to both I.A and I.B challenges. The proposed system is based on PHOC attributes [31] where for the segmentation-free challenge (I.B), a sliding window based approach similar to [237] was used. The winner of track II was the Pattern Recognition Group (PRG), TU Dortmund University, Germany (Leonard Rothacker, Sebastian Sudholt, Gernot A. Fink) which submitted solutions to both challenges. The system is actually the SPP-PHOCNet CNN proposed by Sudholt el al. [4] which employs sigmoid activation function at the final FC layer to predict the binary PHOC vector of the query and dataset input images. In word-based QBS formulation, the PHOC was directly extracted from query and test sets. For the segmentation-free case, PRG group use a sliding window over the document images to extract the PHOC for each patch and the CNN output for 6 patch sizes was pre-calculated by clustering the training word image sizes. For QBE, each query was then mapped to its closest pre-computed patch size and retrieval is performed with this size. For QBS, the training word image with minimal Bray-Curtis dissimilarity was used as the query PHOC for retrieval. Tables 2.9 and 2.10 illustrate the results for each case.

Table 2.9: Results for the winner of Track I (QBE)

	A. Segm. based			**B. Segm. free**			
Team	Botany	Konzil.	Average	Botany	Konzil.	Average	Final Score
CVCDAG	75.77	77.91	76.84	0.21	0.0	0.10	76.86

Table 2.10: Results for the winner of Track II (QBS)

	A. Segm. based			**B. Segm. free**			
Team	Botany	Konzil.	Average	Botany	Konzil.	Average	Final Score
PRG	36.47	76.93	56.70	11.80	48.41	30.10	62.72

In order to provide further insight of the state-of-the-art performance achieved in word spotting, we present the results reported by the recently published methods in the GW and IAM databases. Although these datasets are widely used, there exists no standard experimental setup and each work adapts it to the needs of their proposed algorithm. For instance, learning-based methods use cross validation and do not

evaluate the method on the same data used to fit their model. This reduces the amount of queries as query words must appear both in train and test folds. However, we choose these specific datasets since the reported results are comparable between various methods, at least at a certain degree. In this context, we review these results in Tables 2.11 and 2.12 by distinguishing various methods according to the query formulation, the segmentation level required, the use of learning with labeled data (i.e. number of training, validation and testing folds) and the employed experimental setup (i.e. query list). In each case, the MAP measure is used for performance assessment.

With respect to the GW database, Almazán et al. [31] partition the dataset into two sets at word level containing 75% (15 pages) and 25% (5 pages) of the words. The first set is used to learn the attributes representation and the calibration, as well as for validation purposes, whereas the second set is used for testing purposes. The experiments are repeated four times with different training and testing partitions and the results are averaged. In the QBE case, each word of the test set is used as a query in a leave-one-out style. Moreover, the query image is removed from the test set and queries without relevant occurrences are discarded. This setup is also used by Sudholt et al. [4, 100], Krishnan et al. [32, 165] and Wilkinson et al. [69]. In the QBS case, Almazán et al. [31] use only words that also appear in the training set as queries. This setup is also used by Fischer et al. [71, 227] and Frinken et al. [29]. In [71] though, punctuation marks are treated as individual words and they are excluded from the query list. This reduces the number of queries leading to an increased performance. Sudholt et al. [4, 100] use all words appearing more than once in the test set as queries. This is also followed by [32, 34, 69].

Rodríguez-Serrano and Perronnin [99] split the dataset uniformly into five folds, one for training, one for validating the parameters and three folds for testing their method. For each run, they compute the MAP of the test queries, using the best validation parameters. This process is repeated for all 20 different combinations of their setup and the results are averaged. Aldavert et al. [51] use as queries all dataset word images which appear at least 10 times and contain three or more characters. The query images themselves are also discarded from the retrieved results during evaluation. Kovalchuk et al. [95] employ the same setup as [51] for word-based word spotting and further perform segmentation-free word spotting. In the segmentation-free case, the query image is included in the retrieved areas when assessing the performance and a retrieved region is considered as relevant if it overlaps more than

Table 2.11: State-of-the-art performance for the GW database.

Reference	Query	Segmentation	Learning	Setup	MAP
Almazán et al. [31]	QBE	Word	4-fold cross validation: 2 training, 1 validation and 1 testing folds	All words in test set as queries	0.9290
Sudholt et al. [4]	QBE	Word	Same as [31] (SPP)	Same as [31]	0.9671
Sudholt et al. [100]	QBE	Word	Same as [31] (TPP)	Same as [31]	0.9796
Krishnan et al. [32]	QBE	Word	Same as [31]	Same as [31]	0.9440
Krishnan et al. [165]	QBE	Word	Same as [31] (HWNetv2)	Same as [31]	0.9824
Wilkinson et al. [69]	QBE	Word	Same as [31]	Same as [31]	0.9800
Rodríguez-Serrano and Perronnin [99]	QBE	Word	5-fold cross validation: 1 training, 1 validation and 3 testing folds	All words in training set as queries	0.5310
Aldavert et al. [51]	QBE	Word	N/A	All words with ≥ 10 occurrences and ≥ 3 letters as queries	0.7650
Kovalchuk et al. [95]	QBE	Word	N/A	Same as [51]	0.6630
Zagoris et al. [154]	QBE	Word	N/A	Same as [51]	0.6920
Retsinas et al. [163]	QBE	Word	N/A	Same as [51]	0.8110
Almazán et al. [31]	QBS	Word	4-fold cross validation: 2 training, 1 validation and 1 testing folds	All words in training set appearing in all 4 folds as queries	0.9390
Sudholt et al. [4]	QBS	Word	Same as [31] (SPP)	All words appearing more than once in the test set are used as queries	0.9260
Sudholt et al. [100]	QBS	Word	Same as [31] (TPP)		0.9792
Krishnan et al. [32]	QBS	Word	Same as [31] (QBS)	Same as [4] (QBS)	0.9280
Krishnan et al. [34]	QBS	Word	Same as [31] (QBS)	Same as [4] (QBS)	0.9898
Wilkinson et al. [69]	QBS	Word	Same as [31] (QBS)	Same as [4] (QBS)	0.9360
Fischer et al. [227]	QBS	Line	Same as [31] (QBS)	Same as [31] (QBS)	0.6000
Frinken et al. [29]	QBS	Line	Same as [31] (QBS)	Same as [31] (QBS)	0.8400
Fischer et al. [71]	QBS	Line	Same as [31] (QBS)	Same as [31] (QBS), excluding punctuation marks from query list	0.7380
Kovalchuk et al. [95]	QBE	Free	N/A	Same as [51], 50% overlap	0.5010
Rothacker et el. [58]	QBE	Free	N/A	All words as queries, 20% overlap	0.6110
Almazán et al. [96]	QBE	Free	N/A	All words as queries, 20% overlap	0.6880
Almazán et al. [96]	QBE	Free	N/A	All words as queries, 50% overlap	0.5910
Rusiñol et al. [197]	QBE	Free	N/A	All words as queries, 50% overlap	0.6130
Rothacker et al. [156]	QBS	Free	Same as [31] (QBS)	All unique transcriptions appearing on test set as queries (50% overlap)	0.8460
Wilkinson et al. [70]	QBS	Free	Same as [31] (QBS)	Same as [156] (50%)	0.9100
Zhao et al. [175]	QBS	Free	Same as [31] (QBS)	Same as [156] (50%)	0.9406

Table 2.12: State-of-the-art performance for the IAM database.

Reference	Query	Segmentation	Learning	Setup	MAP
Almazán et al. [31]	QBS	Word	3-fold cross validation: 1 training, 1 validation and 1 testing folds	All words in training set appearing in all 3 folds as queries	0.8060
Sudholt et al. [4]	QBS	Word	Same as [31] (SPP)	All words appearing more than once in the test set are used as queries	0.8290
Sudholt et al. [100]	QBS	Word	Same as [31] (TPP)		0.9342
Krishnan et al. [32]	QBS	Word	Same as [31]	Same as [4]	0.9150
Krishnan et al. [34]	QBS	Word	Same as [31] (S+DE)	Same as [4]	0.9509
Retsinas et al. [37]	QBS	Word	Same as [31] (WSRNet)	Same as [4]	0.9633
Wilkinson et al. [69]	QBS	Word	Same as [31]	Same as [4]	0.8940
Fischer et al. [227]	QBS	Line	Same as [31]	Same as [31]	0.3600
Fischer et al. [71]	QBS	Line	Same as [31]	Same as [31]	0.5500
Frinken et al. [29]	QBS	Line	Same as [31]	Same as [31]	0.7800
Retsinas et al. [178]	QBS	Line	Same as [31]	Same as [31]	0.7531
Wilkinson et al. [70]	QBS	Free	Same as [31] (QBS)	Same as [156] (50%)	0.8030

50% with the ground truth one. Similar protocol to [51] is followed by Zagoris et al. [154] and Retsinas et al. [163].

In the segmentation-free paradigm, Rothacker et al. [58], Almazán et al. [96] and Rusiñol et al. [197] use all word images as queries to retrieve candidate regions inside the document pages of the GW collection. The overlap percentage criterion used in [58,96] is set to 20%. In addition, Almazán et al. [96] also use a 50% overlap criterion in their reported results rendering their work directly comparable with [197]. For the QBS case in [70,156,175], all unique transcriptions appearing more than once used as queries for the test set and overlap criterion is set to 50%.

The experimental setup employed in the IAM benchmark is common for most of the reported results. There is an official partition for text line recognition which splits the pages into three different sets. The first one is used for training and contains $6,161$ lines, the validation set contains $1,840$ lines and the test set contains $1,861$ lines. These sets are writer independent, i.e., each writer contributed solely to one of the three sets. Although stop words are excluded from queries, they still appear in the dataset and act as distractors. The IAM dataset also contains a set of lines whose transcription is uncertain. These lines are excluded from training and testing. Only words that appear in the training set are used as queries. All non-stop words among the 4000 most frequent words that also occur in the training set are selected as queries as in [29], resulting in 2807 queries in total. Almazán et al. [31] retrieve whole lines that are correct if they contain the query word, so as to compare their approach with Fischer et al. [71, 227] and Frinken et al. [29]. To this end, all the words of

each line are grouped as a single entity and the distance between a query and a line is defined by the distance between the query and the closest word in the line. We should note here that the results reported by Fischer et al. [227] in Tables 2.11 and 2.12 are evaluated in [29] through a common experimental setup which allows direct comparison. Sudholt et al. [4, 100], Krishnan et al. [32, 34] and Wilkinson et al. [69,70] follow the same protocol as Almazán et al. [31] for training while at query time they use all words appearing more than once in the test set as queries. A minor modification is followed by Retsinas et al. [178] where, from all non-stop words among the 4000 most frequent words which also occur in the training set, the selected queries are more than those in [29], resulting in 3421 queries in total. We should note here that for most of the recently proposed learning-based methods, experimental results are also reported for the QBE paradigm. However, we prefer focusing only on the QBS case for the IAM dataset to render a slightly closer comparison among different methods.

2.6.4 Results discussion

Regarding the results presented in KWS competitions it is concluded that training-based methods can achieve much higher performance than training-free approaches which mostly rely on knowledge about geometric and structural properties of handwritten images without incorporating information obtained from the respective transcriptions. In that sense, training-based methods are the best choice if annotated data are available, to build efficient systems in terms of scalability and performance. However, training data obtained from documents written in a particular language, render the system's adaptivity dependent on a language written in a corresponding script. This can be also confirmed by the work of Almazán et al. [236] who perform training on GW and testing on Bentham. Segmentation-free word spotting methods should also be given attention since they still have much room for improvement and they are part of a relatively new research topic. Actually, approaches that bypass the segmentation step present a clear advantage in historical document collections where perfect word or line segmentations are hindered by various factors. Therefore, future competitions in this field should focus on such aspects to finally help understanding the relative capabilities and requirements of the different approaches to keyword spotting.

As for the performance achieved by the state-of-the-art methods presented in Tables 2.11 and 2.12, we can distinguish the top results per each distinct category for the GW and IAM benchmarks. We particularly consider the segmentation level as the main categorization factor between different approaches. In the GW dataset, the top MAP obtained under the QBE scenario, for word-based spotting using training data is reported by Krishnan et al. [165] (0.9824) slightly superseding Wilkinson et al. [69] (0.9800), which was the previous state-of-the-art method since 2016. The advantage of this method over other methods that do not rely on supervised learning is clear. In the same direction, though under the learning-free paradigm without using any labelled data at all, the results reported by Retsinas et al. [163] (0.8110) are quite promising and actually close to several deep-based KWS approaches. Among the QBS methods, the work of Sudholt et al. [100] (0.9792) which is the TPP-PHOCNet variation of [4] as well as the recently proposed HWNet $v2$ architecture of Krishnan et al. [165] achieve (0.9898), give excellent results that are numerically very close to one another. Also, they give superior numerical results compared to the line-oriented methods reported. Nonetheless, their approach requires the pages to be segmented at word level during training, which is not the case for the three line-oriented approaches. In the segmentation-free case and under the training-free and QBE paradigm, the best results are reported by Rusiñol et al. [197] (0.6130). In word-to-page learning-based approaches for the QBS case, the highest performance is achieved by Zhao et al. [175] (0.9406). In the IAM dataset, the top MAP is reported by the unified NN-based model of Retsinas et al. [37] (0.9633), closely followed by the synthetic and deep embedding (S+DE) features approach of Krishnan et al. [34] (0.9509) for QBS word spotting. Of note is also the work of Wilkinson et al. [70] (Ctrl-F-Net) for segmentation-free QBS KWS on IAM (0.8030). Other NN-based approaches come very close to this figure, confirming again the usefulness of neural networks in word spotting.

There also exists a computational analysis for some of the state-of-the-art methods. More specifically, for the IAM dataset, the QBS method of Almazán et al. [31] requires about 1 second to compare all $5,000$ queries against all $16,000$ dataset words on an 8-core Intel Xeon W3520 at 2.67GHz with 16Gb of RAM. Actually, it involves only one matrix multiplication to compare all queries using the attributes embedded with Common Subspace Regression (CSR), which is about 0.2 milliseconds per query. This is heavily contrasted with the work of Frinken et al. [29] which needs a few milliseconds to compare a keyword with a single text line. In the segmentation-free

framework, the work of Almazán et al. [96] requires less than 15 milliseconds on average to match a query image with a single document page. To our knowledge this complexity does not correspond to their full system. It rather employs the Exemplar Word Spotting system with product quantization (without re-ranking and query expansion) with a lower MAP (0.518) than that reported in Table 2.11 for the GW dataset. The method of Kovalchuk et al. [95] on the other hand takes 33 milliseconds on average to match a query image with all the 20 pages of the GW collection obtaining 0.501 MAP.

2.7 Remarks

In an attempt to compare different word spotting systems, we end up with the following conclusions. The research community is moving towards scalable systems that could effectively deal with the large amount of documents. At the same time, the general objective of a word spotting system is to reduce the user interference as much as possible in terms of preprocessing, parameter tuning and relevance feedback. To this end, learning-based systems which train on adequate annotated data might be more suitable than leaning-free methods. Since most learning-based methods allow the user to cast arbitrary text queries without the need for manually picking an example to trigger the search, they might yield a more preferable solution for large scale indexing and retrieval. Generally, a learning-based method achieves higher performance than a learning-free method, especially in documents which present writing style variability and are mainly written in languages of a corresponding script. However, such a method will most likely fail if tested on languages written in a substantially different script without retraining on newly annotated data. Training may also result in overfitting to a particular writing style or font. Recent works are promising in this respect though adaptiveness between completely different scripts is still a goal to be reached [174]. In the case where it is difficult to obtain labeled data, learning-free approaches provide a more practical solution. In that sense, we may say that it always depends on the application field and the available resources.

Intrigued by the aforementioned observation related to the superior performance of learning-based systems with respect to training-free methods, when training data are available for a document collection, in Section 4.3, we explore the adaptation

of a learning-based KWS system to a target domain where labelled data exist in very low quantity, therefore, being unable to represent a large variety of possible inherent writing style variabilities. The proposed framework is thus trained in a weakly supervised manner, similar in concept with the work of Gurjar et al. [159].

Chapter 3

Structural local features for handwritten keyword spotting

This chapter is divided into two methodologies which rely on the same family of local contour features to represent word images for handwritten KWS. These works were published at the early stages of our thesis. In the first part, we describe a learning-based method, which follows a query-by-example (QBE) variant, as the paradigm for segmentation-based keyword search in modern Greek handwritten text written by multiple authors. Particulary, we employ a specific case of QBE, namely query-by-word-class, using a model trained from a subset of images belonging to that class. Our approach derives from a technique for object detection in natural scene images [293], where class models are learnt directly from images and novel object instances are localized up to their boundaries in the presence of intra-class variations, clutter and scale changes. For real images, clutter is a deterrent factor in detecting an object inside an image, due to the fact that parts of this object may be covered by another object or missing. In our case of handwritten words, we consider clutter either as information not relevant to the specific word, for instance, segmentation errors, or as parts of the

word that do not reoccur among training samples, such as semicolons, full stops and accents which are not deemed useful when training a class-specific model.

Hence, our first contribution is a technique for learning a representative shape of a word-class which allows for multi-writer word spotting by adapting the model to the learnt intra-class writing style deformations. Moreover, improved feature extraction is achieved by a preprocessing step on the segmented word image. The proposed system's performance is evaluated for KWS, when applied as a classification task, using a vocabulary of word models built to this end. We note here that the proposed KWS system should not be confused with a word recognition system, in which a transcription is typically expected as the output of an input query. On the contrary, our KWS method obtains a ranked list of images, visually similar with the example-based query. Its only limitation is related to the lexicon-based (IV) fashion for query image selection as opposed to out-of-vocabulary (OOV) queries.

In the second part of this chapter the proposed KWS system differs significantly from the former one, in the sense that it is purely learning-free. Instead of requiring multiple instances of the query to cast QBE, local contour features are extracted from word images, without the need for building an average shape to resemble a word-class model. Assuming that document images have already been binarized and segmented at word level, the contribution of the proposed technique lies on the direct use of these features to retrieve the location and scale of the center of the query's bounding box inside a test image. This acts as an alignment step which initializes a non-rigid point set matching algorithm, which deforms the query word in order to approach the shape of the target dataset word images. The outcome of this matching process is a detection at point level (boundary) which is then scored by a weighted sum of four terms [293] to deal with writing-style variations of a word. As a further contribution, we extend this weighted sum with an extra term to account for false detections obtained from partial matches of the query inside the test image. The proposed method is evaluated using the Mean Average Precision (MAP) index for heterogenous handwritten scripts, written in English and early-modern Greek, respectively, and shows superior performance to a number of learning-free approaches.

3.1 Local contour features

The main ingredient of both systems presented in this chapter are the discriminative local features. Such features are generally preferred in graph-based representations [84] to encode the structural properties of a word, regardless of the inherent writing style variability. In this work, we focus on a particular case of local contour features, appropriate to represent the boundaries of a word image. To obtain these features, one should first extract the edge-maps of word images, using a standard edge detector, such as the Berkeley natural boundary detector [294]. In this respect, edge pixels (edgels) comprising these edges are initially chained into edgel-chains, then linked at their discontinuities so as to approximately fit straight segments to them, using the technique of [1], described in Section 3.1.2. Segments are typically fit over individual edgel-chains and bridged across their links. This brings robustness to the unavoidable broken edgel-chains, or else, to missing parts of letters.

3.1.1 Preprocessing

In our work however, we prefer a skeletonized version of the binary output of word images to alleviate potential errors of the underlying edge detection for a word image. Moreover, local features extracted from the skeletonized images are far less than those obtained from the edge detector output and also contain significantly less noise, which can be confirmed in Sections 3.2 and 3.3 from Figures 3.5 and 3.11, respectively.

The skeleton of each word image is extracted by applying a thinning morphological operation. The binary structuring elements used for thinning are of the extended type described under the hit-or-miss transform (i.e. they can contain both ones and zeros). Hence, the thinning of an image I by a structuring element J in terms of the hit-or-miss transform is:

$$\text{thin}(I, J) = I - \text{hit-or-miss}(I, J),$$

where the subtraction is a logical operation defined by

$$X - Y = X \cap \text{NOT } Y.$$

Since the data set used in our experiments consists of pre-segmented words, the skeleton of foreground pixels results into edge-maps which can be efficiently used

for further processing. In the following subsection, we further discuss the extraction methodology of the local contour features, which is adopted by both of our proposed KWS approaches.

3.1.2 Feature extraction

After extracting the skeleton from a binary pre-segmented word image using the aforementioned thinning operation, the edgels comprising the skeleton are chained and a smoothing spline curve is fit to each edgel-chain, providing estimates of the edgel's tangent orientations. Since a contour may be broken into several edgel-chains, or it might have branchings which are not captured by simple edgel-chaining, we link edgel-chains to counter these issues with the following criterion:

- An edgel-chain c_1 is linked to an edgel-chain c_2 if any edgel of c_2 lies within a search area near an endpoint of c_1 as it is illustrated in Figure 3.1. The search area is an isosceles trapezium. The minor base rests on the endpoint of c_1 and is perpendicular to the curve's tangent orientation, while the height points away from c_1.

This criterion links c_1 to edgel-chains lying in front of one of its endpoints, thereby indicating that it could continue over c_2. The trapezium shape expresses that the uncertainty about the continuation of c_1's location grows with the distance from the breakpoint. Note how c_1 can link either to an endpoint of c_2, or to an interior edgel. The latter allows to properly deal with T-junctions, as it records that the curve could continue in two directions (Figure 3.1(b)). Besides, it is pointed out that it is not necessary for the end of c_1 to be oriented like the bit of c_2 it links to, as in Figure 3.1(b). Tangent-discontinuous links are also possible (Figure 3.1(c)).

These edgel-chain links are the backbone structure on which the contour segment network will be built. To obtain the elements composing the network, namely, the *contour segments*, each edgel chain is partitioned into roughly straight segments. In addition to these regular segments, we also construct segments bridging over tangent-continuous links between edgel-chains, as it is shown in Figure 3.1(d). The idea is to bridge the breaks in the edges, thus recovering useful segments missed due to such gaps.

Before explaining how to build the CSN, a few definitions are provided in line with Ferrari et al. [1].

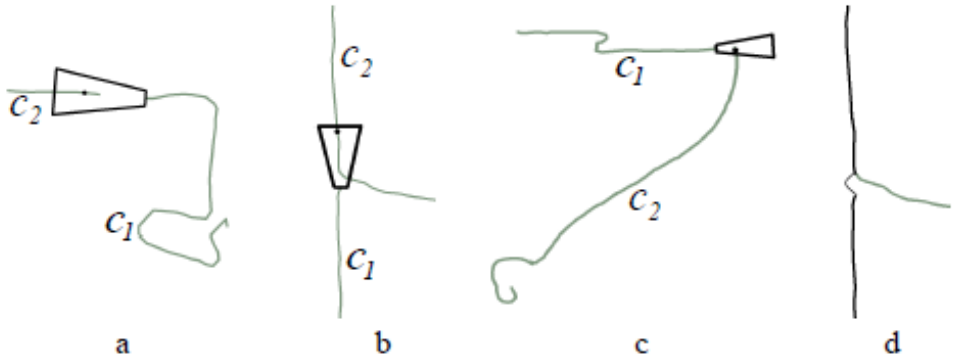


Figure 3.1: (a-c) Example links between edgel-chains. (a) Endpoint-to-endpoint link. (b) Tangent-continuous T-junction link. (c) Tangent-discontinuous link. (d) A segment (marked with an arc) bridging over link b. Figure reproduced from [1].

- Every segment is directed, in that it has a back and a front. This only serves to differentiate the two endpoints, they have no semantic difference. As a convention, the front of a segment is followed by the back of the next segment on the edgel-chain.

- Every edgel-chain link is directed as well, meaning that the edgel-chain c_1, on which the trapezium search-area rests, is at the back, while the other edgel-chain c_2 is at the front. This also defines the front and back endpoints of a segment bridging between two edgel-chains.

- For clarity, we use the word links between edgel-chains, and connections between segments.

The network is built by applying the following six rules, as it is presented in Figure 3.2. These rules connect the front of each segment to a set of segments, and its back to another set of segments. Therefore, the network structure is unconstrained and its complexity adapts to the image content.

1. The front of a segment is connected to the back of the next segment on the same edgel-chain.

2. When two edgel-chains c_1, c_2 are linked at endpoints, the segment of c_1 before the link is connected to the segment of c_2 after the link.

3. Consider a T-junction link (i. e. from an endpoint of c_1 to the interior of c_2). The segment of c_1 before the link is connected to the two segments of c_2 with the

closest endpoints. As can be seen in Figure 3.2(3), this records that the contour continues in both directions.

4. Let s be a segment bridging over a link from c_1 to c_2. The segment s is connected to the segment of c_2 coming after its front endpoint, and to the segment of c_1 coming before its back endpoint.

5. Two bridging segments which have consecutive endpoints on the same edgel-chain are connected. Here, "consecutive" means that no other segment lies in between.

6. Consider a bridging segment s without front connection, because it covers the front edgel-chain c_2 until its end. If c_2 is linked to another edgel-chain c_3, then s is connected to the segment of c_3 coming after its front endpoint. A respective rule applies if s lacks the back connection.

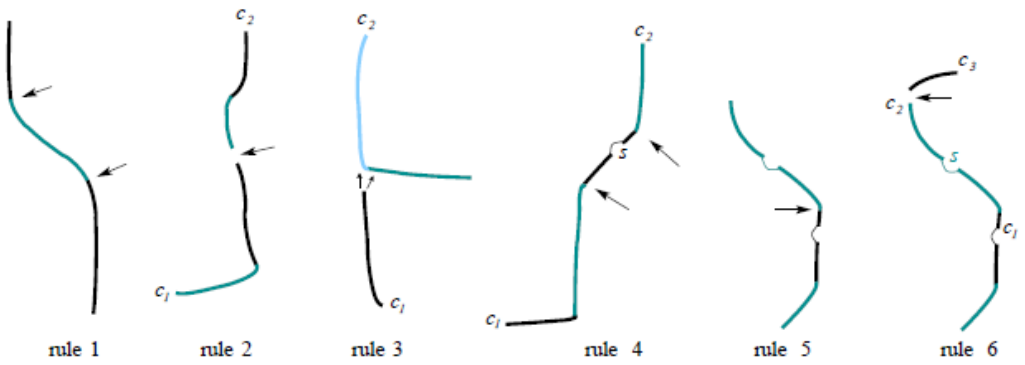


Figure 3.2: The six rules used in order to build the Contour Segment Network. They connect (arrows) regular segments and bridging segments (marked with an arc). Rules 2-6 connect segments over different edgel-chains c_i. Figure reproduced from [1].

The above rules naturally connect two segments if the edges provide evidence that they could be connected on an ideal edge-map, where all edges would be detected and perfectly chained. Moreover, it is interesting to notice that the last three rules, dedicated to bridging segments, create connections analog to those made by the first three rules for regular segments. As a consequence, both types are treated consistently.

Since each edgel-chain is typically linked to several others, these rules generate a complex branching structure, a network of connected segments. The systematic connections across different edgel-chains, together with the proper integration of bridging segments, make the network robust to incomplete or broken edgel-chains. This might

be unnecessary for the data set used in our experiments, since edge-maps resulting after the thinning operation are sufficiently connected. However, the connectivity between segments provided by the CSN, allows an efficient searching for paths through the network that resemble the word's outlines, even in error-prone segmented word images, such as handwritten historical document images.

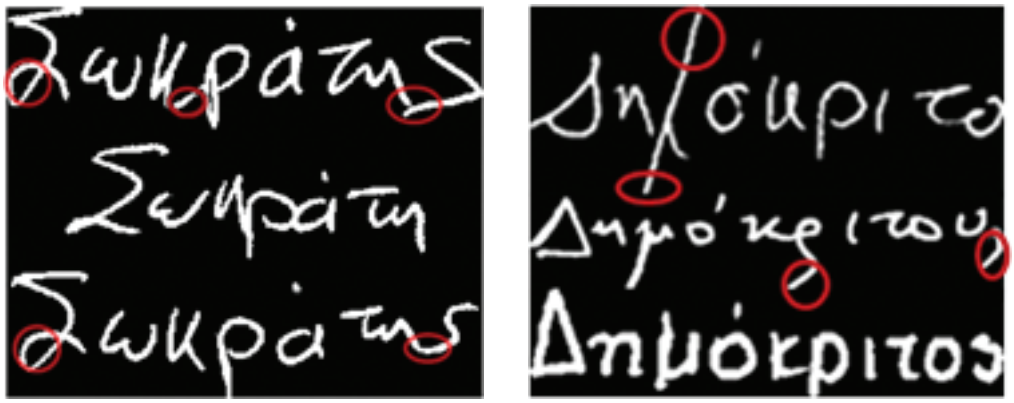
3.2 Learning-based KWS in handwritten text using contour-based models

At the heart of our learning-based KWS approach lies the word image matching algorithm. In order to achieve a matching of high accuracy, we utilize an objection detection technique, initially proposed in [293] to detect a query word up to its boundaries. The key to accomplish boundary-level localization of a query word in a test image is to build an explicit shape model formed by continuous connected curves, completely covering the word's outlines. Therefore, the challenge is to determine which contour points should belong to the word-class boundaries, while discarding possible segmentation errors and details specific to individual instances, such as the extended parts of calligraphy letters or accents, as it is depicted in Figure 3.3. Sample word images are selected from the *Modern14* dataset which is presented in Section 2.6.1).

3.2.1 Feature description

The local features we use are the scale invariant PAS features conceived by Ferrari et al. [293]. A PAS feature, for *pair of adjacent segments*, $P = (x, y, s, e, d)$ has a location (x, y) which consists of the mean over the two segment centers, a scale s which is the distance between the segment centers, a strength e as the average edge detector confidence over the edgels, with values in [0, 1] (in our case of thinned binary word images, $e = 1$) and a descriptor $d = (\theta_1, \theta_2, l_1, l_2, r)$, invariant to translation and scale changes. Figure 3.4 depicts a typical example of a PAS when it is approximated by straight adjacent segments.

A number of example PAS features are illustrated in Figure 3.5, both for an edge-map extracted from Berkeley's boundary detection algorithm and in the case of



(a) (b)

Figure 3.3: (a) Three instances of the words "Σωκράτης" (Socrates in English) written in Greek by the same writer. (b) Three examples of the word "Δημόκριτος" (Democritus) written by different authors. The red areas indicate parts of the words which are rarely repeated among instances.

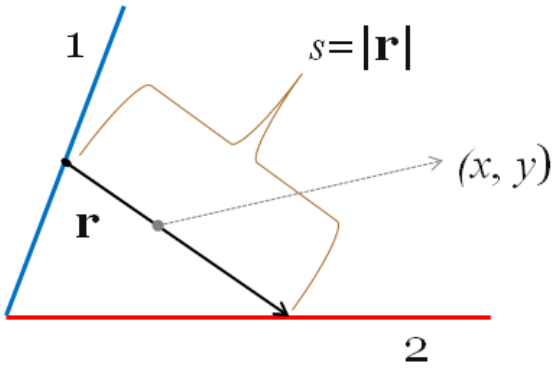
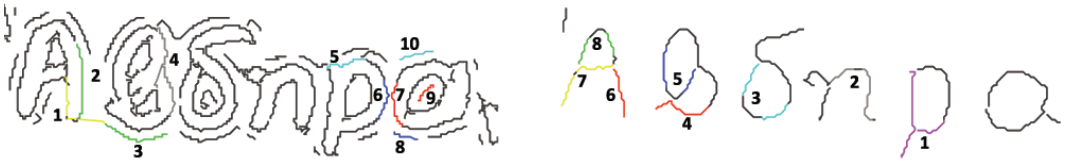


Figure 3.4: Pair of adjacent segments (PAS) description in the simplified case where straight lines are fit to regular image segments

thinned word images. These examples show that skeletonization by thinning has a positive impact on a word image before detecting PAS features since the image noise levels decrease, as the number of clutter PAS becomes smaller and PAS stemming from thinned images cover mainly informative parts. This also reduces the computational complexity of detecting them.



(a) (b)

Figure 3.5: Examples of PAS features for the word "`$A\beta\delta\eta\rho\alpha$" (Abdera in English). (a) 10 PAS for the edge-map resulted after Berkeley's edge detection, (b) 8 PAS detected on the same word image after thinning.

3.2.2 Feature similarities - codebook

A flexible measure to accommodate intra-class variability is the PAS dissimilarity $D(\mathbf{P}, \mathbf{K})$ between the descriptors $\mathbf{d}^p$, $\mathbf{d}^k$ of two PAS $\mathbf{P}, \mathbf{K}$, defined by:

$$D(\mathbf{d}^p, \mathbf{d}^k) = w_r \|\mathbf{r}^p - \mathbf{r}^k\| + w_\theta \sum_{i=1}^{2} D_\theta(\theta_i^p, \theta_i^k) + \sum_{i=1}^{2} |\log(\frac{l_i^p}{l_i^k})| \tag{3.1}$$

The first term is the difference in the relative locations of the segments, the second term contains the difference between segment orientations and the last term accounts for the difference in lengths. As segment lengths are often inaccurate, higher weight is given to the two other terms of the dissimilarity measure. The parameters w_r, w_θ are fixed weights with values 4 and 2, respectively.

Finally, following the bag of features paradigm, we compose the codebook of PAS types, also employed in [293], as a "visual vocabulary", each capturing a different kind of local shape structure. The codebook is created by clustering the PAS inside the training images according to their descriptors. Apart from revealing the frequency at which feature types occur, the codebook allows to avoid explicitly comparing every test image features to every feature from the training images. Instead, comparison to much fewer feature types suffice. For each cluster, the medoid PAS, minimizing the sum of dissimilarities to all the others is selected as a representative. The codebook C is the collection of the descriptors of these centermost PAS, the PAS types t_i, a number of which are illustrated in Figure 3.6.

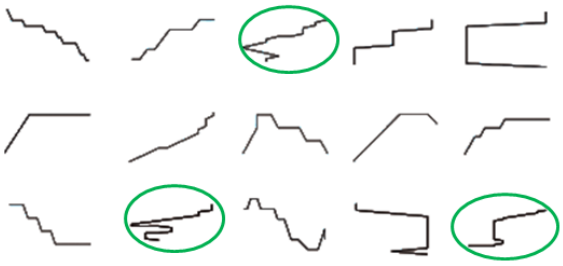


Figure 3.6: The 15 most frequent PAS types from 60 thinned instances of the word "Σωκράτης" (Socrates) used to train the average word. The green areas contain the upper parts of Σ or the whole letter ς.

3.2.3 Shape model representation

The challenge in training the mean shape to represent a word-class is to discover which contour points belong to the common class boundaries and to put them in to full point-to-point correspondence across the training samples. The only prerequisite for training such a shape is for the words to be annotated by a bounding box, which for our segmented words is set to be the whole image. The technique for building this model is composed of four stages, as it is illustrated in Fig. 3.7(b-e). Our intention is not to analyze in depth the underlying learning methodology, but to show why this approach is suitable for our problem. Thus, we refer the reader to [293] for a detailed description with respect to the training procedure used and briefly present the intermediate steps.

3.2.4 Collection of parts model

The first step is to determine model parts as PAS, frequently reoccurring with similar locations, scales and shapes. This implies that PAS not belonging to the class boundaries are not correlated across different examples. To remove translation and scale differences as well as cancel out word variations due to different aspect ratios, the training images are properly aligned. Since a correspondence between two PAS induces a translation and scale change, they can be efficiently used within a Hough-style voting scheme. Each voting space is associated to a codebook PAS type and has three dimensions, two for location and one for size. Subsequently, each PAS inside the training images votes for the existence of a part of the class boundary with shape, location and size like its own. Then, all voting spaces are searched for local maxima, which in turn yield model parts with a specific location, size and shape.

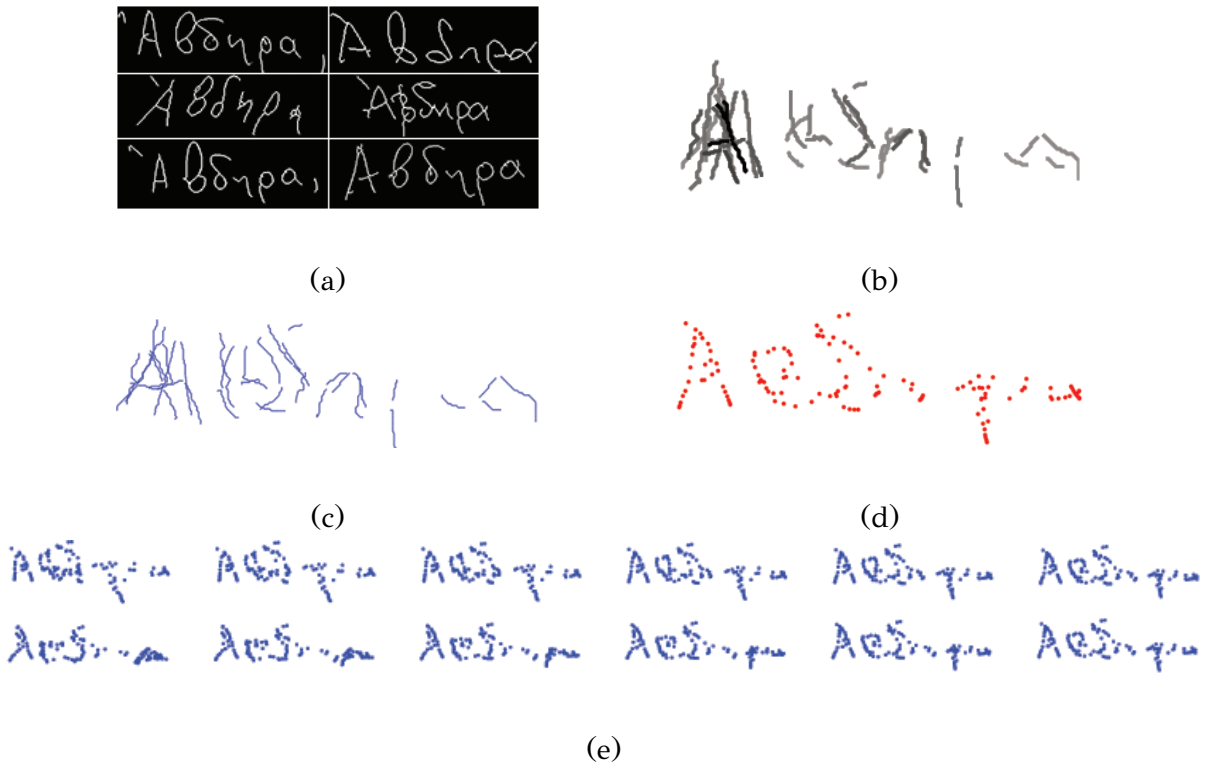


Figure 3.7: Learning the shape model. (a) Six training examples (out of a total of 60). (b) Collection of parts (COP) model. (c) Occurrences selected to form the initial shape. (d) Refined shape. (e) First two modes of variation (mean shape on the right top-bottom).

The success of this process is partly attributed to adopting PAS as basic shape elements. Unlike other local features, such as individual edgels, the shape of the PAS, expressed as the assignment to codebook types and its size (relative to the aligned image), are more distinctive than the orientation of an edgel. Hence, it is very unlikely for clutter PAS to accidentally have similar locations, sizes and shapes concurrently. In addition, while PAS are soft-assigned to all types, a substantial spatial smoothing to the voting spaces before detecting local maxima allows for PAS from different images to vote for the same part. This way intra-class variability is dealt with a low risk of accumulating clutter. Finally, the proposed method sees all training data at once, thus avoiding pairwise matching and rendering its computational complexity linear in the total number of PAS inside the training images.

3.2.5 Assembling the initial shape

The collection of parts (COP) learned so far captures class boundaries well and delivers a sense of the general shape of the word-class (Figure 3.7(b)). However, the COP

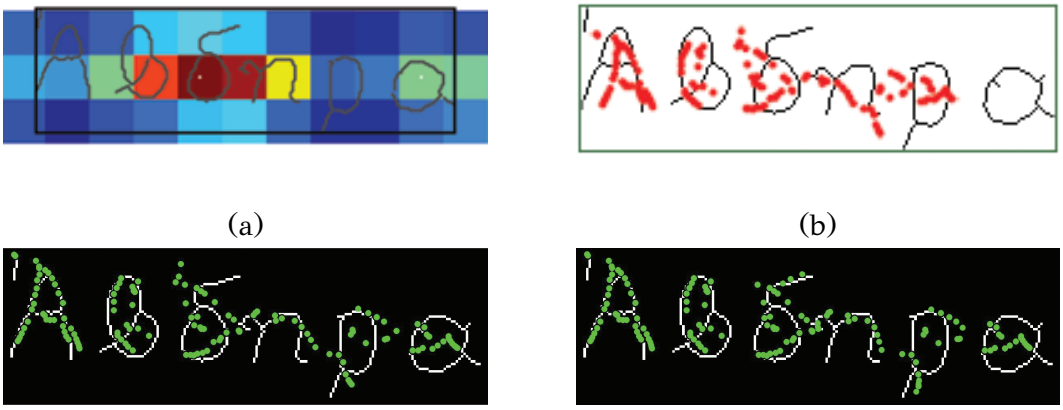
model does not take into account the shape of whole words. It is a loose collection of parts learnt rather independently, each focusing on its own local scale. To support localizing word instances up to their boundaries, a more globally consistent shape is needed. Ideally, its parts would be connected into a whole shape featuring smooth, continuous lines. As model parts may occur several times in different images, variants of such a shape model can be assembled by selecting different occurrences for each part, so as to form larger aggregates of connected occurrences which finally lead to the best connected shape (Figure 3.7(c)).

3.2.6 Model shape refinement

A refined version of the initial shape is obtained by matching it back onto the training images, using the deformable matching algorithm of Chui and Rangarajan [295]. More specifically, each point set E inside the training images is firstly aligned with a sample set S, obtained from the initial shape and then put in to point-to-point correspondence using the aforementioned non-rigid point matcher. This estimates a thin-plate spline (TPS) transformation from S to E, while rejecting edgels not corresponding to any point of S and the process results in a backmatched shape for every image. These backmatched shapes are averaged through Cootes' variant of Procrustes analysis [296], thus yielding an improved mean shape. The resulting mean shape is then used as the new sample point set and the whole process is iterated two to three times till the refined shape model is produced (Figure 3.7(d)).

3.2.7 Learning intra-class deformations

The backmatching of the previous step provides different examples of the variations within the desired word-class due to different non-rigid registrations of the model upon training images. These examples can be used to learn a statistical model of intra-class deformations [296]. The key component is to consider each example shape as a point in a $2p - D$ space (with p the number of points on each shape) and model their distribution with principal component analysis (PCA). The eigenvectors returned by PCA represent modes of variation and the associated eigenvalues their importance, namely, how much the example shapes deform along them, as it is depicted in Figure 3.7(e). By keeping only the $n < 15$ eigenvectors corresponding to the largest eigenvalues representing 95% of the total variance, it is feasible to approximate the valid



(a) (b)

(c) (d)

Figure 3.8: Word detection. (a) A local maximum in Hough space defines the word's center. (b) Initialization of TPS-RPM by centering the model to the word's center. (c) The output shape with unconstrained TPS-RPM. It captures the word relatively well, except for the letters 'δ','η' and 'ρ', where it is strongly attracted by the edgel orientations. (d) Output of the shape-constrained TPS-RPM. Now the word is more properly recovered.

region in which the training examples lie.

3.2.8 Word image matching

The matching of the word-class model learnt in the previous section presents several challenges. Word segmentation errors may result in a cluttered image, where only a percentage of points is deemed valid for further processing. Moreover, to cope with inter and intra-writer variability for a word-class, the model must be deformed into a shape, similar to the particular instance. These limitations are tackled in a two-stage algorithm.

The model is firstly aligned to the test image by a Hough-style voting scheme (similarly to the learning stage) which determines its approximate position and scale inside the image (Figure 3.8(a)). This acts as an initialization (Figure 3.8(b)) to the subsequent non-rigid point matcher which deforms the model according to its learnt modes of variation to capture the shape of the unknown word, as it is depicted in Figure 3.8(d).

The method for estimating the word's location and scale inside the test image and the constrained version of the non-rigid TPS robust pointer matcher, which searches only inside the valid region spanned by the training examples, are analyzed in [293].

Hence, we simply illustrate its beneficial outcome. As it can be seen in Figure 3.8(c-d), the constraints on which the matcher is subject to, have a positive impact.

Since the data set used in our experiments consist of segmented word images, the first stage of estimating the candidate location and scale of the desired word may be redundant and therefore, the matching could already begin without initialization. Yet, possible locations and scales (different Hough maxima) of the word of interest result into separate detections, from which we retain the one with the highest score. This provides better outputs, contrary to those accrued by applying the matcher solely in the first place.

3.2.9 Experimental evaluation of learning-based KWS

The data sets used in our experiments originate from the ICDAR'07 Handwriting Segmentation Contest and were particularly used by Papavassiliou et al. [208] in both ICDAR'07 and ICDAR'09 contests. The document images, which actually are part of the *Modern14* dataset (Section 2.6.1) cover a wide range of cases which occur in unconstrained handwriting. One type of such cases derives from 25 writers, who were asked to copy a given Greek text of approximately 150 words. The segmentation output of the algorithm described in [208], on these document images, results into clean word images comprising our training and testing data sets.

3.2.10 Setup

In order to train a class-specific model, we have manually annotated the words belonging to a particular word-class. This was carried out for 10 word-classes, as it is shown in Table 3.1. Each class contains one to four instances per writer and thus the number of words for a class varies from 25 to 100. The data set comprising each class is split into training and validation data. Each class-specific model is trained from a random sample containing 80% of the images belonging to that class. We iterate this process 5 times, yielding 5 models per class, from different training sets, to prove the stability of the learning process through a *repeated random sub-sampling validation*. We refer to learning and testing on a particular split of the images as a *trial*. Finally, all experiments are run with the same parameters.

To account for false positives, Table 3.1 also illustrates negative test sets containing no instance of the respective word-class but other words written by different authors.

Figure 3.9: Examples of thinned word images from the data sets used in our experiments.

Both validation and negative sets are equally distributed. Example thinned images for our vocabulary, which consists of 10 word-classes, are depicted in Figure 3.9.

Table 3.1: Number of examples used in the experimental protocol

word-class	training set	validation set	negative set
$\Sigma\omega\kappa\rho\acute{\alpha}\tau\eta\varsigma$	60	15	15
$\Delta\eta\mu\acute{o}\kappa\rho\iota\tau o\varsigma$	80	20	20
$\grave{A}\beta\delta\eta\rho\alpha$	60	15	15
$\alpha\rho\epsilon\tau\acute{\eta}$	20	5	5
$\alpha\gamma\alpha\theta\acute{o}$	40	10	10
$\delta\iota\kappa\alpha\sigma\tau\acute{\eta}\rho\iota o$	40	10	10
$\sigma o\varphi\acute{\iota}\alpha$	20	5	5
$\Theta\rho\acute{\alpha}\kappa\eta$	60	15	15
$\varphi\iota\lambda o\sigma o\varphi\acute{\iota}\alpha$	40	10	10
$\pi\alpha\tau\acute{\epsilon}\rho\alpha\varsigma$	40	10	10

3.2.11 Intra-class word detection

Initially, we evaluated the detection of a novel word instance up to a bounding box. A detection is deemed correct only if the intersection-over-union (IoU) ratio between the detection's bounding-box and the ground-truth's one overlap more than 50%. Otherwise, it counts as a false positive. Hence, we match the models learnt for each class to the images of the validation and negative sets, respectively. Although such a measure is typically used in segmentation-free KWS, its usefulness in this work is to prefer an initialization of the TPS-RPM algorithm that will lead to a more accurate boundary localization.

We present the results in Table 3.2 through indices such as the total detection rate (TDR) and the detection rate at 0.1 false positives per image (DR at 0.1 FPPI), averaged over the 5 trials. Moreover, we show the weighted mean value of each index and its standard deviation, averaged over all classes. The weights of TDR correspond

to the number of positive images while those of DR at 0.1 FPPI correspond to the number of positive and negative images.

Table 3.2: Statistics for all word-classes averaged on all trials.

word-class	TDR	DR at 0.1 FPPI
$\Sigma\omega\kappa\rho\acute{\alpha}\tau\eta\varsigma$	0.987	0.960
$\Delta\eta\mu\acute{o}\kappa\rho\iota\tau\varsigma$	0.860	0.848
$\grave{A}\beta\delta\eta\rho\alpha$	0.880	0.861
$\alpha\rho\epsilon\tau\acute{\eta}$	0.640	0.574
$\alpha\gamma\alpha\theta\acute{o}$	0.820	0.808
$\delta\iota\kappa\alpha\sigma\tau\acute{\eta}\rho\iota\varsigma$	0.920	0.899
$\sigma\varsigma\varphi\acute{\iota}\alpha$	0.800	0.755
$\Theta\rho\acute{\alpha}\kappa\eta$	0.880	0.855
$\varphi\iota\lambda\varsigma\sigma\varsigma\varphi\acute{\iota}\alpha$	0.920	0.896
$\pi\alpha\tau\acute{\epsilon}\rho\alpha\varsigma$	0.840	0.804
mean	0.874	0.850
std	0.090	0.103

3.2.12 Word spotting using a vocabulary

To this end, we have examined the system's ability of detecting novel keywords correctly. The latter index shown in Table 3.2 provides us insight about the difference in the scores between correct and false detections. Essentially, the scores of false detections are low enough, so as to allow discriminating them among other classes and classify them properly. Based on this observation, we assess the system's performance in a word spotting task by combining the information provided by all models.

Particularly, given an unknown word that belongs to an already known (in the vocabulary) word-class, the system matches the word to all the class-specific models learnt and classifies it to a particular class according to the following criterion:

- The class-specific model achieving the highest matching score with the keyword, is the one specifying the keyword's original class.

Following the configuration defined in Section 3.2.10, we make use of the same models learnt for each word-class, using 80% of the available images. The test set consists of the words that were not used to train each class-specific model. Then, we match the 10 models (one per word-class) to a new word from the test set and predict the word's original class by favoring the model which achieved the maximum score (among 10 scores). This process is iterated 5 times using the corresponding models of each trial.

Thus, we cast word spotting as an image retrieval and pattern classification task and estimate the efficiency of the proposed method using the F-measure metric, which is defined by:

$$F = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

We also present the confusion matrix produced at the first of the 5 trials in Figure 3.10(a). Each row of the matrix represents the percentage of word instances in a predicted class, while each column represents the percentage of words retrieved by each class. Figure 3.10 also shows the F-measure obtained by the system, both for the first trial (F_1) and on average (F_{avg}) over all trials.

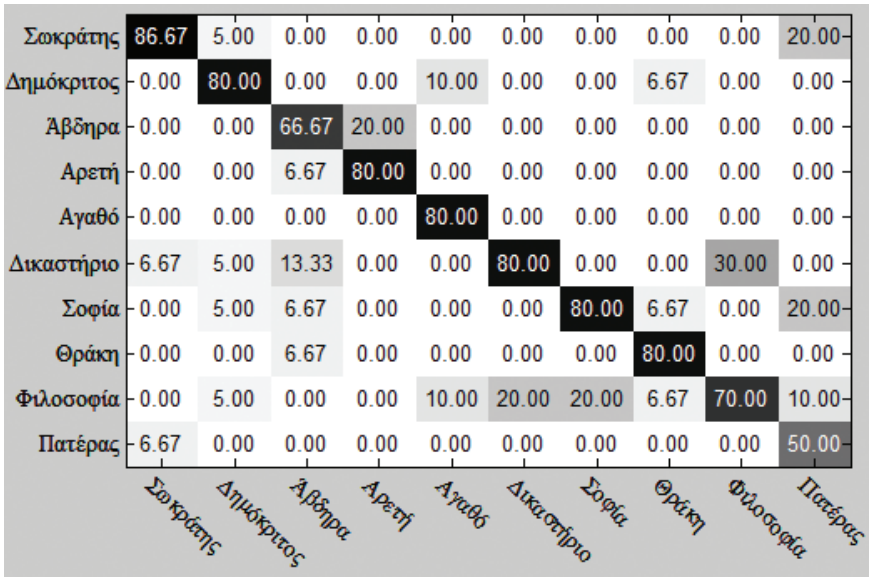
A way to further improve the performance is to combine the scores of separate models learnt for the same word-class and incorporate them in the decision step of the classification task. Specifically, in setup (b) we train 5 models per class using random samples containing 80% of the images inside the respective training subset, namely, each sample consists of $80\% \times 80\% = 64\%$ of the initial training images. Again, the test set contains no instance used for training. In the same spirit with the previous setup, we iterate the whole process 5 times and present the confusion matrix along with the F-measure values for the first of the 5 trials and on average. Although the system performs substantially better (almost 12% on average), the computational costs are by far increased.

3.3 Learning-free approach for language independent HKWS

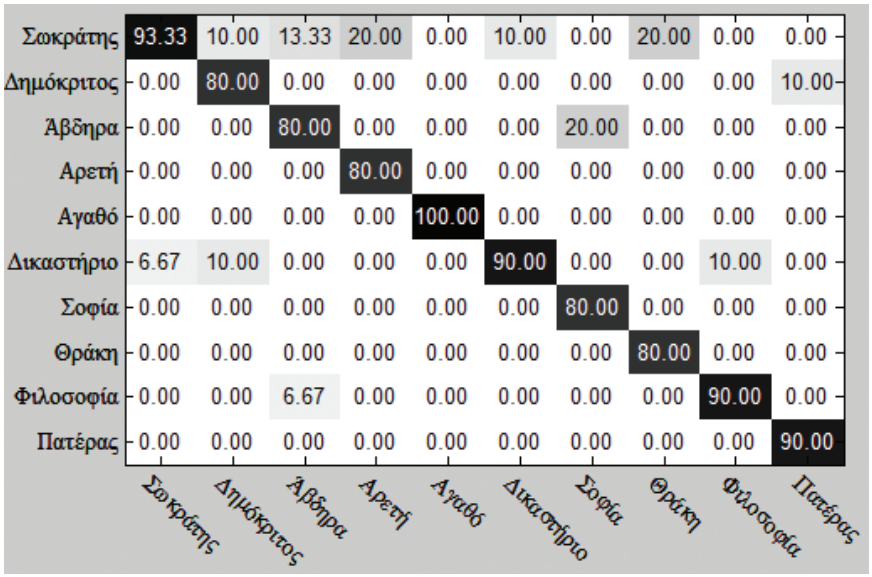
3.3.1 Word representation

In our prior work [41] (Section 3.2), it was shown that to achieve a matching of high accuracy in documents which present variability in writing style, it is essential to detect a query word at boundary level. Such a detection requires a contour-shape, formed by continuous connected curves, to describe each word image. This representation allows for determining the candidate location and scale of the query inside the test image which is then used as input to the subsequent non-rigid point matching scheme (Section 3.2.8).

To create this contour shape, again, we extract the skeleton of a word by applying a thinning morphological operation to the binarized word images. This procedure



(a) $F_1 = 0.763$, $F_{avg} = 0.716$



(b) $F_1 = 0.865$, $F_{avg} = 0.834$

Figure 3.10: Confusion matrices for one of the five trials of the first (a) and second (b) experimental setup and corresponding F-measures.

(a)

(b)

Figure 3.11: (a) The word "$M\acute{\eta}\tau\eta\rho$" ("Mother" in English) from the ST46 dataset written in early modern Greek. (b) The word "Orders" from the GW20 dataset. Extracted PAS features from each thinned image are shown on the right (the figure is better seen in color).

erodes away the boundaries of foreground shapes as much as possible, but does not affect pixels at the ends of lines. Edge pixels (edgels) comprising the skeleton are initially chained into edgel-chains, which are then linked at their discontinuities and approximately straight segments are fit to them, using the technique described in Section 3.1.2. Segments are fit over individual edgel-chains and bridged across their links, similarly in spirit with [1].

3.3.2 Word description

The next step is to detect the pairs of adjacent segments (PAS) originally conceived by Ferrari et al. [293] and use them to represent each word. As it was mentioned in the previous section, a PAS feature, $\mathbf{P} = (x, y, s, d)$ has a location (x, y) which consists of the mean over the two segment centers, a scale s which is the distance between the segment centers and a descriptor $\mathbf{d} = (\theta_1, \theta_2, l_1, l_2, r)$, invariant to translation and scale changes. Example binary instances of the words "$M\acute{\eta}\tau\eta\rho$" ("Mother" in English) written in early modern Greek and the word "Orders" from the GW20 benchmark [2], along with their respective skeletons and a subset of PAS features are illustrated in Fig. 3.11. Each color on the right of the figure corresponds to a PAS whereas the numbers correspond to its segment IDs.

3.3.3 Descriptor similarities

Connecting segments over edge discontinuities renders PAS features robust to interruptions along the word contour and to short missing parts. These may be due to segmentation errors, faded ink or poorly pressed thin strokes. It is interesting to notice that PAS may overlap, meaning that they can share segments and thus cover pure portions of a word's boundary. Consequently, they can be easily detected across instances of the same word-class, in terms of finding a common structure among similar instances.

To this end, we make use of the similarity measure between two word images as it is defined by equation (3.1) in Section 3.2.

3.3.4 Word image matching

The first step to detect occurrences of the query inside the test images is to determine their possible location and scale using the predefined dissimilarity measure (3.1). More specifically, each PAS $\mathbf{P}$ inside the query is matched with every PAS $\mathbf{K}$ from the test image according to $D(\mathbf{P}, \mathbf{K})$. If the dissimilarity is lower than a specific threshold γ then this match votes for a candidate location and scale of the query's center inside the test image. Each vote is weighted by $(1 - D(\mathbf{P}, \mathbf{K})/\gamma)$.

For instance, Figure 3.12(a) depicts the query "$Μήτηρ$" and test word "$Μητέρα$", which is rather relevant, though not an actual occurrence. Local maxima inside the $3D$ voting spaces (location, scale) yield approximate positions and scales of the query's center inside the test image. These act as different initializations (Figure 3.12(b), 3.12(c)) to the subsequent non-rigid point matcher which deforms the query to capture the shape of the unknown word, as it is shown in Figure 3.12(d) for the initialization of Figure 3.12(b).

Regarding the first stage of the matching process, the success of this alignment of the query inside the test image is attributed to adopting PAS as basic shape elements. Unlike other local features, such as individual edgels, the shape of the PAS and its size, are more distinctive than the orientation of an edgel. Hence, it is very unlikely for a set of PAS not belonging to a common shape structure of the query-class, to accidentally have similar locations, sizes and shapes across instances. In other words, a subset of the query's PAS is common among its instances.

As for the second step, we apply the thin plate spline robust point matching

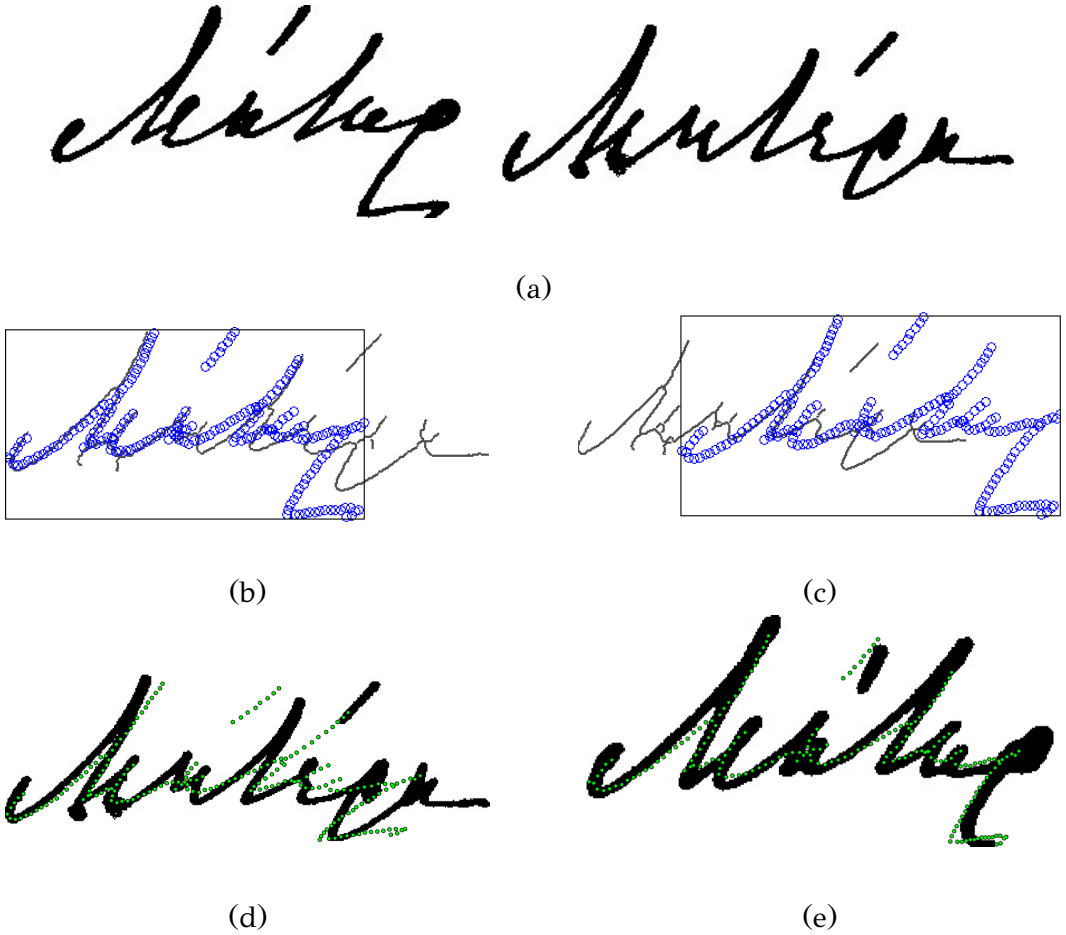


Figure 3.12: Query detection. (a) Query image on the left, test image on the right. (b)-(c) Initializations of TPS-RPM by centering the query to the word's center. (d) The output shape (false positive) is superimposed in green on the test image. (e) Superimposed output shape in green upon an actual instance (the figure is better seen in color).

(TPS-RPM) algorithm [295], which matches two point sets $\mathbf{V} = \{\boldsymbol{v}_i\}_{i=1,\dots,N}$ and $\mathbf{X} = \{\mathbf{x}_i\}_{i=1,\dots,M}$, by applying a non-rigid TPS mapping parameterized by $\{\mathbf{c}, \mathbf{w}\}$ to $\mathbf{V}$. TPSs are chosen because they can be decomposed into affine and non-affine subspaces as it is shown by the following vector valued function:

$$\mathbf{f}(\boldsymbol{v}_i) = \boldsymbol{v}_i \cdot \mathbf{c} + \boldsymbol{\phi}(\boldsymbol{v}_i) \cdot \mathbf{w} \tag{3.2}$$

where $\mathbf{c}$ is the affine component and $\mathbf{w}$ is a non-affine warping coefficient, which is combined with the TPS vector valued kernel $\boldsymbol{\phi}(\boldsymbol{v}_i)$ to form the non-rigid warp. TPSs minimize an energy function by iteratively alternating between updating a correspondence matrix, while keeping the transformation $\{\mathbf{c}, \mathbf{w}\}$ fixed and vice versa. Moreover, it rejects points for which no correspondence exists.

In line with [293], a detection at point level is scored by a weighted sum of four terms which is explained as follows:

1. The amount of matched query points to the points of the test image with a high confidence measure. These are all points $\boldsymbol{\upsilon}_i$ with $\max_{j=1,\ldots,N}(\mathbf{m}_{ij}) > 1/N$, where $\mathbf{m}$ is the correspondence matrix.

2. The sum of square distances between the matched query points and the corresponding image points, which is made scale-invariant by normalizing them by the squared range r^2 of the image point coordinates (width or height, whichever is larger).

3. The deviation $\sum_{i,j\in\{1,2\}} (\mathbf{I}(i,j) - \mathbf{c}(i,j)/\sqrt{|\mathbf{c}|})^2$ of the affine component $\mathbf{c}$ of the TPS from the identity $\mathbf{I}$. The normalization by the determinant of $\mathbf{c}$ factors out deviations due to scale changes.

4. The amount of the non-rigid warp $\mathbf{w}$ of the TPS $\text{trace}(\mathbf{w}^T\boldsymbol{\Phi}\mathbf{w})/r^2$, where $\boldsymbol{\Phi}$ is a $N \times N$ matrix formed by the kernels $\boldsymbol{\phi}(\boldsymbol{\upsilon}_i)$.

This scoring integrates the information provided by a matched shape. Its value is high when TPS fits many points well (terms 1 and 2), without having to distort much (terms 3 and 4). It is also interesting to note that different initializations from the previous stage result into separate detections from which we retain the one with the highest score. The second step of the proposed matching scheme is crucial for obtaining a more accurate detection. While the query alignment stage handles invariance in terms of translation and scale, the non-rigid registration algorithm deals with the case of skewed words or slanted characters, which are rather frequent in handwritten documents.

Finally, we add a term to tackle false detections of partial matches, such as that of Figure 3.12(d). Assuming that B_{test} expresses the image boundary points and that B_{query} consists of the matched output points to the test image, we propose an accuracy term as the average value between two measures:

1. *Coverage* is the percentage of points from B_{test} closer than a threshold t from any point of B_{query}.

2. *Precision* is the percentage of points from B_{query} closer than t from any point of B_{test}.

The measures are complementary and t is set to be 4% of the diagonal of the bounding-box of B_{test}. In our implementation, the relative weights between these five terms have been selected manually and kept fixed in all experiments. The impact of this extra term on the scoring function is that it renders scores between correct and false detections even more discriminative. In fact, the output shape of Figure 3.12(d) achieves a matching score with value 25.61% whereas the true positive score of the output shape in Figure 3.12(e) is 82.66%.

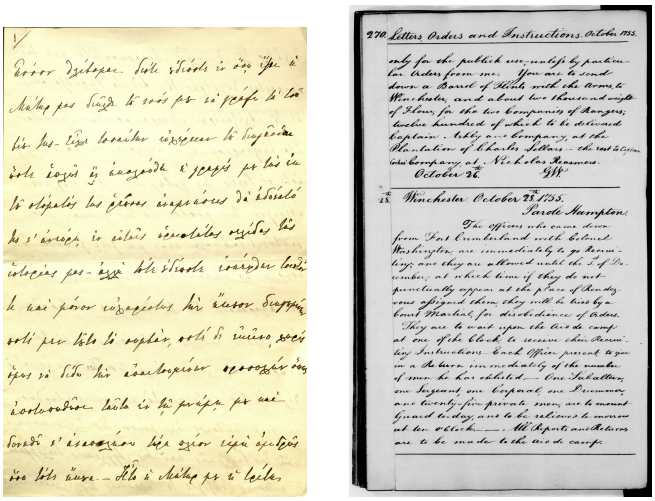
3.3.5 Experimental evaluation

In this section, we present the datasets used to evaluate the proposed word spotting approach as well as the criteria applied for selecting appropriate queries. Then we briefly refer to the state-of-the-art (at the time this work was published back in 2015) QBE systems upon which comparisons are made for each dataset.

3.3.6 Datasets and protocol

Experiments are carried out on two challenging datasets. The first dataset is written in early modern Greek by Sophia Trikoupi, during the 19^{th} century. There are 46 pages of handwritten polytonic text containing 4939 words, which derive from the archives of the Hellenic Parliament library. A sample page from the ST46 dataset is illustrated in Figure 3.13(a). Text is rather cursive accompanied by intra-writer variability among instances of the same word. In order to evaluate our method we selected words whose occurrences appear more than five times and their length is greater than 6 characters. The query list provided by this criterion includes 21 distinct words along with their instances, yielding a total number of 141 queries. All pages are binarized using the technique described in [192] and manually segmented at word level. Each word is manually annotated and we only deem an exact match of the query inside the test image as a hit.

The second dataset is the English manuscript GW20 from the George Washington collection [2], containing 20 pages of historical handwritten cursive text which include 4860 words. A sample page from this collection is shown in Figure 3.13(b). Similarly to Leydier et al. [196], we selected the same 15 words to evaluate our method. These are the most significant words in terms of occurrence frequency and semantics. We consider all instances of each of the 15 words, comprising a total number of 306



(a) (b)

Figure 3.13: Sample pages from (a) the ST46 dataset and (b) the GW20 benchmark [2], respectively.

queries. In line with the ST46 benchmark, close hits such as the words “Fort” and “fort” are deemed as false positives in the evaluation task.

Finally, one important but not restrictive aspect of our approach is the parameter estimation of our system. All parameters concerning the proposed system are estimated once using a small subset of handwritten word images from the IAM dataset and kept fixed in all experiments. Neither query nor dataset specific tuning is applied. As a means to improve the speed of the proposed matching scheme we introduce a pruning criterion which discards unlikely similar matches. This is based on the difference in the size of the descriptors between two words as well as the difference in their respective number of PAS. Such a pruning decision step, before comparing two words, seems to not only avoid at least half of the total matches to be processed per query, but also improve the average precision of our system, with low risk of reducing its recall.

Considering the above, we evaluated the performance of the proposed approach using the Mean Average Precision (MAP). This metric is calculated using the trec_eval software as it is implemented by the National Institute of Standards and Technology (NIST) [1]. Concisely, it is the average value of the area under the Precision-Recall curve over all queries.

[1]The trec_eval software is available at http://trec.nist.gov/trec_eval

3.3.7 Word spotting results

Before presenting the results we briefly discuss the reference systems used to compare the performance of our approach. The first system is the work of Gatos et al. [297]. Therein, a combination of word image normalization and feature extraction methods is presented for cursive handwritten word recognition. The second approach, which is described in [238], introduced the idea of adaptive zoning features for word recognition in historical, machine-printed documents. These features are extracted after adjusting the position of every zone based on local pattern information. The adjustment is performed by moving every zone towards the pattern body according to the maximization of the local pixel density around each zone. The final approach is the DTW method, based on the word profiles of Rath et al. [248] for handwritten historical documents.

Following the configuration defined in Section 3.3.6, we compare our system with these reference systems and illustrate the results for both datasets in Table 3.3. With respect to the first two reference systems [238, 297], we should note that they were originally created for different datasets. The method of Gatos et. al [297] was tested on the IAM benchmark, containing text written by multiple authors, while [238] was applied on historical machine printed text. The results shown in Table 3.3 indicate that their adaptation flexibility to different scripts is not trivial. As for the DTW method, it is only almost 4% worse than the proposed system in the ST46 dataset, whereas in the GW20 benchmark, it's MAP is by far lower than that of our approach. This confirms our expectation that our system would be able to perform well in different scripts, as it treats word images as $2D$ shapes, independently of the underlying language.

Table 3.3: Mean Average Precision for various methods

Method	**ST46** (141 queries)	**GW20** (306 queries)
Efficient Recognition [297]	39.44%	21.93%
Adaptive Zoning [238]	40.38%	22.50%
DTW [248]	56.18%	22.08%
Proposed	60.04%	37.86%

In this work, we propose a shape matching technique for spotting handwritten words in the presence of intra-class variability. The approach was tested in two challenging datasets and outperformed a number of QBE techniques, thereby assuring its

stability across different scripts. There is, however, a tradeoff between the accuracy and computational cost of the shape matching procedure. This means that we could re-estimate the parameters of the whole system in order to increase the speed at the cost of precision.

Chapter 4

Compact word image representations for unconstrained HKWS

This chapter includes two word segmentation-based approaches for handwritten KWS in historical and modern document images. The first work was published in 2015 [43] and extends a seminal at that time approach, which relies on the attribute-based model of Almazan et al. [31] for multi-writer word spotting and recognition, aiming to realize it for polytonic Greek documents. To this end, three alternatives are suggested to expand the model's capacity so as to handle the Greek alphabet and its various combinations of diacritic marks. The proposed descriptor actually extends the binary encoding of a word image which simulates the existence or absence of an attribute at a specific position in the word, so as to include language-dependent characteristics present in polytonic Greek documents. Numerical experiments on polytonic machine-printed and handwritten text are carried out for both word spotting and recognition. However, in this thesis, we only focus on the prior KWS task. The extended model is shown to outperform other state-of-the-art methods in word spotting trials.

The second part of this chapter follows the recent trends in deep learning-based approaches. Its main focus lies on the use of similar, successfully employed attributes

[31] when they are obtained from deep feature extractors [4], and their application for KWS on document collections when very few annotated images exist. To this end, adversarial learning in combination with spatial transformer networks [5] is proposed to obtain discriminative deformations of the feature space leading to compact deep feature representations which alleviate the adaptation of the proposed KWS system into weakly supervised manuscripts.

4.1 Using attributes for KWS in polytonic Greek documents

Similarly in text nature with historical manuscripts, polytonic Greek script also presents several challenges. It is a script that has been used to write the Greek language throughout various stages of its evolution, since its introduction as a standard in the late antiquity and up until modern times. The polytonic Greek script is based on the Greek alphabet, and comprises both capital and small versions of the letters, accompanied by the addition of special diacritics that are placed above or below the respective letters [298]. It has been practically the sole script available to write Greek until the introduction of monotonic Greek -a simplified version of polytonic Greek- in 1982. Thus, it can be well understood that a huge amount of both handwritten as well as machine-printed documents exists in polytonic Greek.

Not much work in the literature of text understanding is targeted to polytonic Greek, albeit the volume and academic importance of many of the available texts. Commercial OCR systems do exist for monotonic, typewritten Greek, but processing of polytonic printed text is known to give poor results [132]. Word spotting techniques that use learning-free, zoning features are proposed in [132, 299]. The elaboration of recognition or spotting techniques for polytonic Greek handwritten texts remains a challenge largely unaddressed. One exception to this rule is [300], where an OCR system for early Christian Greek documents is proposed. The target documents are written in a form of polytonic Greek, but the proposed model is fine-tuned towards the specific writing style and conventions of the given era and context, thus constraining its scope of use.

In this section, we present a method for word spotting of handwritten and machine-printed documents written in polytonic Greek. The work is largely based on the state-of-the-art model that was proposed in [31] and won the segmentation-based

track of the H-KWS 2014 word spotting competition [287]. The current work can essentially be seen as an extension of this previous work for the polytonic Greek script. As it is thoroughly discussed in Chapter 2, Section 2.4.3, Almazan et al. [31] present a learning-based model for segmentation-based word spotting. A training set is required, where each word image is to be supplied with a transcription. The word image data is used to create a Fisher Vector (FV) descriptor [301], while the transcription is used to create a histogram-based descriptor that the authors name Pyramidal Histogram of Characters (PHOC). PHOC actually records the appearance of a specific letter (or digit) in the transcription, a strategy that builds on the concept of attribute-based models used for natural image understanding in the related literature [302]. The two descriptor sets are used together to learn a projection to a new space and create a new, single descriptor based on the scaled output of a structured SVM, which yields the final, binary embedding. The output fixed-length descriptors can then simply be compared to each other and to descriptors from a test set using the Euclidean distance. The result is efficient word spotting, which when coupled with a lexicon can also be used for word recognition. In addition, due to the attribute-based structure of the model, words that do not appear in the training set can also be retrieved and/or recognized. A closely related model to [31] is proposed in [303], for text recognition, and a completely analogous transcription descriptor is used in the same spirit with the PHOCs of [31].

The transcription descriptor (PHOC) proposed in [31] is script and language-dependent, as it is comprised of a bin for each script character plus bins for the most likely language bigrams. This means that there can be no comparison between words of different scripts and/or languages and that adaptation to complex scripts such as the polytonic Greek script, that also includes diacritics of various types, is not necessarily straightforward. For this reason, we address the latter issue in this work. In this respect, we propose and compare three different ways to extend PHOC to polytonic Greek, and with it the model of [31]. The proposed scheme is tested on word spotting trials, over handwritten as well as machine printed Greek documents, outperfoming state-of-the-art learning-free methods. In the following subsections, we review the basic components of the model, the image and transcription representations, and the model mechanism. Then we concisely present the polytonic Greek script, its diacritics and particularities, and describe alternative ways to model it in the form of an extended PHOC descriptor.

4.1.1 Base model description

In the following parts of this section, we review the data representation and pipeline of the model introduced in [31]. The basic framework to represent word images is the Fisher vector description [301]. We assume that our input is already segmented at word level. For each image we extract dense SIFT descriptors [245]. A Gaussian mixture model (GMM) is trained using SIFT descriptors from all input images, and Fisher vectors are calculated for each image as a function of their SIFT description and the gradients of the GMM with respect to its parameters. This results to a fixed-length, highly discriminative representation, that can be seen as an augmented bag of visual words description that encodes higher order statistics. Fisher vectors have been used previously with success in various fields of computer vision [258, 301].

The Fisher vector representation is shown to give good results on word spotting when used as a standalone descriptor on a Query-by-Example (QBE) setting [31]. However, if the presence of a training set of word images is assumed, for which ground truth transcriptions are known, a more discriminative descriptor can be created. The proposed descriptor is based on the concept of attributes, which have recently gained increasing popularity in the machine vision literature [302, 304, 305]. Attributes are semantic properties defined over images and categories, and in effect are used as labels that denote the presence or absence of a specific feature. An image attribute may be defined for example as, "does this image contain a person?", or "is this object brown/shiny/furry?". Attributes also allow for zero-shot learning, where new, unseen instances of images or classes can be correctly processed.

In the context of document processing, attributes can be defined on word images on the basis of appearance or not of a specific letter. The set of attributes each of which is defined as a function of the presence or absence of a specific alphabet letter in the word can be aggregated to a single multivariate binary vector. This descriptor can be duplicated to answer if specific letters are found on the first or second half of the word, and so on for any partition of the word transcription into k equal parts. The subsequent aggregation of higher levels of this set of attributes makes up for the Pyramidal Histogram of Characters (PHOC) representation, where more attributes are added to capture the presence of letter bigrams. An example of a fixed-length descriptor obtained from a three-level PHOC for the transcription of the word *place* is illustrated in Figure 4.1. In [31], this scheme is applied on the 26 letters of the

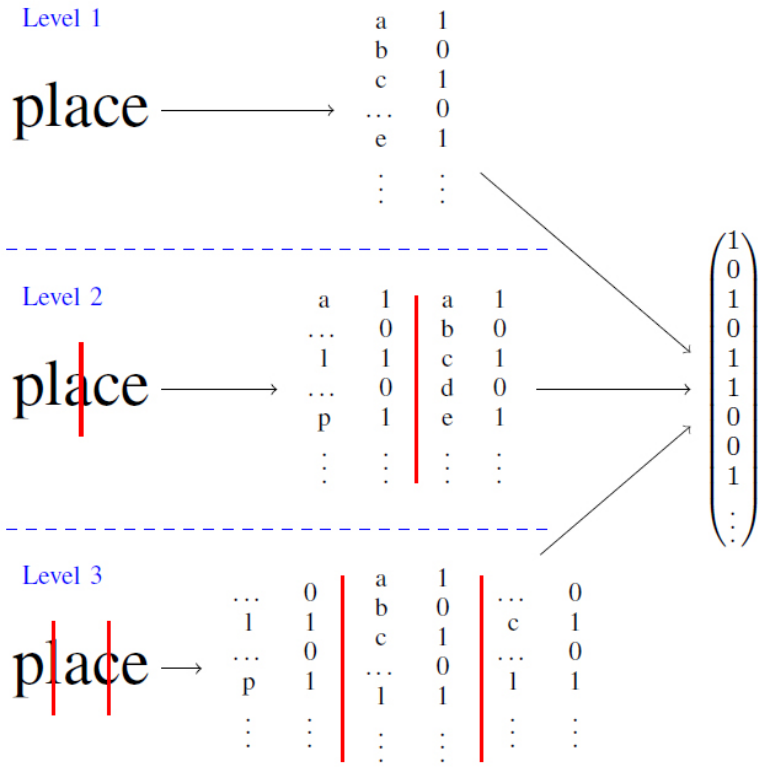


Figure 4.1: The figure exemplarily visualizes the creation of a three-level PHOC from a word string. Figure reproduced from [3].

English language plus bins for digits and the 50 most frequent bigrams (at level 2 of the pyramid) of the English language, leading to a 604-variate vector.

The Fisher vector representation of the word images and the PHOC representation of the word transcriptions are subsequently combined to create a single, more discriminative descriptor. For each variate i of the PHOC vector, a support vector machine (SVM) [306] is trained using all Fisher vectors as inputs, labelled according to attribute i. The model parameters for each SVM are saved and can be then used to calculate attribute outputs for unseen data. Such data are typically non-training set data, for which their Fisher vector can be computed since it depends on image data, while their PHOC vector cannot be computed since the transcription is unknown. The output of the structured SVM model parameters given some Fisher vector gives an output attribute vector that has the same dimensionality as the PHOC vector.

Summing things up, for every training point n we would have a Fisher vector representation f_n, a PHOC binary representation p_n, and an attribute vector representation a_n. For non-training points only the FV representation f_n and attribute vector a_n is available. The attribute vector can be used as a valid feature vector and

can be compared against other attribute vectors simply by calculating their Euclidean distance, making Query-by-Example (QBE) word spotting possible. Also, comparing the PHOC representation of a query against attribute vectors is also possible since both vectors are of the same dimensionality, allowing for Query-by-String (QBS) [31]. However, in both cases it is desirable to apply a notion of scaling or calibration over the PHOC and attribute vectors, since (a) PHOC vectors and attribute vectors are not necessarily comparable in principle, even if of the same dimensionality, (b) vector variates are not necessarily commensurate, since training of each element of the structured SVM is done independently from others, leading some of the outputs to possibly dominate over the others and (c) the inter-bin correlation is not taken into account. In the light of this, Canonical Correlation Analysis (CCA) [306] can be applied with the PHOC vectors p_n and attribute vectors a_n as its input views. In CCA, a projection is calculated for each view that maximizes correlation between vectors in the projected space. Formally, we are looking for projection vectors w_p, w_a that maximize $\arg\max_{w_p,w_a} \frac{w_p^T C_{pa} w_a}{\sqrt{w_a^T C_{aa} w_a}\sqrt{w_p^T C_{pp} w_p}}$ where C_{aa}, C_{pp}, C_{ap} are respectively sample covariance matrices between vectors in the set of attribute descriptors, vectors in the set of PHOC descriptors, and cross-covariance between the two latter sets. In practice, we are looking to combine a series of k optimal orthogonal projection vectors $w_{a1}, \cdots, w_{ak}, w_{p1}, \cdots, w_{pk}$ to project our views to a k-dimensional space. It can be shown that the required projection vectors are given by identifying vectors $w_{a1}, \cdots, w_{ak}$ with the k leading eigenvectors of matrix $C_{aa}^{-1} C_{ap} C_{pp}^{-1} C_{pa}$, and vectors $w_{p1}, \cdots, w_{pk}$ with the k leading eigenvectors of matrix $C_{pp}^{-1} C_{pa} C_{aa}^{-1} C_{ap}$. Embedding inputs to an appropriate feature space before applying CCA is equivalent to a kernel version of CCA (KCCA) and has given the best experimental results (with a random Fourier feature mapping [307], corresponding to a Gaussian kernel embedding [31]).

4.1.2 Polytonic word description

The polytonic Greek script has been introduced in the late antiquity to write the Greek language [298]. It is comprised of the standard 24 Greek letters, in upper-case and lower-case versions of the characters. Also, a number of diacritics are used. These have originally been introduced in the script with the rationale of aiding the reader with proper pronunciation of the words, while in later phases of evolution of the Greek language they have retained largely only an orthographic and etymological

value. These diacritics are the smooth and rough breathing, the accute, grave and circumflex accent, the subscript and the diaeresis. A visual example of these diacritics can be seen in figure 4.2. These diacritics can have a combined appearance on the

Diacritic type		Usage examples		
Breathings	Smooth	̓	ἀ	ἀποκάλυψη
	Rough	̔	ἁ	ἱστορία
Accents	Acute	́	ά	πάτερ
	Grave	̀	ὰ	τὸν
	Circumflex	͂	ᾶ	κλασικοῦ
Subscript		ͅ	ᾳ	χριστῷ
Diaeresis		̈	ϋ	λαϊκὸς

Figure 4.2: Polytonic Greek diacritics.

same character or on the same word, according to a certain set of grammatical rules. Further discussion of these rules is out of the scope of this paper.

4.1.3 Extending PHOC

In adapting PHOC to work with polytonic Greek, our basic problem is what would be the most efficient way to integrate the use of polytonic diacritics in the word transcription representation. To this end, we propose three possible alternatives. We dub these (i) Atonic PHOC (A-PHOC), (ii) Polytonic Header PHOC (PH-PHOC) and (iii) Mixed Bin PHOC (MB-PHOC). The difference of each one to the other is to the number and meaning of the bins used for the descriptor.

In Atonic PHOC we use 24 bins for letters at the base level of the descriptor. Each one corresponds to a single letter of the Greek alphabet. All letters can appear in two forms, capital or lowercase; capital and lowercase letters are therefore merged to the same bin, making the model case-insensitive. The letter sigma (Σ, σ) is an exception to this rule, as it can appear in one extra form besides its capital and lowercase forms, that of the final sigma (ς). This is also merged on the same bin with the other forms of the letter. We also add bins for numerical digits and bigrams. The 50 most frequent bigrams of the Greek language are added at level 2 (Figure 4.3), and the rest of the bins are iterated at levels 2, 3, 4 and 5. Level 1 histograms are excluded altogether. The most frequent bigrams are extracted by processing a corpus of 34 million Greek words (corpus "C", [308]). The total number of bins of Atonic PHOC therefore sums

αι	ια	ες	πρ	ντ
ει	ικ	ην	ρα	νο
κα	με	ης	ρι	αλ
ου	να	ισ	σε	ετ
πο	ρο	κο	ση	ιο
στ	τι	μα	τε	λε
τα	απ	νο	υν	λο
τη	ατ	οι	αρ	μο
το	εν	ον	δι	ος
αν	ερ	πα	νε	πε

Figure 4.3: The 50 most frequent bigrams of the Greek language.

up to $(2+3+4+5)*(24+10)+2*50=576$ bins. Polytonic diacritics are ignored altogether, so a letter with no diacritics uses the same bin as the same letter with any diacritics added to the letter. In the sense that the descriptor bins correspond to letters + digits + bigrams, A-PHOC can be understood as the conceptually closest to, or the most straightforward adaptation of the standard PHOC descriptor of [31] for Greek.

In Polytonic header PHOC we add information about polytonic diacritics, in the form of a number of extra bins added to what we described as Atonic PHOC. These are 7 bins, each one corresponding to the appearance or not of a diacritic in the word. We do not reiterate their use to higher levels of the PHOC pyramid. This choice makes the descriptor non-spatially aware when it comes to polytonic diacritics, at a gain of smaller descriptor length. The rationale of this choice is founded on the fact that many of the diacritics are gramatically constrained to appear at fixed positions at the beginning or the end of the word, with some rare exceptions. The Polytonic header PHOC is $576+7=583$ bins long.

With the third proposed alternative, Mixed Bin PHOC, we consider each letter, with all combinations of diacritics, as a separate case. Our histogram of base does not comprise 24 bins as in A-PHOC and PH-PHOC, but one bin for each combination of letter and polytonic diacritic. For example, α with no diacritics is assigned to a different bin than α with a smooth breathing, while α with a smooth breathing and an acute accent is assigned to a third bin, and so on. This sums up to 128 extra bins to the already existing 24 for the plain versions of letters. The size of the descriptor

totals to $(2+3+4+5)*(128+24+10)+2*50=2368$ bins.

4.1.4 Experimental results

We have run numerical experiments over databases of polytonic Greek documents, both machine-printed and handwritten. For these documents, the original text as well as a binarized version of all document pages and word-level segmentations along with ground truth transcriptions for each word is available[1]. We have available one set of handwritten documents and two sets of machine-printed documents. We refer to these sets in this work as *ST46* (also tested in Section 3.3), *Journal* and *Proceedings*. Excerpts from these sets can be seen in figures 4.4 and 4.5. Our handwritten set, *ST46*,

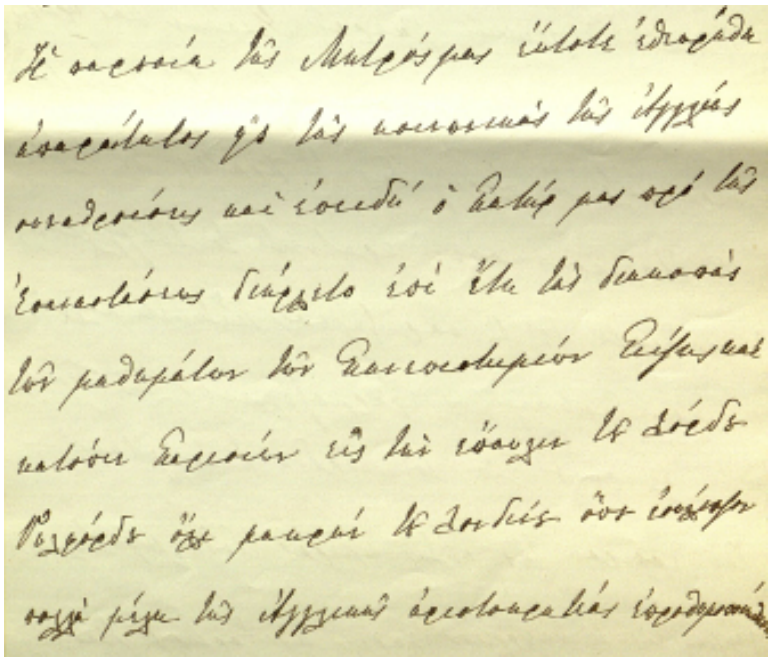
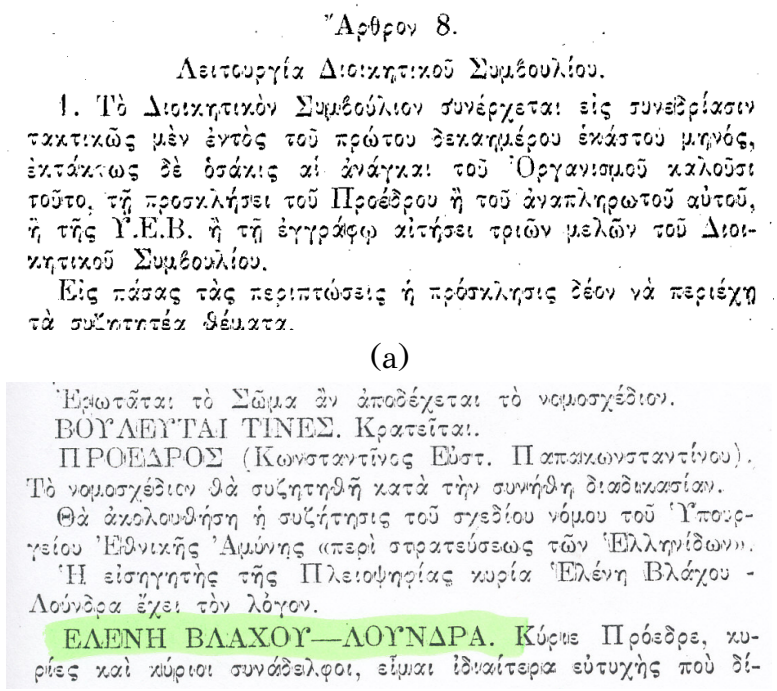


Figure 4.4: Handwritten polytonic text sample, "*ST46*". Excerpt from the memoirs of Sophia Trikoupi (1838-1916).

consists of 46 pages segmented into 4941 word images. The text is written by a single author in the late 19^{th} century, and is part of the memoirs of Sophia Trikoupi, sister of the important Greek prime minister Charilaos Trikoupis. The machine printed text *Gazette* consists of 5 pages segmented into 5004 word images. This text contains pages taken from the official journal of the Greek government describing laws and edicts, published from the mid-19^{th} to the mid-20^{th} century. The machine printed text *Proceedings* is made up of 33 pages segmented into 26783 word images and records

[1] http://www.iit.demokritos.gr/~nstam/GRPOLY-DB

various speeches delivered in the Greek parliament within almost the same time period as the dataset *Gazette*. All texts are therefore of historical value, and written in the polytonic Greek script.



"Αρθρον 8.

Λειτουργία Διοικητικοῦ Συμβουλίου.

1. Τὸ Διοικητικὸν Συμβούλιον συνέρχεται εἰς συνεδρίασιν τακτικῶς μὲν ἐντὸς τοῦ πρώτου δεκαημέρου ἑκάστου μηνός, ἐκτάκτως δὲ ὁσάκις αἱ ἀνάγκαι τοῦ Ὀργανισμοῦ καλοῦσι τοῦτο, τῇ προσκλήσει τοῦ Προέδρου ἢ τοῦ ἀναπληρωτοῦ αὐτοῦ, ἢ τῆς Υ.Ε.Β. ἢ τῇ ἐγγράφῳ αἰτήσει τριῶν μελῶν τοῦ Διοικητικοῦ Συμβουλίου.

Εἰς πάσας τὰς περιπτώσεις ἡ πρόσκλησις δέον νὰ περιέχῃ τὰ συζητητέα θέματα.

(a)

Ἐρωτᾶται τὸ Σῶμα ἂν ἀποδέχεται τὸ νομοσχέδιον.

ΒΟΥΛΕΥΤΑΙ ΤΙΝΕΣ. Κρατεῖται.

ΠΡΟΕΔΡΟΣ (Κωνσταντῖνος Εὐστ. Παπακωνσταντίνου). Τὸ νομοσχέδιον θὰ συζητηθῇ κατὰ τὴν συνήθη διαδικασίαν.

Θὰ ἀκολουθήσῃ ἡ συζήτησις τοῦ σχεδίου νόμου τοῦ Ὑπουργείου Ἐθνικῆς Ἀμύνης «περὶ στρατεύσεως τῶν Ἑλληνίδων».

Ἡ εἰσηγητὴς τῆς Πλειοψηφίας κυρία Ἑλένη Βλάχου - Λούνδρα ἔχει τὸν λόγον.

ΕΛΕΝΗ ΒΛΑΧΟΥ—ΛΟΥΝΔΡΑ. Κύριε Πρόεδρε, κυρίες καὶ κύριοι συνάδελφοι, εἶμαι ἰδιαίτερα εὐτυχὴς ποὺ δί-

(b)

Figure 4.5: Machine-printed polytonic text samples. (a) "*Gazette*". Excerpt from the official journal of the Greek government. (b) "*Proceedings*". Excerpt from the proceedings of the Greek parliament.

We have tested the proposed models in word spotting trials as well as in word recognition. In the word spotting scenario, we evaluate methods using the Mean Average Precision (MAP) benchmark [287] over a selected set of queries taken from each database. We choose the set of queries in each case on the basis of word length and appearance frequency, following [287]. For the handwritten *ST46* set we choose all words that have more than 5 letters and 4 instances as queries, for a total of 21 queries. For the machine-printed *Proceedings* we choose all words that have more than 6 letters and 5 instances as queries, for a total of 103 queries respectively. Results for all instances of each query class were averaged to calculate the total MAP in the QBE word spotting tests.

We have used two different evaluation settings in our word spotting experiments. In the first setting, we train our model using a part of the handwritten *ST46* dataset and use the rest as our test set (the test set comprises 2000 words [2]). Only queries that

[2]Indices of words used for training, test and validation respectively: $1-2000$, $2001-4000$, $4001-4941$.

are situated in the test set are used as queries, and are matched against only word images within the test set. The 21 selected query classes correspond to 50 query word image instances. In the second spotting setting, we train and test on different texts. We train our model using the full *Gazette* set and test on the *Proceedings* set. In this manner, the model capability to generalize its training over a different test set is also evaluated. While it would be interesting to use a similar setting for handwritten texts also, unfortunately only one corpus of handwritten polytonic Greek was available to us (*ST46*) at the time this work was carried out. We should also note that the 103 selected query classes of *Proceedings* correspond to 959 query word image instances.

Table 4.1: QBE word spotting results (MAP%).

Method	*ST46*	*Gazette/Proceedings*
A-PHOC	81.8%	52.5%
PH-PHOC	85.2%	56.6%
MB-PHOC	**96.6%**	**74.4%**
Adaptive zoning	60.8%	57.8%
Profiles+DTW	69.6%	62.0%

Results for our Query-by-Example (QBE) word spotting trials can be seen in table 4.1. We also show results comparing the proposed schemes against two state-of-the-art methods learning-free methods, adaptive zoning [238] and profile features with Dynamic Time Warping (DTW) [62]. Concerning the tests over *ST46*, all proposed schemes outperform considerably the learning-free methods. This is not the case with the *Gazette/Proceedings* scenario, where only MB-PHOC gives better results than the learning-free methods. This is not surprising, since the first scenario uses training and test sets taken from the same base document of the same writer. Also, performance variance has shown to be high in this latter scenario, with parts of the text corresponding to excellent results, while others corresponding to very mediocre results. We must assume that this variance is related to the similarity of the font in the given part of *Proceedings*, with the fonts used in *Journal*, which has been used for training. In all cases, PH-PHOC is better than A-PHOC, giving a difference of about 4% with A-PHOC, validating the utility of adding the polytonic header bins to the descriptor, at almost the same cost in terms of training time. MB-PHOC on the other hand outperforms all other methods in all cases with a considerable difference of about $11 - 12\%$ from the second winner.

We have run an experiment to test the robustness of MB-PHOC when used with a specific type of entering string queries. We have run Query-by-String (QBS) spotting trials on the handwritten *ST46*. We used the same criterion to select string queries as the one used for QBE. We compared two alternative scenarios for running a QBS query: the first scenario assumes the same set of queries as the ones in the QBE test, that is using the frequency/length criterion. The second scenario assumes the same queries but omitting all diacritics from them. These tests correspond to a scenario where the end-user of the document retrieval system would be unsure of the correct diacritics to use for his query. MB-PHOC has given a MAP of 81.3% in the first scenario against 79.6% in the second scenario, proving to be quite robust despite the fact that bins of the same character with and without diacritics are, implementation-wise, unrelated.

4.1.5 Discussion

This work addresses the problem of word spotting and recognition of polytonic Greek texts. We have proposed three different ways to adapt the attribute-based model of [31] for polytonic Greek, which correspond to three different transcription representations. Experiments have shown that including information about polytonic diacritics always gives better results. A-PHOC is the most naive adaptation of [31], and closest to the original PHOC descriptor in the sense that it uses bins for letters, digits and bigrams, completely disregarding polytonic diacritics. The PH-PHOC representation includes polytonic information using a short information header, which is low-cost and character-independent. PH-PHOC outperformed A-PHOC at the price of only a few extra variates added in the descriptor. The last proposal, MB-PHOC, includes feature vector variates that correspond to all valid combinations of letter and diacritic, and has been shown to be universally the most efficient choice, albeit with a high computational cost in the training phase. It has also shown to be robust in the case that a query string comprising no diacritics is used. This latter scenario may be very relevant today, if one takes into account the declining familiarity of users of modern Greek with the polytonic Greek script. Finally, to our knowledge, this work is the first attempt to extend the seminal PHOC representation for polytonic Greek text by adding extra binary encodings to this end. Similar attribute-based representations such as PHOC, are also adopted in the following section by employing

convolutional neural networks to produce compact representations from intermediate layer activations, able to adapt to unlabelled portions of challenging datasets.

4.2 Transition from shallow to deep features

In this section, we shortly present the fundamentals regarding the theoretical background on neural networks, which will be the main focus for the rest of this dissertation. Until the previous decade, the majority of document image processing techniques for text recognition and retrieval were populated by a shallow-learnt feature extraction and a machine learning step. Various types of features (see Section 2.4.1) based on gradient information (e.g. orientation) or local patterns of image pixels from interest points or zones around them and others, aimed to extract as much information as possible from an image into discriminative representations. Such handcrafted feature representations were then inserted to a task-related machine learning model (e.g. HMM) for image classification. Typically, such features were manually designed to be robust to different types of intra-class variations for the task at hand. However, since training data were not abundant, the generalization ability of these shallow representations was limited to the dataset specific inter and intra-class variabilities, usually known during training.

Nonetheless, while more and more training data, as well as computational resources became available (especially GPU parallelized capabilities for complex computations such as matrix multiplications) deep learning approaches have gained increased popularity, since they regularly outperform conventional (i.e. shallow) machine learning methods and can extract features automatically from raw data, with little or no preprocessing [309]. In other words, deep-learning methods are representation-learning tools with multiple levels of representation, obtained by composing simple but non-linear modules that can transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level. By stacking enough compositions of such transformations, very complex functions can be learned to fit or classify image data. Among the many different variants of deep learning models, convolutional neural networks (CNNs) ([310] has been the standard practice to infer reliable and accurate predictions or correctly classify input images.

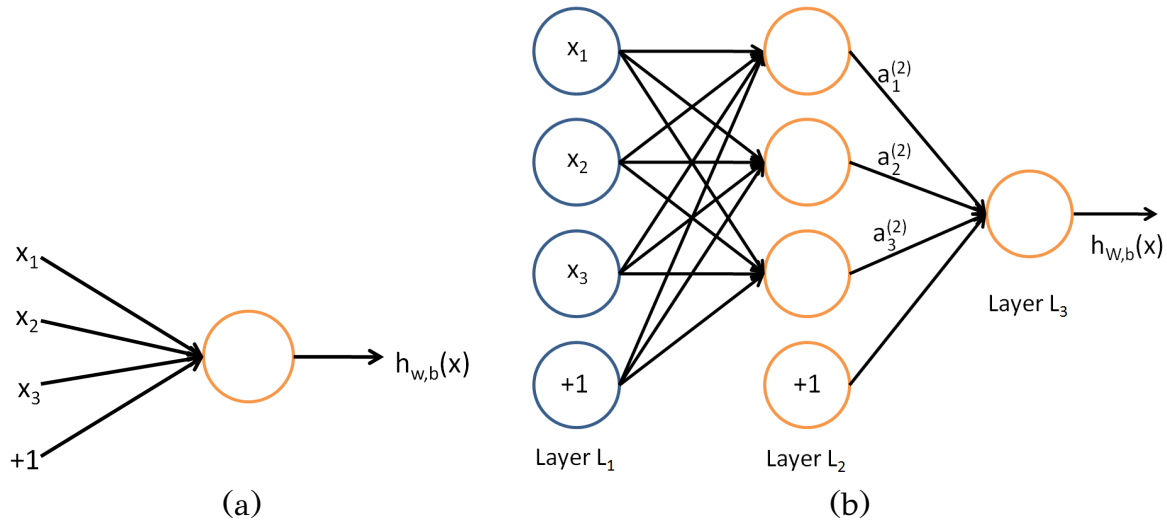


Figure 4.6: (a) A single perceptron that takes as input x_1, x_2, x_3 (and a +1 bias term), and outputs a dot product $h_{W,b}(x) = f(W^T x) = f(\sum_{i=1}^{3} W_i x_i + b)$, where $f : \Re \mapsto \Re$ is called the activation function. (b) 3-layer (input, hidden, output) feed forward neural network.

4.2.1 Overview of multi-layer neural networks

Standard feed forward neural networks (NNs) are composed of a number of layers comprising one or more artificial neurons also called perceptrons. Historically, perceptrons were developed back in the 1950s and 1960s by the scientist Frank Rosenblatt, inspired by earlier work of McCulloch and Pitts. The McCulloch-Pitts neuron was an early model of brain function. This linear model could recognize two different categories of inputs by testing whether $f(\mathbf{x}, \mathbf{w})$ is positive or negative. Of course, for the model to correspond to the desired definition of the categories, the weights needed to be set correctly. The perceptron (Figure 4.6(a)), an extension of the McCulloch-Pitts neuron, introduced the idea of trainable weights along with an appropriate training algorithm for binary classification. In order to extend the neuron function to classify non-linearly separable classes, multiple neurons are stacked together across a number of layers to build a multi-layer feed forward neural network (Figure 4.6(b)). In this framework, layers between the input and the output layer, are known as *hidden layers*.

The success of NNs as universal function approximators is attributed to a alternative process between linear transformations and non-linearities on each input. Assuming training samples $(x^{(i)}, y^{(i)})$ a neural network computes a complex, non-linear form of hypotheses $h_{W,b}(x)$, with parameters W, b that we can fit to our data. *Activation functions* are non-linear mappings following every linear transformation

of a neural network. These non-linearities greatly contribute to the representational capabilities of NNs. A typical choice of activation function is the sigmoid function:

$$f(z) = \frac{1}{1 + \exp(-z)}$$

Other popular choices include the hyperbolic tangent, or tanh, function:

$$f(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

and the Rectified Linear Unit (ReLU) activation function $f(z) = \max(0, x)$, which is not bounded or continuously differentiable and is typically preferred to alleviate cases of vanishing gradients during the backpropagation process.

Figure 4.6(b) illustrates a 3-layer feed forward NN. The circles labeled '+1' are called bias units. The leftmost layer of the network is the input layer followed by 1 hidden and the output layer which consists of one node. Let $n_l = 3$ in our example. We denote layer l as L_l, so layer L_1 is the input layer, and layer L_{n_l} the output layer. Our neural network has parameters $(W, b) = (W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)})$ where $W_{ij}^{(l)}$ denotes the parameter (or weight) associated with the connection between unit j in layer l, and unit i in layer $l + 1$. Also, $b_i^{(l)}$ is the bias associated with unit i in layer $l + 1$. Hence, we have $W^{(1)} \in \Re^{3\times3}$, and $W^{(2)} \in \Re^{1\times3}$. Note that bias units do not have inputs or connections going into them, since they always output the value $+1$. We also let s_l denote the number of nodes in layer l (not counting the bias unit).

Moreover, let $a_i^{(l)}$ be the activation of unit i in layer l. For $l = 1$, we also use $a_i^{(1)} = x_i$ to denote the i-th input. Given a fixed initialization of the parameters W, b, our NN computes a hypothesis $h_{W,b}(x)$ that outputs a real number. Specifically, the computation that this neural network represents is given by:

$$a_1^{(2)} = f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)}) \tag{4.1}$$

$$a_2^{(2)} = f(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_2^{(1)}) \tag{4.2}$$

$$a_3^{(2)} = f(W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3 + b_3^{(1)}) \tag{4.3}$$

$$h_{W,b}(x) = a_1^{(3)} = f(W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)} + b_1^{(2)}) \tag{4.4}$$

For notation simplicity, we assume $z_i^{(l)}$ to be the total weighted sum of inputs to unit i in layer l, including the bias term (e.g. $z_i^{(2)} = \sum_{j=1}^{n} W_{ij}^{(1)}x_j + b_i^{(1)}$), so that $a_i^{(l)} = f(z_i^{(l)})$. If

we extend the activation function $f(\cdot)$ to apply to vectors in an element-wise manner (i.e. $f([z_1, z_2, z_3]) = [f(z_1), f(z_2), f(z_3)]$, then the above equations are simplified as follows:

$$z^{(2)} = W^{(1)}x + b^{(1)} \tag{4.5}$$

$$a^{(2)} = f(z^{(2)}) \tag{4.6}$$

$$z^{(3)} = W^{(2)}a^{(2)} + b^{(2)} \tag{4.7}$$

$$h_{W,b}(x) = a^{(3)} = f(z^{(3)}) \tag{4.8}$$

The above procedure is known as *forward propagation*. Given $a^{(1)} = x$ to denote the values from the input layer and l's activations $a^{(l)}$, we can compute layer $l+1$'s activations $a^{(l+1)}$ as:

$$z^{(l+1)} = W^{(l)}a^{(l)} + b^{(l)} \tag{4.9}$$

$$a^{(l+1)} = f(z^{(l+1)}) \tag{4.10}$$

Concerning the training and optimization procedures of NNs the selection of the loss function is critical for the effectiveness of the trained model. For this reason, one has to take into account that the loss function should reflect the task's goal, meaning that Mean Squared Error (MSE) is more appropriate for regression whereas Cross Entropy (CE) loss for classification tasks. Moreover, the loss function needs to be differentiable to be incorporated to the upcoming gradient-based optimization scheme. Assuming an MSE loss:

$$J(W, b; x, y) = \frac{1}{2} \left\| h_{W,b}(x) - y \right\|^2$$

for a single training example (x, y) and m training samples $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ the overall loss function can be written as:

$$J(W, b) = \left[\frac{1}{m} \sum_{i=1}^{m} J(W, b; x^{(i)}, y^{(i)}) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_l - 1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} \left(W_{ji}^{(l)} \right)^2 \tag{4.11}$$

$$= \left[\frac{1}{m} \sum_{i=1}^{m} \left(\frac{1}{2} \left\| h_{W,b}(x^{(i)}) - y^{(i)} \right\|^2 \right) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_l - 1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} \left(W_{ji}^{(l)} \right)^2 \tag{4.12}$$

The first term in the definition of $J(W, b)$ is an average sum-of-squares error term. The

second term is a regularization term (known as weight decay) that tends to decrease the magnitude of the weights to avoid overfitting. Our goal is to minimize $J(W, b)$ as a function of W and b to train the neural network. We also initialize each parameter $W_{ij}^{(l)}$ and bias $b_i^{(l)}$ to a small random value, usually near zero (e.g. according to a $Gaussian(0, \sigma^2)$ distribution for a small σ), and then apply an optimization algorithm. Due to the complexity of the underlying function, there is no analytic solution for the aforementioned optimizing scheme. Therefore, iterative gradient-based algorithms are employed in order to gradually minimize the overall loss (e.g. gradient descent). Vanilla gradient descent computes the gradient score of the loss function with respect to the parameters W, b for the entire training set followed by a weight update step.

Gradient computation over the entire dataset can introduce significant computational overhead. For this reason a *Stochastic Gradient Descent* (SGD) optimization algorithm is preferred. SGD performs a parameter update for each training sample and when all training samples are processed an epoch has passed. Each pair (x, y) is fed to the SGD in a different sequence at each epoch and hence the Stochastic term in SGD. Since $J(W, b)$ is a non-convex function, gradient descent is susceptible to local optima. Therefore a mini-batch alternative of SGD is employed, where the gradients are computed over batches of samples. Particularly, one iteration of gradient descent updates the parameters W, b as follows:

$$W_{ij}^{(l)} := W_{ij}^{(l)} - \alpha \frac{\partial}{\partial W_{ij}^{(l)}} J(W, b) \tag{4.13}$$

$$b_i^{(l)} := b_i^{(l)} - \alpha \frac{\partial}{\partial b_i^{(l)}} J(W, b) \tag{4.14}$$

where α is the learning rate. We also note here that to avoid getting trapped in plateaus during optimization for a single batch, a momentum term is typically incorporated into SGD, entailing a short memory of gradients computed in previous iterations. Both learning rate and momentum are hyper-parameters, experimentally set by the user. To train NN, the key step is computing the partial derivatives above.

Backpropagation provides an efficient way to compute these partial derivatives. Initially, we first show how backpropagation is employed to obtain $\frac{\partial}{\partial W_{ij}^{(l)}} J(W, b; x, y)$ and $\frac{\partial}{\partial b_i^{(l)}} J(W, b; x, y)$, for the cost function $J(W, b; x, y)$ defined with respect to a single example (x, y). The generalization to the computation of the derivatives for the overall

cost function $J(W,b)$ can then be computed as:

$$\frac{\partial}{\partial W_{ij}^{(l)}} J(W,b) = \left[\frac{1}{m} \sum_{i=1}^{m} \frac{\partial}{\partial W_{ij}^{(l)}} J(W,b;x^{(i)},y^{(i)}) \right] + \lambda W_{ij}^{(l)} \tag{4.15}$$

$$\frac{\partial}{\partial b_{i}^{(l)}} J(W,b) = \frac{1}{m} \sum_{i=1}^{m} \frac{\partial}{\partial b_{i}^{(l)}} J(W,b;x^{(i)},y^{(i)}) \tag{4.16}$$

In other words, given a training example (x,y), forward propagation is done to compute all the activations throughout the network, including the output value of the hypothesis $h_{W,b}(x)$. Then, for each node i in layer l, an 'error term' $\delta_i^{(l)}$ that measures how much that node was 'responsible' for any errors in the output is computed. For an output node, we then measure the difference between the network's activation and the true target value, and use that to define $\delta_i^{(n_l)}$ (where layer n_l is the output layer). Regarding the hidden units, $\delta_i^{(l)}$ computation is based on a weighted average of the error terms of the nodes that use $a_i^{(l)}$ as input. In that sense, the key steps to backpropagation algorithm for the example of Figure 4.6(b) are:

1. Perform a feed forward pass, computing the activations for layers L_2, L_3, and up to the output layer L_{n_l}.

2. For each output unit i in layer n_l (the output layer), set:

$$\delta_i^{(n_l)} = \frac{\partial}{\partial z_i^{(n_l)}} \ \frac{1}{2} \left\| y - h_{W,b}(x) \right\|^2 = -(y_i - a_i^{(n_l)}) \cdot f'(z_i^{(n_l)})$$

3. For $l = n_l - 1, n_l - 2, n_l - 3, \ldots, 2$ For each node i in layer l, set

$$\delta_i^{(l)} = \left(\sum_{j=1}^{s_{l+1}} W_{ji}^{(l)} \delta_j^{(l+1)} \right) f'(z_i^{(l)})$$

4. Compute the partial derivatives:

$$\frac{\partial}{\partial W_{ij}^{(l)}} J(W,b;x,y) = a_j^{(l)} \delta_i^{(l+1)} \tag{4.17}$$

$$\frac{\partial}{\partial b_{i}^{(l)}} J(W,b;x,y) = \delta_i^{(l+1)}. \tag{4.18}$$

Using vectorial representation, where "$\bullet$" denotes the element-wise product operator

so that if $a = b \bullet c$, then $a_i = b_i c_i$ steps $2 - 4$ can be rewritten as:

$$\delta^{(n_l)} = -(y - a^{(n_l)}) \bullet f'(z^{(n_l)}) \tag{4.19}$$

$$\delta^{(l)} = \left((W^{(l)})^T \delta^{(l+1)}\right) \bullet f'(z^{(l)}) \tag{4.20}$$

$$\nabla_{W^{(l)}} J(W, b; x, y) = \delta^{(l+1)} (a^{(l)})^T, \tag{4.21}$$

$$\nabla_{b^{(l)}} J(W, b; x, y) = \delta^{(l+1)}. \tag{4.22}$$

In the pseudo-code below, where $\Delta W^{(l)}$ is a matrix (of the same dimension as $W^{(l)}$), and $\Delta b^{(l)}$ is a vector (of the same dimension as $b^{(l)}$) one iteration of batch gradient descent is as follows:

1. Set $\Delta W^{(l)} := 0$, $\Delta b^{(l)} := 0$ (matrix/vector of zeros) for all l.
2. For $i = 1$ to m
   (a) Use backpropagation to compute $\nabla_{W^{(l)}} J(W, b; x, y)$ and $\nabla_{b^{(l)}} J(W, b; x, y)$
   (b) Set $\Delta W^{(l)} := \Delta W^{(l)} + \nabla_{W^{(l)}} J(W, b; x, y)$
   (c) Set $\Delta b^{(l)} := \Delta b^{(l)} + \nabla_{b^{(l)}} J(W, b; x, y)$
3. Update the parameters:

$$W^{(l)} := W^{(l)} - \alpha \left[\left(\frac{1}{m} \Delta W^{(l)} \right) + \lambda W^{(l)} \right] \tag{4.23}$$

$$b^{(l)} := b^{(l)} - \alpha \left[\frac{1}{m} \Delta b^{(l)} \right] \tag{4.24}$$

To train our neural network, we iterate between steps of batch gradient descent to reduce the loss $J(W, b)$. Finally, an alternative popular optimization algorithm to SGD, is Adaptive Moment Estimation [311], also known as Adam Optimizer. Adam is a method that computes adaptive learning rate for each parameter, which attempts to decrease the user-defined hyper-parameters compared to SGD, while at the same time keeping an exponentially decaying average of past gradients, similar in spirit with momentum. Adam tends to be more robust when sparse targets are considered.

4.2.2 Convolutional neural networks

Nowadays, deep learning-based approaches have gained great success to overcome several limitations of document image processing problems. These deep learning-based approaches perform both feature extraction and classification tasks simultaneously through convolutional neural networks (CNNs). Such models derive a higher level of abstracted representations for the complex patterns in their deep hidden layers. CNNs can effectively handle images taking into account spatial context. Traditionally, spatial filtering was performed by convolution with handcrafted kernels, designed to capture specific patterns (e.g. edges, blobs, etc.). CNNs instead use the responses from filter banks to obtain shape information of textures within the image. The actual novelty comprises on their trainable filters that can generate discriminative feature maps, optimized with respect to the task at hand. Such an approach surpassed by far suboptimal handcrafted features, too specific to the inherent particularities of the datasets.

CNNs mostly consist of stacked convolutional layers, which perform the convolution operation: $\mathbf{Y} = \mathbf{X} * \mathbf{W}$, where $\mathbf{X}$ and $\mathbf{Y}$ are the input and output $3-D$ tensors respectively, while $\mathbf{W}$ is the $4-D$ kerneled weight tensor (4^{th} dimension is related to the depth of the feature/activation map). Specifically, using the typical spatial $(2-D)$ cross-correlation operation ($\star$), the convolution operation ($*$) is defined as:

$$\mathbf{Y}[m] = \sum_{n=1}^{C_{in}} \mathbf{X}[n] \star \mathbf{W}[m,n], m = 1, \ldots, C_{out}$$

$$(\mathbf{Y} \in \Re^{C_{out} \times H \times W}, \mathbf{X} \in \Re^{C_{in} \times H \times W}, \mathbf{W} \in \Re^{C_{in} \times C_{out} \times k_H \times k_W})$$

The spatial dimensions $H \times W$ and $k_H \times k_W$ correspond to the activation or feature map and the kernel size, respectively, while C_{in} and C_{out} correspond to the number of $2-D$ feature maps or channels on the input and on the output of the convolution.

4.2.3 CNN generic architecture

Convolutional layer is the core building block of a CNN architecture and it performs most of the computational heavy lifting. The parameters of this layer consist of a set of kernels. During forward processing, the input image is convoluted with each kernel and compute the dot product between the entries of the filter and the input

and produce a feature-map corresponding to that kernel. As a result, the network learns filters that activate when it detects some specific type of feature at some spatial position in the input.

Down-sampling is way to reduce the spatial size of the feature map representation as well as the computation and number of parameters in the network. Typical CNN architectures increase the number of channels after a group of layers. The generated feature maps contain abstract encodings of the underlying information and therefore there is no need for per pixel representation, which adds unnecessary overload. For this reason, a down-sampling process is performed. Such methods are known as pooling operations. In this respect, a *max pooling layer* is inserted to reduce the spatial size of the feature map and to control the over-fitting problem. The pooling layer typically operates with filters of size 2×2 applied with a stride of two down samples every depth slice in the input feature map by 2 along both width and height. In other words, neighborhoods of pixels are replaced by a single one, which in this case, is the one with the highest value in the neighborhood.

One main problem of CNNs is overfitting, i.e. learning the particularities of the training dataset without the ability to generalize well. To address this approaches that alter the training data in ways that change the array representation while keeping the label the same are known as *data augmentation* techniques. They are a way to artificially expand the dataset by creating augmentations such as grayscales, horizontal flips, vertical flips, random crops, color jitters, affine transformations (e.g. translations, rotations) and much more. By applying just a couple of these transformations to the training data, one can easily double or triple the number of training examples to avoid overfitting. This approach might introduce random noise into the network. Another form of such noise is the random zeroing of channels, known as dropout. *Dropout* actually allows the creation of multiple flows of information through different channels and avoids correlating a neuron/channel with a specific input sample. In simple words, this layer "drops out" a random set of activations in that layer by setting them to zero. This way it makes sure that the network is not getting too "fitted" to the training data and thus helps alleviate overfitting. An important note is that this layer is only used during training, and not during test time.

Fully connected layer considers all the features and establishes a relationship between them. This layer is performed after several convolutional and max-pooling layers and takes all neurons from the previous layers, usually couples them with

ReLU activations and finally connects them to the every single neuron in the next layer until it reaches the output layer.

4.3 Adversarial deep features for weakly supervised KWS

In this section, we first overview the baseline *PHOCNet* keyword spotting model of Sudholt et al. [4]. This model is then utilized in our proposed adversarial learning-based framework for the adaptation of deep features from a small source document collection to a much more diverse target dataset, where little annotations exist to fine-tune the original model. This way, we aim to produce compact deep representations able to adapt to variations not known during training.

4.3.1 Baseline PHOCNet model architecture

In Section 4.1 we presented a method for KWS in polytonic Greek documents adopting the attribute-based PHOC representation (Figure 4.1), initially suggested by Almazan et al. [31]. In the following, we describe the seminal work of Sudholt et al. [4] which adapts the extraction of PHOC attributes using convolutional neural networks. Their novel deep neural network, dubbed *PHOCNet*, is the baseline feature extraction model in our proposed adversarial framework to obtain deep compact representations for weakly supervised KWS.

Sudholt et al. [4] utilized the representational capabilities of a VGG-16 CNN architecture [312] to predict the PHOC representations for segmentation-based QBE and QBS KWS. A word image of arbitrary size is fed in the input of the network which predicts in the output binary attribute encodings corresponding to PHOC labels of the respective word transcription. Figure 4.7 illustrates the PHOCNet architecture, consisting of stacked convolutional layers using 3×3 convolutions followed by ReLU activations in the convolutional parts of the neural network. The authors suggest a low number of filters in the lower layers and an increasing number in the higher layers to enforce the neural network learning fewer features for smaller receptive fields and more abstract features for higher levels. An important novelty of the proposed architecture compared to standard feed-forward CNNs is a spatial pyramid max pooling (spmp) layer [229] placed between the last convolutional layer and fully connected layers. The output of the convolutional layers depends on the size of

the input image (or feature maps of previous layers). However, fully connected layers coming after the convolutional layers expect a standard size representation. This layer allows variable-sized inputs by partitioning the feature maps of the last convolutional layer in a 3-level pyramidal fashion where each partition is then max-pooled into a fixed-sized vectorial representation.

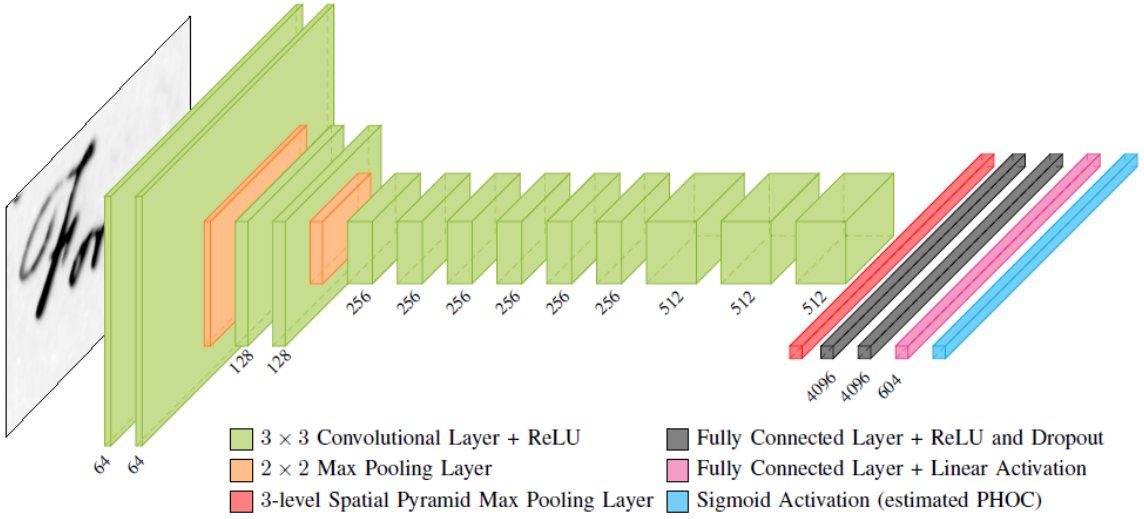


Figure 4.7: PHOCNet architecture. Green corresponds to the convolutional layers, orange to the max pooling layers and black to the fully connected layers. Red color is used to highlight the spatial pyramid max pooling layer while blue color represents the sigmoid activation layer. The number of filters for each convolutional layer is shown underneath as are the number of neurons for the fully connected layers. The number of neurons in the last layer is equal to the size of the PHOC. Convolutional layers use stride 1 and apply 1 pixel padding. Pooling layers use stride 2. Figure reproduced from [4] and is better seen in color.

Finally, the common task for a CNN is a 1 out of k classification. Usually, this is achieved by applying the softmax activation to the output of the CNN which produces pseudo-probabilities for each class. In order to predict binary estimations, corresponding to PHOC labels, the network output is replaced with a sigmoid activation instead, where the binary cross entropy (BCE) loss (equation 4.25) is selected:

$$E(\mathbf{a}, \hat{\mathbf{a}}) = -\frac{1}{n}\sum_{i=1}^{n}[a_i \log \hat{a}_i + (1-a_i)\log(1-\hat{a}_i)] \tag{4.25}$$

Assuming $\hat{a}_i$ denotes the pseudo probability for attribute i being present in the word image, the network is trained by applying BCE loss to the output $\hat{a}$ of the CNN and backpropagate the error gradient. Here, a_i represents the annotation label for attribute i extracted from the PHOC label. Figure 4.8 depicts the difference in the output for

each corresponding activation.

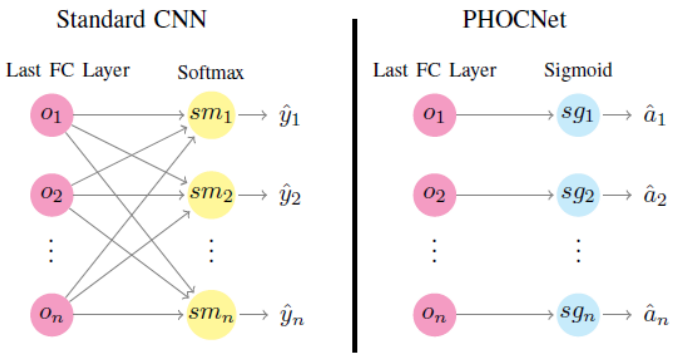


Figure 4.8: Visualization of a standard softmax output vs PHOCNet output. Figure reproduced from [4].

4.3.2 Need for feature adaptation

Handwritten keyword word spotting and recognition systems have evolved significantly over the years. Modern deep-learning based approaches [34,100] seek to be able to robustly spot handwritten text by learning local invariant patterns across diverse handwriting styles that are consistent in individual characters, words and scripts. These deep learning algorithms require vast amounts of data to train models that are robust to practical applications for handwritten image retrieval. While a considerable amount of annotated document collections are nowadays available for scripts like Latin, a large number of scripts with larger vocabularies have limited labelled data to be used for training efficient keyword spotting systems, able to retrieve query instances, across various languages. Furthermore, the process of creating such large amounts of annotated data can prove expensive and labour-intensive. Techniques like model pre-training and data augmentation may be successful in reducing the required labelled data. However, the need for producing more transferable features that can adapt to target collections where little manual annotations are present still persists. Not to mention that segmentation-free KWS in historical manuscripts is a field where annotations can be hard to get since the annotator is expected to have knowledge of the particular document collection so as to provide an accurate transcription of the underlying words and their corresponding ground-truth bounding boxes to be used for training.

In the case of scripts where abundant training data is not available, deep neural

networks (DNNs) do not always solve the task at hand, usually overfitting on the training set and thus generalizing poorly during inference. A typical example confirming this statement can be observed in the work of Retsinas et al. [74], wherein the authors experimentally evaluate intermediate layer representations along with the standard *PHOCNet* output of the CNN model from Sudholt et al. [4], for QBE word-based handwritten KWS. Specifically, *PHOCNet* is trained on the George Washington collection (GW), which is rather limited with regard to writing-style (few authors) and inter-class (limited unique labels) variability. Then the model is directly evaluated on the IAM dataset under a number of different configurations with respect to various proposed feature embeddings and similarity measures. The same experiment is then performed with the model pre-trained on IAM and evaluated on GW datasets. In the former case, the pre-trained PHOCNet model (on GW), is highly overfitted on the particularities of limited writing styles and therefore, its reported MAP index when tested on the IAM benchmark drops down to 2.8%, from the optimal $77,6\%$, in case it had been entirely trained on the IAM dataset. This is to be expected, since the number of unseen writing-styles as well as unknown word image classes of the IAM collection differs significantly from those of the GW document collection. Regarding the opposite experiment of training the model on the rather larger IAM collection and then evaluating it on the GW benchmark, the authors report a slighter drop on the performance, almost 20% less from the corresponding optimal MAP, which can be attributed to the intra-class (different writing styles from multiple authors for the same word) variances which are dominant for the IAM dataset, as opposed to the GW collection.

4.3.3 Motivation for intermediate data augmentation

Popular methods such as *data augmentation* allow models to use the existing data more effectively, while batch-normalization [313] and dropout [314] prevent overfitting. Augmentation strategies employing spatial transformations, such as random translations, flips, rotations and addition of Gaussian noise to input samples are often used to extend the original dataset [310] and prove to be beneficial for not only limited but also large datasets. The existing literature on deep learning-based KWS [4, 34, 165] heavily relies on adequate available datasets along with image augmentation techniques to increase the number of training samples prior to feature extraction. Such

transformations, however, do not always incorporate the variations in writing style and the complex word structures, due to the unconstrained sequential nature of handwriting. Since there exists a wide range of possible variabilities in handwritten images, training by generating deformed examples through such generic means might not be sufficient as the network easily adapts to such practices. Models need to become robust to uncommon deformations in inputs by learning to effectively utilize more informative invariances, whereas it is not always practical to manually pick "hard" samples to achieve high generalization capacity, as it is the case in [315].

In addition to image data augmentation, *transfer-learning* describes another approach to reduce the amount or even the need of training data. It has been shown that data from another domain can be used to efficiently pre-train a model. For instance, in [165], a large synthetic dataset which resembles handwriting from rendered computer fonts is used for training word spotting models. The resulting dataset is used for pre-training a network that is then fine tuned on samples from the target domain. However, as shown in the work of Gurjar et al. [159], training a model exclusively on synthetic data might not allow for state of the art performances just by directly applying fine-tuned inference. Actually, the amount of training data necessary to achieve competitive results can be reduced significantly. For this reason, contrary to the abovementioned approaches, we propose an adversarial learning-based framework for word segmentation-based handwritten KWS, when label resources of target datasets are limited, with respect to the number of samples we can afford for fine-tuning a model, which can be initially trained on a small as well as low (intra-class) variance dataset, such as the GW collection, comprising of 4860 word images in total.

To that end, motivated by the recent success of adversarial learning for a number of tasks such as cross-domain image translation [316] or domain adaptation [317] we propose a generative adversarial learning-based component to augment the word images in the feature space using spatial transformations, in the same spirit with Spatial Transformer Networks [5]. This model is dubbed Feature Map Adversarial Deformation (FMAD) which is injected in between intermediate layer representations of the original deep neural network for word spotting. Actually, the proposed model is placed right before the spatial pyramid pooling layer (cf. Section 4.3.4) of the PHOCNet model. Its purpose is to deform the last convolutional layer feature maps, which are obtained from arbitrary-size input images, before they get pooled into fixed-length representations which will be then transformed into final PHOC labels.

This way, we aim to alleviate overfitting for the PHOCNet when it is trained on non-discriminative features from datasets with low intra-class variability while being able to generalize well to real-world testing data where little annotations exist with much more and possibly rare deformations. Both the adversarial generator (FMAD) and the original KWS model are trained in the same end-to-end framework. In fact, the adversarial generator intends to produce 'difficult' examples, whereas the KWS network attempts to learn robustness to difficult variations, which gradually becomes better over time. Our work is similar to [318] in terms of distorting the feature space in an adversarial manner using the Thin Plate Spline (TPS) transformation, which renders the KWS model robust to unseen writing-styles and inter-class variances. However, we propose a simpler, affine transformation of the feature space from the low resource labelled set, since the pre-trained baseline KWS model already understands to search for word images, visually similar to a query. Moreover, we also investigate the adaptation ability of deep features extracted directly after the adversarial process, when such features are pooled to fixed-length representations. Specifically, in a similar concept with [74], we employ the representation accrued right after the spatial pyramid max pooling operation from the last convolutional layer of the standard PHOCNet model, dubbed as *spmp*. The final descriptor from this layer, is further reduced to a lower dimensional space, using principal components analysis. Numerical experiments validate the effectiveness of deep features for KWS compared to simply using the network output. Finally, contrary to [318] our method requires far less annotations from the target dataset to achieve competitive performance with similar transfer-learning approaches [159].

Hence, our main contribution lies on a supervised representation for KWS in low resource scripts using adversarial learning to augment the initial data in high-dimensional convolutional feature space. In this respect, deformations induced by the adversarial generator enlarge the baseline KWS model's capacity to learn from unseen writing-style variabilities even when the amount of labelled data is limited. We also experimentally confirm that the KWS performance can improve at a certain degree with respect to the state of the art, especially when a model pre-trained in small datasets such as the GW collection, is available in our arsenal to be fine-tuned for the task at hand, in much larger manuscript collections where only a few training samples are available.

4.3.4 Adversarial deep feature adaptation

Most of the recently proposed deep leaning-based KWS systems assume a pre-processing step where data augmentation is applied in terms of jittering the original images to extend the respective datasets. This is not only true for small datasets, but also in the case of large collections, such as the *HW-SYNTH/IIIT-HWS* synthetic dataset utilized by Krishnan et al. [32,144] wherein artificially generated images are augmented along with the target dataset images to train the employed model architecture. Of note are also approaches which aim to learn representative word image embeddings by focusing on difficult training examples [153] or on specific parts of the dataset that yield the most reliable results [283] according to some confidence measure for KWS.

Even more promising results have been introduced by the recent bloom of adversarial learning and Generative Adversarial Networks (GANs) [319] which basically employ generative modeling to augment data in limited datasets. In our proposed method, we use a similar strategy to render KWS models such as the PHOCNet, robust and invariant to a large number of variations present in handwritten datasets without abundant training data. To achieve this, we propose a NN component, namely FMAD, which is based on Spatial Transformer Networks [5]. This network is trained to regress a set of parameters aiming to deform the features learned by the original PHOCNet model, thereby encouraging it to adapt to challenging examples and uncommon variations not known during training.

For clarity of description we will define the modified PHOCNet as the *Spotting Model* (SM) whose input is an image I. The corresponding loss function of the PHOCNet model is the binary cross entropy loss, in line with [4] which is dubbed L_{sm}. Initially, we split the SM network into three parts, namely SM_A, SM_B and SM_{PHOC}, where the first part is the last convolutional layer of the PHOCNet model, the second part is the spatial pyramid max pooling layer ($spmp$) and the last part is the final PHOC label prediction of the KWS model. Assuming $\mathbf{U}$ is the output feature map of SM_A, i.e. $\mathbf{U} = SM_A(I)$, the distorted feature map, $\mathbf{V}$, from FMAD is afterwards passed through SM_B and SM_{PHOC} for final label prediction. While the complete SM network is trained with the objective to accurately estimate the output PHOC, the FMAD network tries to deform intermediate layer features so that SM will not predict the correct labels easily. This way, SM is pushed to adapt to more discriminative features

and intra-class variances in the target datasets during testing. The deformation model FMAD and the spotting model SM compete in this adversarial two player game in an alternative manner during training. We use solely SM (PHOCNet) model for the inference task. The proposed combined system is summarized in Figure 4.9.

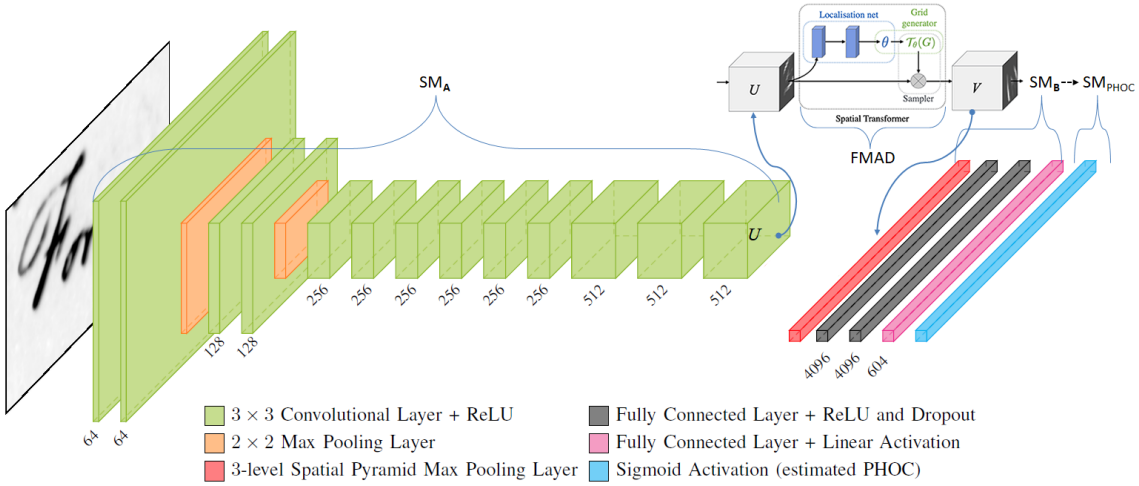


Figure 4.9: Proposed architecture of the PHOCNet model [4] combined with the Feature Map Adversarial Deformation (FMAD) component between the last convolutional and spatial pyramid max pooling layers. FMAD comprises the Localisation Network, Grid Generator and the Sampler which compose the Spatial Transformer [5].

Apart from the output layer of the PHOCNet model which predicts the final PHOC label of a query image to be compared with dataset PHOC descriptors for QBE, word-based KWS, the use of intermediate layers can also be employed to produce features from layer activations. Such features corresponding to one or more hidden layers of the network are typically flattened to final word descriptors and they are very common in the literature as *deep features* [32, 34, 121, 150, 180, 235]. In several computer vision problems, deep features have often led to superior performance over the standard use of the employed network. This can be attributed to their ability to capture more abstract patterns of the input space. In this work, we explore the transferability of deep features accrued from *spmp* layer which comes right after the distorted (by FMAD) feature map $\mathbf{V}$, further reduced to its pca-equivalent vector, when they are tested against challenging intra-class deformations that are not present during training.

4.3.5 Feature map spatial transformation

Our proposed FMAD model is inspired by the spatial transformer networks [5]. A Spatial Transformer is an image model block that explicitly allows the spatial manip-

ulation of data within a CNN by actively spatially transform feature maps, conditional on the feature map itself, without any extra training supervision or modification to the optimisation process. Unlike pooling layers, where the receptive fields are fixed and local, the spatial transformer module is a dynamic mechanism that can actively spatially distort a feature map by producing an appropriate transformation for each input sample. The transformation is then performed on the entire feature map (non-locally) and can include scaling, cropping, rotations, as well as non-rigid deformations. In this work, we prefer employing only affine transformations to the last convolutional layer feature maps, prior to the spatial pyramid max pooling operation. Since the employed PHOCNet model can be already pre-trained to predict PHOC labels from a small source dataset, we consider that random affine distortions produced by the FMAD network in the feature space will be enough to render PHOCNet model's generalization ability harder for KWS, when tested on low resource target datasets.

As it can be seen in Figure 4.9, the Spatial Transformer is composed of three modules, the localisation network, the grid generator and the sampler. Particulary, the input feature map $\mathbf{U} \in \mathcal{R}^{H \times W \times C}$, being that from the last convolutional layer of the spotting model (SM) with height H, width W and C channels, is passed to the localisation network $\mathcal{A}$ which through a number of hidden layers regresses the parameters θ of the transformation $\mathbf{T}_\theta$. To perform a warping of the input feature map, each output "pixel" (i.e. an element on feature map $\mathbf{V}$) is computed by applying a sampling kernel centered at a particular location in the input feature map $\mathbf{U}$. The output "pixels" are defined to lie on a regular grid $G = G_i$ of elements $G_i = (x_i^t, y_i^t)$ forming the output feature map $V \in \mathcal{R}^{H' \times W' \times C}$, where H' and W' are the height and width of the grid and C is the same number of channels as in the input feature map. As stated above, we use an affine transformation defined by the following equation:

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \mathbf{T}_\theta\left(\mathbf{G}_i\right) = \mathcal{A}_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \tag{4.26}$$

where (x_i^t, y_i^t) the target coordinates of the regular grid in $\mathbf{V}$, (x_i^s, y_i^s) the source coordinates in $\mathbf{U}$ that define the sample points and $\mathcal{A}_\theta$ the affine transformation matrix computed by the localization network $\mathcal{A}$. We use height and width normalised coordinates, such that $-1 \leq x_i^t, y_i^t \leq 1$ when within the spatial bounds of the output and

$-1 \leq x_i^s, y_i^s \leq 1$ when within the spatial bounds of the input.

The grid generator iterates over the regular grid G of the output/target image and uses the transformation $\mathbf{T}_\theta(\mathbf{G})$ to calculate the corresponding (usually non-integer) sample positions in the input/source image. This way it produces a sampling grid $\mathbf{T}_\theta(\mathbf{G})$. The sampler then iterates over the entries of the sampling grid and extracts the corresponding pixel values from the input map $\mathbf{U}$ using bilinear interpolation to produce the distorted sampled output feature map $\mathbf{V}$. In other words, each (x_i^s, y_i^s) coordinate in $\mathbf{T}_\theta(\mathbf{G})$ defines the spatial location in the input where a sampling kernel is applied to get the value at a particular pixel in the output $\mathbf{V}$. For a bilinear sampling kernel we derive the output feature map elements using equation 4.27:

$$\mathbf{V}_i^c = \sum_n^H \sum_m^W U_{nm}^c \max(0, 1 - |x_i^s - m|) \max(0, 1 - |y_i^s - n|) \; \forall i \in [1 \ldots H'W'] \forall c \in [1 \ldots C] \tag{4.27}$$

where U_{nm}^c is the value at location (n, m) in channel c of the input, and V_i^c is the output value for pixel i at location (x_i^t, y_i^t) in channel c. Note that the sampling is done identically for each channel of the input, so every channel is transformed in an identical way to preserve spatial consistency between channels.

Finally, since the matrix operations for grid-generation and affine transformation are differentiable (cf. [5] for more technical details on gradient computations), the FMAD component can back-propagate gradients as well. In fact, the parameters predicted adversarially by the localization network $\mathcal{A}$ denote 3 control points pointing to coordinates in $\mathbf{U}$ by regressing over their x, y values, which are normalized to lie within $[-1, 1]$ as stated above. The network represented by $\mathcal{A}$ includes a final fully connected (fc) layer predicting 2×3 normalized coordinate values and it is fitted with the $\tanh(\cdot)$ activation function.

4.3.6 Proposed adversarial learning scheme

In traditional adversarial learning frameworks, such as GANs [319], the generator $\mathbf{G}$ takes a noise vector $\mathbf{z}$ from a distribution $P_{noise}(z)$ as an input and outputs an image $G(z)$. The discriminator D takes either $G(z)$ or a real image x with a distribution $P_{data}(x)$ as an input and outputs the classification probability. The generator G is learned to maximize the probability of D making a mistake. Using the standard cross

entropy loss, the objective loss function for training G and D is defined as follows:

$$\mathcal{L} = \min_{G} \max_{D} \mathbb{E}_{x \sim P_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim P_{noise}(z)}[\log(1 - D(G(z)))] \tag{4.28}$$

where the G and D networks are trained simultaneously. The training encourages G to fit $P_{data}\,(x)$ so that D will not be able to discriminate x from $G\,(z)$. While in standard GANs, the Generator G learns a mapping of z from the noise distribution $P_{noise}\,(z)$ to the data distribution $P_{data}\,(x)$ over data x, in our proposed adversarial scheme, G (i.e. FMAD) learns a mapping of $\mathbf{U}$ from the distribution of original features $P_{origin}\,(\mathbf{U})$ to the space of distorted features $P_{distorted}\,(\mathbf{V})$ as described below for the Spotting Model (SM):

$$\mathcal{L}_{SM} = \min_{G} \max_{D} \mathbb{E}_{\mathbf{V} \sim P_{distorted}(\mathbf{V})}[\log D(\mathbf{V})] + \mathbb{E}_{\mathbf{U} \sim P_{origin}(\mathbf{U})}[\log(1 - D(G(\mathbf{U})))] \tag{4.29}$$

This way we train our proposed FMAD component, comprising only of the parameters of the localization network $\mathcal{A}$ (both grid generator and sampler are parameterless) similarly with G and SM network similarly to D, alternatively, in an adversarial fashion. Initially, $\mathcal{A}$ generates random deformations, but with the progress of adversarial learning, it learns strategies to jitter the intermediate feature space so that it becomes hard to spot words for SM. In addition, contrary to classical GANs which can perform in a fully unsupervised manner, we aim to train the discriminating network, i.e. the PHOCNet SM model in a supervised way using labelled samples, while encouraging it to successfully retrieve word images in spite of adversarial deformations present in them. We note here that $\mathbf{U}$ is deformed uniformly so that the distorted feature map $\mathbf{V}$ has the same dimensions ($H \times W \times C$).

During inference, the output $\mathbf{U}$ of component SM_A is further passed through SM_B and SM_{PHOC} to predict a PHOC label $\hat{a}$ for a word image I. Assuming the ground-truth label for a PHOC is a, our previously proposed loss L_{sm} of the SM model can be defined as:

$$L_{sm} = \text{BCE}_{loss}(a, \hat{a}) \tag{4.30}$$

where BCE is the sigmoid binary cross-entropy (BCE) loss defined in equation 4.25 (Section 4.3.1).

While training the complete model, we have two different components, namely, the KWS network SM and the localisation network $\mathcal{A}$ with corresponding parameters

θ_{SM} and $\theta_{\mathcal{A}}$ respectively. In line with [318], a forward pass of a single iteration flows as follows: $I \rightarrow SM_A(\cdot) \rightarrow FMAD(\cdot) \rightarrow SM_B(\cdot) \rightarrow SM_{PHOC}(\cdot) \rightarrow \hat{a}$, where $FMAD(\cdot)$ represents the complete deformation operation including parameter prediction by $\mathcal{A}$ as well as parameterless grid-generation and sampling operations. $\mathcal{A}$ tries to learn feature deforming ways through SM_A so that the PHOC label prediction should fail. Hence, we obtain $\theta_{\mathcal{A}}$ by maximizing the loss function L_{sm}. On the contrary, the θ_{SM} is optimized to minimize the loss L_{sm}:

$$\theta_{\mathcal{A}} = \arg\max_{\theta_{\mathcal{A}}} L_{sm} \tag{4.31}$$

$$\theta_{SM} = \arg\min_{\theta_{SM}} L_{sm} \tag{4.32}$$

4.3.7 Datasets, protocol and implementation details

We briefly mention the manuscript collections which are used to assess the performance of the proposed adversarial framework for example-based KWS. The first collection is the well known *George Washington* (GW) database [2]. It consists of 20 pages of correspondences from George Washington and his associates which contain a total of 4860 word images (after excluding poorly segmented images from 4894 words). As there is no official partition in training and test images, we use the Almazan et al. [31] protocol and perform a fourfold cross validation. This setup is adopted by the majority of the recent learning-based methods [4, 32, 69, 100, 165], since it is split to training and testing partitions. We use the exact same partitions as were used in [31], consisting of 3 folds for validation and training purposes (15 pages) as well as 1 fold (5 pages) for testing. All words in test set are used as queries in a leave-one-out style, resulting in 3645 words for training and 1215 test words, respectively. The query image is removed from the test set (since it is top-ranked according to similarity with itself) and queries without relevant occurrences are discarded.

The second dataset is the *IAM* handwritten database [290]. It is made up of 115320 words written by 657 writers. We use the official partition available for writer independent text line recognition which splits the database in 6161 lines for training, 1840 for validation and 1861 for testing. One of the main challenges of this data set is that each writer contributed to only one partition (either training, validation or test). In accordance with [31], we exclude the official stop words from the query set but

keep them as distractors in the test dataset which contains 13752 words. All words appearing more than once are used as queries. Again, the query image is removed from the test set.

Before examining the proposed method's capacity to accurately retrieve query instances in low resource target datasets when corresponding samples are not essentially used during training, we first set up the circumstances under which the model has to be pre-trained on a standard benchmark containing little intra-class variances, namely, the GW collection which is typically written by a single writer. Actually, during the experimental part, we noticed that it is imperative to initially train the PHOCNet (SM) network for a certain number of iterations so that it can build a basic model able to predict PHOC labels reflecting handwritten word attributes [31]. In case we directly perform the joint training of FMAD with SM models, it seems that the deformation network dominates over the Spotting Model thus hampering its ability to produce meaningful representations.

For this reason, we first train the baseline PHOCNet for 10000 iterations without the FMAD. Thereafter, we include the latter to fulfill its adversarial objective of deforming the intermediate convolutional feature maps. We use 250 continuous iterations to train the parameter localization network $\mathcal{A}$ alone for better initialization. The Localisation Network is composed of three convolutional layers with stride 2 and filter size 3×3 followed by 2 fully-connected layers, in order to predict 6 parameter values using $\tanh(\cdot)$ activation. We use a batch size of 10. Following the earlier initialization, both the PHOCNet (SM) network and FMAD are trained for a total of 100000 iterations alternatively. We use Adam optimizer for both SM and FMAD networks. Nevertheless, we set the learning rate for SM to 10^{-4}, divided by 10 after 60000 iterations (i.e. 10^{-5} from $60k$ to $100k$ iterations), while for the Localisation Network of FMAD module is 10^{-3}. PHOCNet consists of 13 convolutional layers followed by the $spmp$ layer and 3 fully connected layers where sigmoid activation is used to predict the final PHOC label (see Figure 4.9). In line with [4], the momentum is set to 0.9 and the weight decay is $5 \cdot 10^{-5}$. As mentioned before, the FMAD component is inserted after the last convolutional layer, right before it gets max pooled by the $spmp$ layer. Both FMAD and PHOCNet models are implemented using PyTorch library [320]. Experiments are carried out using a single Nvidia GeForce RTX 2080 Super GPU.

We use the Mean Average Precision (MAP) index to evaluate the performance for QBE word segmentation-based KWS. In line with [4], we augment the training set

partition using affine transformations of the input images, to create 500000 total word samples. When using the proposed adversarial deformation model, we do not perform image level augmentation at all. As we can see from Figure 4.10, the proposed combined adversarial deformation network (SM-FMAD) consistently surpasses the baseline PHOCNet model's [4] performance until $40k$ iterations. This is reasonable since the proposed model processes the whole training set many times in far less iterations than the original PHOCNet model which uses image level augmentations. This way SM-FMAD produces adversarial transformations early enough so as to succeed high performance. In subsequent iterations however, MAP converges to similar values for both image and feature space augmentation alternatives. In fact, for several cases after $40k$ iterations, it is even worse for the feature space augmentation approach. This indicates that the proposed model is starting to learn trivial distortions of the convolutional feature space, which lead to potentially overfitted representations for the low intra-class variabilities of the GW dataset images, thereby not further improving the retrieval performance.

4.3.8 Experiments on weakly annotated datasets

Following the above observation (see Figure 4.10) for the GW dataset with respect to just a slight improvement obtained for a specific number of continuous iterations after injecting the FMAD component in the original SM model to augment the feature space (as opposed to standard image level augmentation), we consider an alterative approach, similar in spirit with transfer learning. In this concept, we achieve efficient KWS in low resource target scripts, where only a few annotations exist for fine-tuning the proposed framework. To this end, we pre-train the standard PHOCNet in the GW dataset along with the proposed feature space augmentation model for $40k$ iterations and then evaluate it on the much more challenging IAM database, concerning the number of distinct word-classes as well as significant intra-class variances. As it was highlighted in Section 4.3.2, when the pre-trained baseline PHOCNet model is directly evaluated on the IAM dataset it is rather unable to generalize for writing styles and query instances not seen during training. In fact, its reported MAP from the work of Retsinas et al. [74] drops to 2.8%, from the optimal $77,6\%$, if it had been entirely trained on the complete IAM dataset.

For this reason, we suggest a transfer learning approach where we freeze the

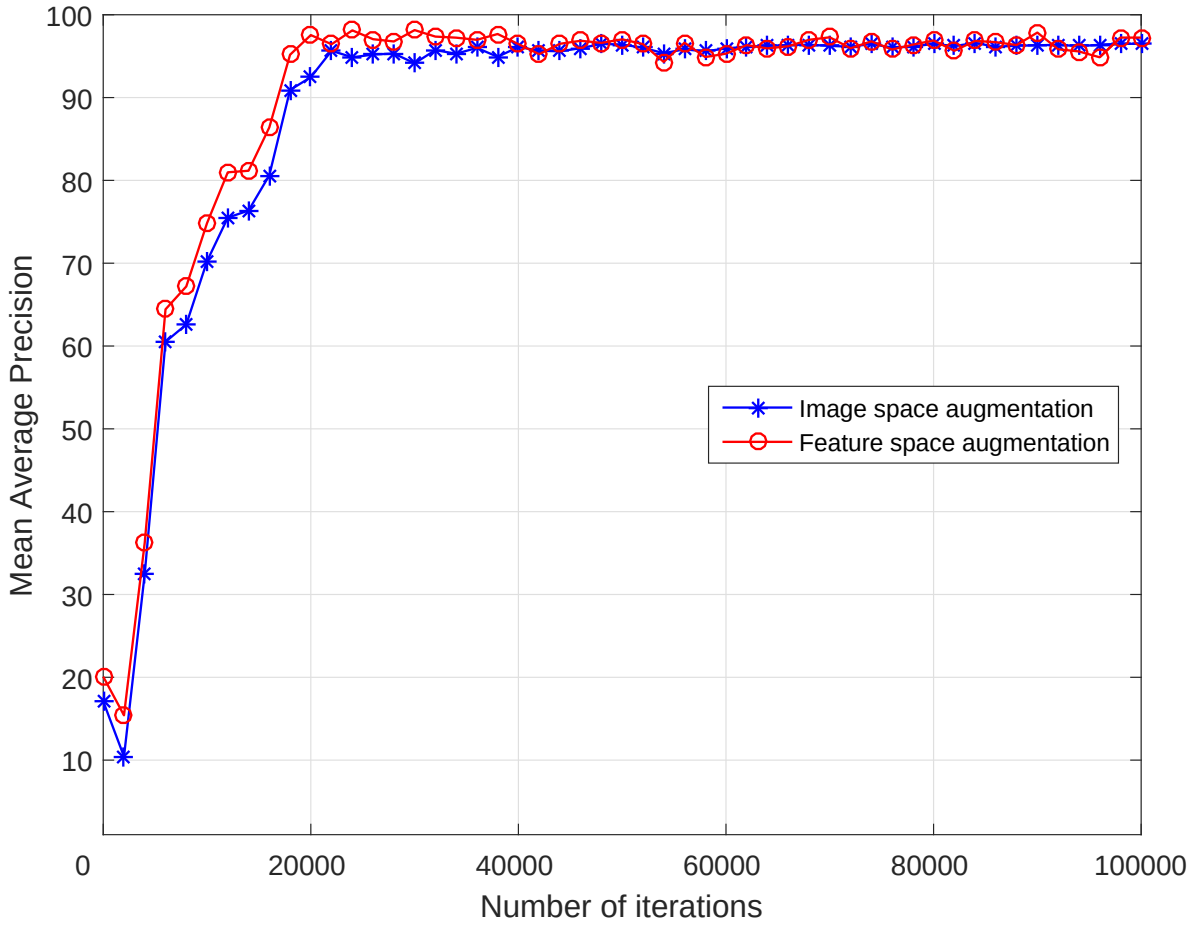


Figure 4.10: Baseline PHOCNet (SM) model is first initialized on GW for 10000 iterations. FMAD is initialized for 250 iterations. Then both networks are trained alternatively for $100k$ iterations in our joint SM-FMAD framework. The figure illustrates MAP obtained (every $2k$ iterations) in the official GW test set, after augmenting the original dataset at image space according to [4] (standard SM model, blue) and feature space (red) using our approach, respectively. The Figure is better seen in color.

weights of the pre-trained (on GW) adversarial complete model (SM-FMAD) up until the penultimate convolutional layer (i.e. the one right before the last convolutional layer) which we consider that already suffice to understand generic handwritten input images, leading to meaningful PHOC labels in the network's output. Afterwards, we train the rest of the proposed adversarial unified model (see Figure 4.9) using a limited number of training samples from the IAM dataset. In other words, we train the weights starting from the last convolutional layer of SM_A, followed by the localization network $\mathcal{A}$, the $spmp$ layer of SM_B which pools the distorted feature maps into fixed-length representations, the fully connected and ReLU layers, until the final PHOC label prediction layer of SM_{PHOC}. We employ the same hyper-parameters described in the previous section to fine-tune the model for 60000 alternating iterations (10^{-4} learning rate) between the SM and FMAD networks, in the proposed adversarial fashion without any prior initialization for each separate network (since both networks are already jointly pre-trained on the GW dataset following the earlier initialization protocol).

To confirm the validity of our assumption which deems augmentation of the feature space far more meaningful against the image space augmentation counterpart, especially when the target (IAM) dataset contains significant intra-class writing style variability, as well as much more distinct word-classes than the source (GW) dataset which is used for pre-training the proposed model, we experiment with two different setups. The first one comprises the proposed model pre-trained on GW as described previously. The second follows the standard PHOCNet model, pre-trained on GW without the FMAD component under the same conditions, which also assumes image level augmentation of the given training samples in a similar fashion with [4]. Figure 4.11 depicts the performance gain obtained when using the proposed adversarial feature space augmentation framework.

In addition to the proposed transfer learning task, we also investigate the adaptation ability of deep features extracted from intermediate layer activations, instead of directly employing the SM_{PHOC} network's output. To that end, after having fine-tuned the complete model on the target IAM dataset, we extract deep features obtained from the activations of the $spmp$ hidden layer, when a specific word image input is provided. In particular, we produce corresponding word descriptors based on deep features for each word image (given all dataset test images along with the query). However, the original dimensions of the $spmp$ layer representation are very

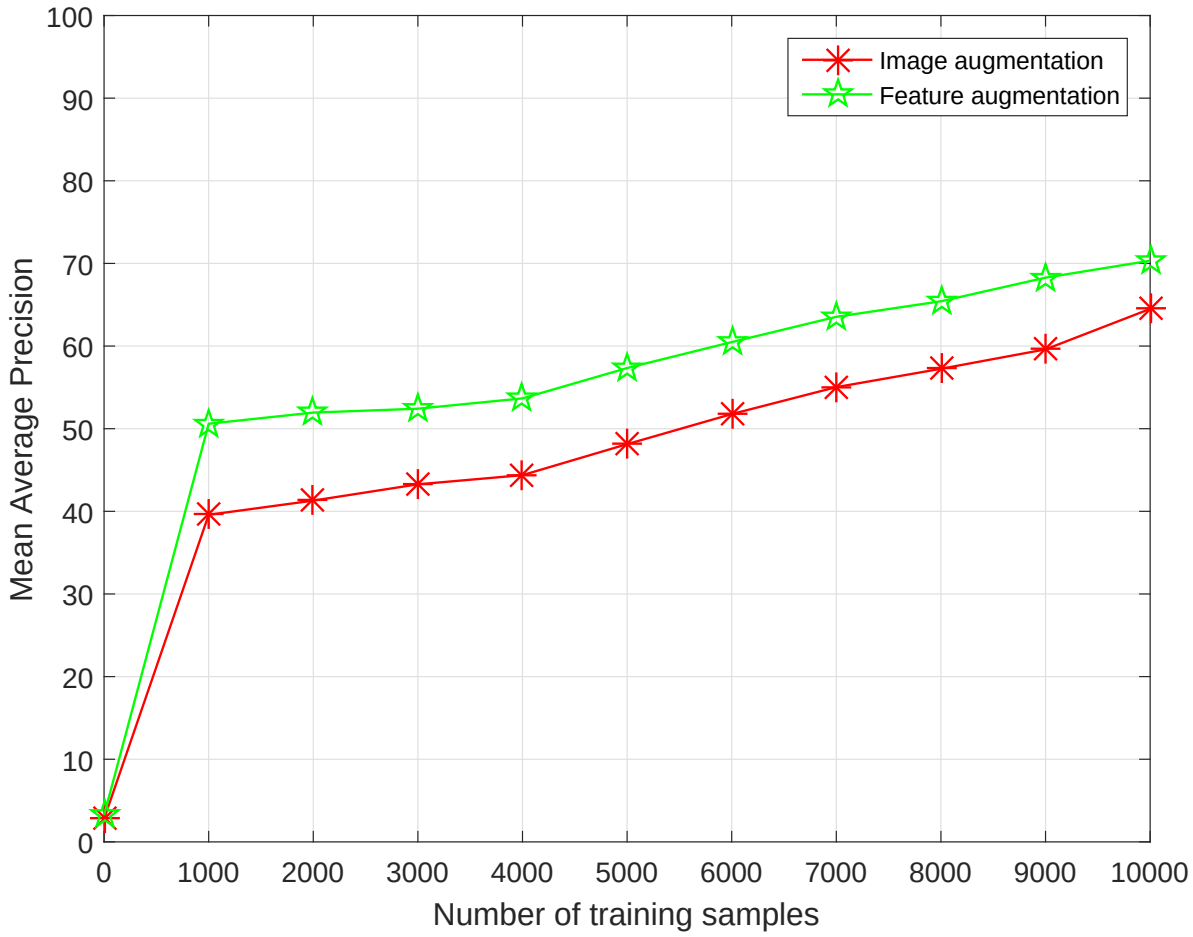


Figure 4.11: Mean Average Precision for QBE-based KWS in the standard IAM test set for different numbers of training samples. Green color corresponds to the proposed model, pre-trained on GW for $40k$ iterations and then fine-tuned on IAM train sets, whereas red color represents the model's performance when augmentation is performed in image space using random affine transformations of the input [4]. Both models are trained for $60k$ iterations. The Figure is better seen in color.

high (10752) since they depend on the number of neurons of that layer. Therefore, we suggest a dimensionality reduction using principal component analysis to reduce this dimensionality to 400 for the final employed word image representation. Finally, word descriptors (deep feature vectors) between dataset and query images are compared with the widely-used cosine distance [100].

We report numerical results in Table 4.2 for different sets of training samples containing 100, 250, 500, and 1000 word images, randomly drawn from the IAM benchmark. This is done to make the method comparable with a similar, in setup, state-of-the-art framework of Gurjar et al. [159] for QBE KWS. As we can see from Table 4.2, the performance of our proposed method is close in comparison with the more generic transfer learning approach of [159] in the low annotation-resource scenario while requiring far less training samples during network pre-training, as we discuss in the following section. With respect to the experimental setup, to be directly in line with [159] when no added training data are used at all from the IAM dataset (0 samples), since we cannot fine-tune the model, we test the ability of the proposed model to directly predict transferable representations as follows: we use each image in the test partitions of the IAM dataset as a query and infer PHOC labels from our proposed adversarial model, which is only pre-trained on the GW dataset as mentioned above. Similarly, deep features are also extracted for both query and the remaining images for KWS.

Table 4.2: MAP for QBE KWS on different amounts of training data for the IAM.

Method	**Training Set Size**				
	0	100	250	500	1000
Gurjar et al. [159]	26.21	38.45	43.78	52.41	55.39
Proposed adversarial SM_{PHOC} output	3.22	22.38	34.71	45.67	50.60
Proposed adversarial deep features ($spmp$)	12.55	27.44	38.23	49.81	54.69

4.3.9 Discussion

From the reported numerical results of Table 4.2 we can understand that in case no training data are used at all, the model fails to generalize in both methods of the proposed adversarial framework, which is to be expected, since it is regularly

overfitted to the particularities of the GW dataset. Nevertheless, as more and more training samples are used from the target IAM dataset to fine-tune the proposed adversarial networks, the retrieval performance is close to the state-of-the-art one [159]. This is actually true for 1000 training samples. Of significant importance are also the proposed deep features extracted from $spmp$ layer of the proposed model whose MAP is on par with the state of the art. This is indicative of their ability to encode discriminative invariances in the feature space, thus leading to more transferable word image representations than the standard network output, especially when annotated target images are very few.

We should note here that the comparison of our method with [159], is adversely performed, as far as the number of training samples used for pre-training the original PHOCNet model is concerned. Actually, Gurjar et al. [159] pre-train the whole model in one million synthetic images (HW-SYNTH) for 80000 iterations, in contrast with the 4860 word images that are employed in our proposed feature space augmentation scheme. More specifically, the HW-SYNTH dataset is a collection of synthetically generated word images [144]. All 26 letters of the Latin alphabet and the digits were used to generate these images. Each image of the 1 million dataset words belongs to 10000 distinct word-classes obtained from the Hunspell dictionary. The authors of [144] utilize 100 publicly available fonts for randomly generating each word image. The images are rendered by varying the inter character space, stroke width, and the mean foreground and background pixel distributions followed by Gaussian filter smoothing. Each word class is rendered using all letters in the lower and capital case, as well as only the first letter capitalized. In fact, the training set used by the model of Gurjar et al. [159] includes 750000 images. On the contrary, our model employs only 3645 words for training which falls by far behind in terms of abundantly different writing styles and distinct word-classes present in the synthetic dataset. Moreover, the training subsets used to test our method against Gurjar et al. [159] in Table 4.2 are augmented by the latter work in image space following the process described in [4] so as to increase the number of images accordingly.

However, the results obtained by the proposed adversarially learnt deep features, are promising enough to provide us insight about the required number of training samples, as well as their degree of intra-class variability that is needed to be available during training, in order to produce transferable representations, able to adapt to much larger document collections. The objective of our proposed KWS pipeline is to

learn a robust model invariant to different types of deformation in handwritten data. Due to the unconstrained nature of handwriting, it is not possible to include every potential variation in the training dataset, even when image level augmentations are used. Instead of trying to learn the invariance only from available supervised data, thereby failing to generalize on unseen irregularities and deformations, we learn a robust model that can generalize well on unseen deformations which are absent in weakly supervised datasets.

In our forthcoming plans, we will also test the improvements achieved when employing much larger collections such as the synthetic HW-SYNTH (III-TK) dataset, for which the transcription information is straightforward to obtain from the various computer fonts, aiming to resemble variable handwritten inputs for the same word-class. In this spirit, we expect the transformations of the feature space to yield much more informative representations, able to discriminate unseen query instances during training, on much more challenging manuscript collections for which annotations are hard to obtain.

Chapter 5

Adversarial learning for text spotting in natural images

In this Chapter, we present a variation of adversarial learning to solve a slightly more abstract problem to KWS, namely, text spotting in natural images. To this end, we introduce and discuss Quaternion generative adversarial networks, a variant of generative adversarial networks that uses quaternion-valued inputs, weights and intermediate network representations. Quaternionic representation has the advantage of treating cross-channel information carried by multichannel signals (e.g. color images) holistically, while quaternionic convolution has been shown to be less resource-demanding. Standard convolutional and deconvolutional layers are replaced by their quaternionic variants, in both generator and discriminator nets, while activations and loss functions are adapted accordingly. We have successfully tested the model on the

task of detecting Byzantine inscriptions in the wild, where the proposed model is on par with a vanilla conditional generative adversarial network, but is significantly less expensive in terms of model size (requires $4\times$ less parameters). We note here that this work is a joint effort carried out during the last year of the thesis.

5.1 Problem at hand

Digitization and online accessibility in cultural institutions such as museums, libraries and archives can achieve much greater visibility to the public when the digitized content is organized in meaningful entities. For example, text in natural images generally conveys rich semantic information about the scene and the enclosed objects, which might be of great use in real scenarios where the digitized raw image information is not directly exploitable for searching and browsing.

One of the most prominent trends in content-based image retrieval applications is to discriminate which part of the image includes useful information, as opposed to background objects, occlusion and task-irrelevant parts [321]. Such tasks may concern image analysis, understanding, indexing or classification of objects according to some inherent property. In the particular case of text understanding applications, the main goal is to retrieve regions that contain solely textual cues, either as holistic region information or as textual parts at line, word or even character level.

Text detection is a challenging task due to the variety of text appearance, the unconstrained locations of text within the natural image, degradations of text components over hundreds of years, as well as the complexity of each scene. To address these challenges, standard convolutional neural networks (CNNs) have been the main attraction over the last five years for text detection [38,39]. However, the effectiveness of CNNs is usually limited by the homogeneity of the dataset images used for training as well as the particular loss function that is to be minimized for the specific task at hand. Generative adversarial networks (GANs) [319] offer a more flexible framework that can in effect learn the appropriate loss function to satisfy the task at hand. GANs setup an adversarial learning paradigm where the game dynamics of two players-networks lead to a model that, in its convolutional variant is the state of the art in numerous vision tasks today.

With the current work, we discuss a novel neural network variant that brings to-

Figure 5.1: Sample images of our inscription dataset.

gether the concepts of GANs with that of Quaternionic convolution and deconvolution, and build a model that can effectively perform text detection in a context where the content of interest is "donor" inscriptions found in byzantine monuments [322, 323] (see Figures 5.1,5.2). Quaternions are a form of non-real numbers that can be understood as 4-dimensional generalizations of complex numbers, with one real part and three independent imaginary parts. The use of non-real numbers as neuron and parameter values has been proposed as far as 1991, with an adaptation of backpropagation for complex numbers [324]. Similar developments for quaternions have followed suit [325]. The more recently proposed quaternion convolutional neural networks (QCNNs) [326, 327] a special form of convolution that makes use of quaternion product rules, effectively treating multichannel information holistically. Furthermore, QCNNs have been shown to be much more economical (i.e. less resource-demanding) networks than their non-quaternionic counterparts, with four times smaller parameter set size [326, 328]. Motivated by the promising properties of quaternionic neural networks, we propose using quaternionic operations with adversarial networks. In particular, the contribution of this work concerns the introduction of quaternionic convolution to the conditional convolutional generative adversarial paradigm, where we replace the encoder-decoder architecture with quaternionic layer versions, and otherwise adapt network architecture where necessary. In order to setup our numerical and qualitative experiments, we test the proposed model for inscription localization in

the wild. In terms of numerical results, we conclude that the proposed model attains comparable evaluation scores to its non-quaternionic counterpart, while being less resource demanding.

Figure 5.2: Example ground-truth annotation for selected samples from our inscription dataset.

The remainder of this chapter is structured as follows. In Section 5.2, we review related work. In Section 5.3, we present the basics of quaternion algebra, followed by quaternionic convolution and its use with convolutional neural networks in Section 5.4. In section 5.5, we discuss the proposed model whereas in Section 5.6, we show the dataset, task and numerical experiments which validate the method's performance

in terms of parameter size, compared to conventional approaches for text spotting in the wild.

5.2 Related work

The automatic detection of text can be categorized into two main families. The first direction includes identifying text of scanned document images whereas the second contains text captured by natural images (indoor or outdoor images with text of more complex shapes, cuneiform tablet images or inscriptions) which is further subject to various geometric distortions, illumination and environmental conditions. The latter category is also known as text detection in the wild or scene text detection [329]. In the first category, text detection in printed documents is usually tackled by OCR techniques [330], while in handwritten document images, the problem is formulated as a keyword search in a segmentation free scenario [40].

In the text detection-in-the-wild paradigm, conditions such as wide variety of colors and fonts, orientations and languages are present. Moreover, scene elements might have similar appearance to text components, and finally, images may be distorted with blurriness, or contain degradations due to low camera resolution during digitization process, capturing angle and partial occlusions. Under such adverse situations deep learning based methods have shown great effectiveness in detecting text. Recent deep approaches for text detection in the wild, inspired by object detection frameworks, can be categorized into *bounding-box regression based*, *segmentation-based* and *hybrid* approaches.

Bounding-box regression based methods for text detection [331] regard text as an object of interest and attempt to predict the candidate bounding boxes directly. Segmentation-based methods in [332] enforce text detection as a semantic segmentation task, aiming to classify text regions at pixel level and then obtain bounding boxes containing text during post-processing. Hybrid methods [333] rely on a segmentation step to predict score maps of text which in turn yield text bounding-boxes as a result of regression. Similarly to [332], our method localizes text in a holistic manner, by performing text detection as a semantic segmentation problem to produce global pixel-wise prediction maps.

While CNNs are at the top of the dominant problem-solvers in image recognition

tasks, such as the text detection in the wild case explored in this work, traditional real-valued CNNs encode local relations of the input features from R,G,B channels of each pixel along with structural relations composed by groups of pixels, independently. On the contrary, our proposed quaternionic conditional adversarial network treats text detection as a semantic segmentation task, performing at input RGB channels holistically with the use of quaternions, so as to obtain a binary output of white text pixels. To our knowledge, GANs have not been used yet for text detection in the wild [329], at least at the time this work was carried out. Moreover, the quaterionic representation of the conditional variant of the generative adversarial networks is a first attempt to discriminate a text region by its non-text counterpart with less computational load.

Recent works on quaternion CNNs [326, 327] indicate that the lower number of parameters required for the multidimensional representation of a single pixel in R,G,B channels leads to better image classification results than traditional CNNs. The authors claim that the performance boost is also due to the specific quaternion algebra. Such a boost is further explored in [334], where instead of a real-valued dot product, a vector product operation allows quaternion CNNs to capture internal latent relations by sharing quaternion weights during the product operation, and in turn by creating relations within the product's elements.

5.3 Elements of Quaternions

Quaternions, introduced in the mid-$19th$ century, form an algebraic structure known as a skew-field, that is characterized by all the properties of a field except that of multiplication commutativity. We denote the quaternion skew-field as $\mathbb{H}$. Quaternions are four-dimensional, in the sense of $\mathbb{H}$ being isomorphic to $\mathbb{R}^4$, and each $q \in \mathbb{H}$ can be written as:

$$q = a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}, \tag{5.1}$$

where $a, b, c, d \in \mathbb{R}$ and $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ are independent imaginary units. Hence, analogous to the representation of complex numbers, which bear one real and one imaginary part, quaternions have one real and three independent imaginary parts. Alternatively, quaternions can be represented as the sum of a scalar (their real part) and a three-

dimensional vector (their imaginary part). Formally we can write:

$$q = S(q) + V(q), \tag{5.2}$$

where $S(q) = a$ and $V(q) = b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}$. Further generalizing the related $\boldsymbol{i}^2 = -1$ formula for complex numbers, for quaternions we have:

$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1,$$

$$\boldsymbol{ij} = -\boldsymbol{ji} = \boldsymbol{k}, \boldsymbol{jk} = -\boldsymbol{kj} = \boldsymbol{i}, \boldsymbol{ki} = -\boldsymbol{ik} = \boldsymbol{j}. \tag{5.3}$$

Quaternion conjugacy is defined as:

$$\bar{q} = a - b\boldsymbol{i} - c\boldsymbol{j} - d\boldsymbol{k}, \tag{5.4}$$

while quaternion magnitude is defined as:

$$|q| = \sqrt{q\bar{q}} = \sqrt{\bar{q}q} = \sqrt{a^2 + b^2 + c^2 + d^2}. \tag{5.5}$$

As a consequence of the properties of a skew-field and eq. (5.3), we have the following multiplication rule for quaternions:

$$\begin{aligned} pq =& (a_p a_q - b_p b_q - c_p c_q - d_p d_q) + && (5.6)\\ & (a_p b_q + b_p a_q + c_p d_q - d_p c_q)\boldsymbol{i} + && (5.7)\\ & (a_p c_q - b_p d_q + c_p a_q + d_p b_q)\boldsymbol{j} + && (5.8)\\ & (a_p d_q + b_p c_q - c_p b_q + d_p a_q)\boldsymbol{k}, && (5.9) \end{aligned}$$

where $p = a_p + b_p\boldsymbol{i} + c_p\boldsymbol{j} + d_p\boldsymbol{k}$ and $q = a_q + b_q\boldsymbol{i} + c_q\boldsymbol{j} + d_q\boldsymbol{k}$. Following the notation of eq. (5.2), we can write the above rule also as:

$$pq = S(p)S(q) - V(p) \cdot V(q) + S(p)V(q) + S(q)V(p) + V(p) \times V(q), \tag{5.10}$$

where $\cdot$ and $\times$ denote the dot and cross product respectively. Interestingly, note that when p, q are pure (i.e., they have zero respective real parts), the quaternion product boils down to a cross product. The above formulae are also referred to as a Hamilton product [328] in the literature.

5.4 Quaternionic convolutional neural networks

Quaternionic convolutional neural networks have been recently introduced as variants of the widely used convolutional neural networks that have quaternionic model parameters, inputs, activations, pre-activations and outputs. This creates issues with a number of network components and concepts, including the definition of convolution, whether standard activation functions are usable and how back-propagation is handled. In theory, multiple proposals for a convolution operation could be considered [335]. Two quaternionic extensions of convolution have been succesfully employed in two recent works [326,327]. In all cases, a quaternionic kernel $g \in \mathbb{H}^{K \times K}$ acts on an input feature map $f \in \mathbb{H}^{M \times N}$ to generate the output map $g \in \mathbb{H}^{M+K-1 \times N+K-1}$. The two extensions differ in the choice of elementary operation used in each case.

In [327], a convolution extension that is based on the equation used to apply quaternionic rotation is employed (i.e. $w \rightarrow qw\bar{q}$, where q is a pure unit quaternion). In particular, they define quaternionic convolution $g = f * w$ as:

$$g_{kk'} = \sum_{l=1}^{K} \sum_{l'=1}^{K} s_{ll'}^{-1} w_{ll'} f_{(k+l)(k'+l')} \bar{w}_{ll'}, \tag{5.11}$$

where $f = [f_{ij}]$ denotes the input feature map, $w = [w_{ij}]$ is the convolution kernel, and $s_{ll'} = |w_{ll'}|$.

In [326], which is the convolution version that we test in this work, convolution is more simply defined as:

$$g_{kk'} = \sum_{l=1}^{K} \sum_{l'=1}^{K} w_{ll'} f_{(k+l)(k'+l')}, \tag{5.12}$$

where the definition is analogous to standard convolution, with the difference that elements are quaternionic and the kernel multiplies the signal from the left on each summation term. Strided convolution, deconvolution and padding are also defined analogously to real-valued convolution.

Concerning activation functions, the most straightforward option is to use standard activations that are used in real-valued networks (e.g. sigmoid, ReLU, etc.) and use them on each quaternion real and imaginary part separately, as if they were separate real channels. This type of activations are referred to in the literature as split-activation functions. In this work, we use split-activation versions of leaky Rectified linear unit

(ReLU) and the sigmoid function.

5.5 Proposed model

The proposed model is made up of the well-known pair of the generator and discriminator networks that are used in standard GANs. The vanilla (non-conditional) GAN objective function [336] is, in its original form as follows:

$$L_{\text{GAN}} = E_x \log D(x) + E_z \log(1 - D(G(z))), \tag{5.13}$$

where $G(\cdot)$ and $D(\cdot)$ denote the generator and discriminator network respectively. x are samples of the training set, while z denotes random noise that is used as input to the generator. For the discriminator, the aim is to maximize this function, while for the generator the aim is to minimize it. These competing terms result in a two-player game, of which we require to obtain a parameter set that would correspond to a Nash equilibrium.

We employ a supervised variant that is referred to as a conditional GAN (cGAN) architecture, made popular with the pix2pix model [337]. Formally, the objective function is written as:

$$L_{\text{cGAN}} = E_x[logD(y)] + E_x[log(1 - D(G(x)))]] + \lambda E_{x,y}[\|y - G(x)\|_1] \tag{5.14}$$

where we can comment on a number of differences comparing with the standard GAN formula of eq. (5.13). In particular, no random noise variable z exists, and on the contrary the generator takes as input a sample x to produce a target y. In that sense, the cGAN is supervised; a cGAN learns a mapping from input x to target y. Also, a second L_1 regularizing term is employed, penalizing the difference of the produced $G(x)$ to the desired target y. A regularizing term λ controls trade-off of the two terms.

In this work, x is a quaternion-valued image, formally $x \in \mathbb{H}^{H \times W}$, where H and W are image height and width in pixels. In particular x is assumed to be a dataset image, and estimate $G(x)$ is a detection heatmap that ranges in $[0, 1]$. A pixel value of $G(x)$ that is close to 1 means a high probability that this pixel is part of a text inscription, and vice-versa. Ground truth target y is binary, with values in $\{0, 1\}$ (see Figure 5.2). In order to form each quaternion-valued input x, we assign each of its

three colour channels (Red, Green, Blue) to each of the quaternion imaginary axes. Hence, we assign $Red \rightarrow \boldsymbol{i}$, $Green \rightarrow \boldsymbol{j}$, $Blue \rightarrow \boldsymbol{k}$. The real part is left to be equal to zero, or in other words all values of x are pure quaternions.

The generator is constructed as a U-net-like model [202] with two symmetric groups of layers, arranged to an encoder and a decoder part. The encoder is composed of strided quaternionic convolutional layers that produce quaternionic feature maps of progressively lower resolution in comparison to the original input image size. The decoder mirrors the encoder layers, by using a quaternionic deconvolutional layer for each forward convolution layer of the encoder, and upsampling feature maps progressively to the original resolution. Furthermore, U-net-like skip connections connect corresponding encoder - decoder layers. We use 4 quaternionic convolutional layers for the encoder, and 4 quaternionic deconvolutional layers for the decoder. Dropout layers top layers 5 and 6. Convolutions are strided with stride=2, kernel sizes=4×4, and output number of channels equal to $16, 32, 64, 64$ for layers 1 to 4 respectively. Deconvolutional layers share the same characteristics, mirroring the encoder architecture, with added skip connections. All layers, except the final layer, are topped by split-activation leaky ReLU functions with parameter = 0.2. These act on each quaternionic pixel value x as:

$$lReLU_q(x) = lReLU_r(x_a) + lReLU_r(x_b)\boldsymbol{i} + lReLU_r(x_c)\boldsymbol{j} + lReLU_r(x_d)\boldsymbol{k} \tag{5.15}$$

where $lReLU_r$ is the well-known real-valued leaky ReLU function and we assume $x = x_a + x_b\boldsymbol{i} + x_c\boldsymbol{j} + x_d\boldsymbol{k}$. The generator implements a mapping $\mathbb{H}^{H \times W} \rightarrow [0, 1]^{H \times W}$, from a quaternion-valued image to a real image. All intermediate layers map quaternion-valued feature maps again to quaternion-valued feature maps, save for the final activation. We define the final activation simply as the sum:

$$qsum(x) = x_a + x_b + x_c + x_d. \tag{5.16}$$

which ensures a real-valued output.

The discriminator is constructed as a cascade of strided quaternionic convolutions, with strides and size identical to those used for the generator encoder. It implements a mapping $\mathbb{H}^{H \times W} \rightarrow [0, 1]$, where the output represents the degree in which the network believes that the input is fake or genuine. Inputs to the discriminator are constructed as concatenations of color inscription images to the estimated target. In particular,

we map $Detection\ estimate \rightarrow real\ part$, $Red \rightarrow \boldsymbol{i}$, $Green \rightarrow \boldsymbol{j}$, $Blue \rightarrow \boldsymbol{k}$. As the output is real while the input is quaternionic, in the final layer we use the activation of eq. (5.16), before applying a sigmoid function on top of it. The discriminator is made up of 6 quaternionic convolutional layers. Output number of channels equal to $16, 32, 64, 64, 128, 1$ respectively for the 6 convolutional layers.

Note also that the setup of the generator and discirminator is such that inputs and outputs can be of variable size. Indeed, the generator is a fully convolutional network, with parameters and layers that are independent of input and feature map size. The discriminator leads to feature maps that are reduced to a single probability value, again regardless of the input image and annotation dimensions.

5.6 Experimental results

5.6.1 Dataset

The dataset is comprised of a total of 67 images containing inscriptions written in Greek, and found in Byzantine churches and monasteries in the region of Epirus, located in Northwestern Greece [322, 323]. Our inscriptions are donor's inscriptions, typically made up of a few lines of text and containing information about who donated funds and other resources required to build the monument where the inscription is located. The photographed images were captured with a Samsung GT-I9505 and a Nikon Coolpix L810 camera. All images were then resized so as their width was at most 1024 pixels, keeping their aspect ratio fixed. We have chosen to partition the set to a training and test set according to a $80\%/20\%$ rule, which resulted to training and test sets of 55 and 12 images respectively.

5.6.2 Experiments

Concerning training, we have used the Adam optimizer with parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$. No data augmentation is used. The trade-off parameter λ was set to 10 and base learning rates were set to 10^{-4} for the discriminator and 5×10^{-4} for the generator. Furthermore, a learning rate scheduling strategy was used, where learning rate is divided by 10 for both networks if test binary cross-entropy deteriorates continuously for 2 consecutive epochs. Batch size was set to 1, as our model was setup to accept

inputs of variant size.

We have used two evaluation measures: a) binary cross-entropy (BCE) of the test images and b) Intersection over Union (IoU). Test BCE is applied in an analogous manner to the corresponding loss component discussed in section 5.5, and effectively tests for correct per-pixel binary classification. The IoU measure is applied after computing a binarized version of the estimate detection map, with a threshold of 0.5 (Pascal VOC challenge [338]). Subsequently, IoU is computed between this binarized estimate and the ground truth.

In Figure 5.3, we show plots for the generator and discriminator loss calculated per training iteration, and test BCE loss and IoU calculated as an average over test images and at the end of each epoch. QGAN and VGAN are compared, as well as two considered model sizes. Standard size corresponds to the model described previously in section 5.5. Large size corresponds to QGAN and VGAN models that have double the number of channels per convolutional or deconvolutional layer.

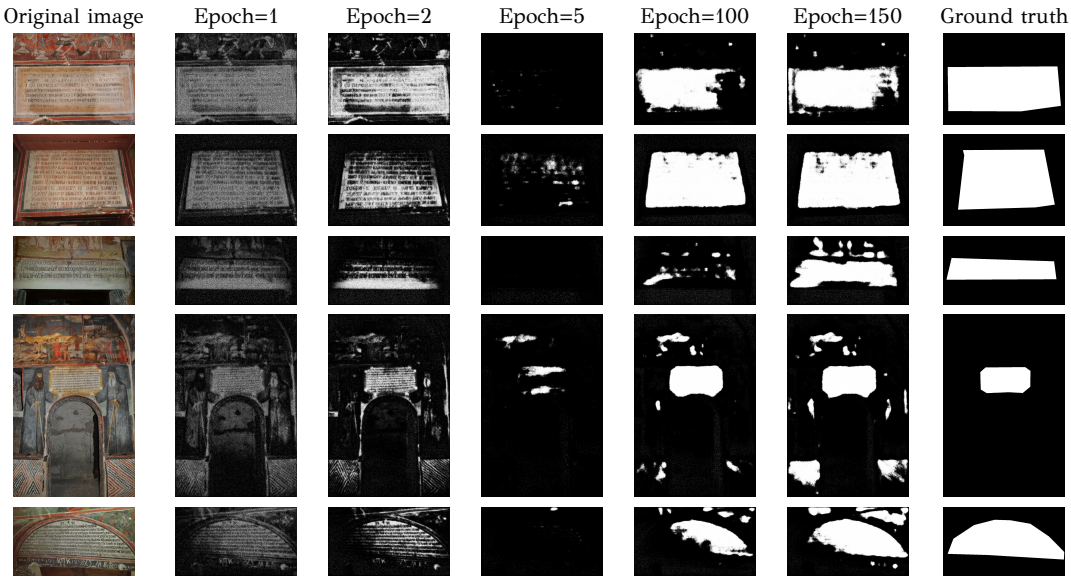


Figure 5.4: Sample results of proposed model on test images.

We show results for test images as a function of current epoch training in Figure 5.4.

We compare each Quaternionic GAN model with its vanilla (non- quaternionic) counterpart, by considering a network with the same amount of neurons. For each quaternionic neuron of the QGAN, we need to create four neurons for the corresponding VGAN, due to the isomorphism between $\mathbb{H}$ and $\Re^4$. As shown before [328], computation of the quaternionic (Hamilton) product and consequently quaternionic

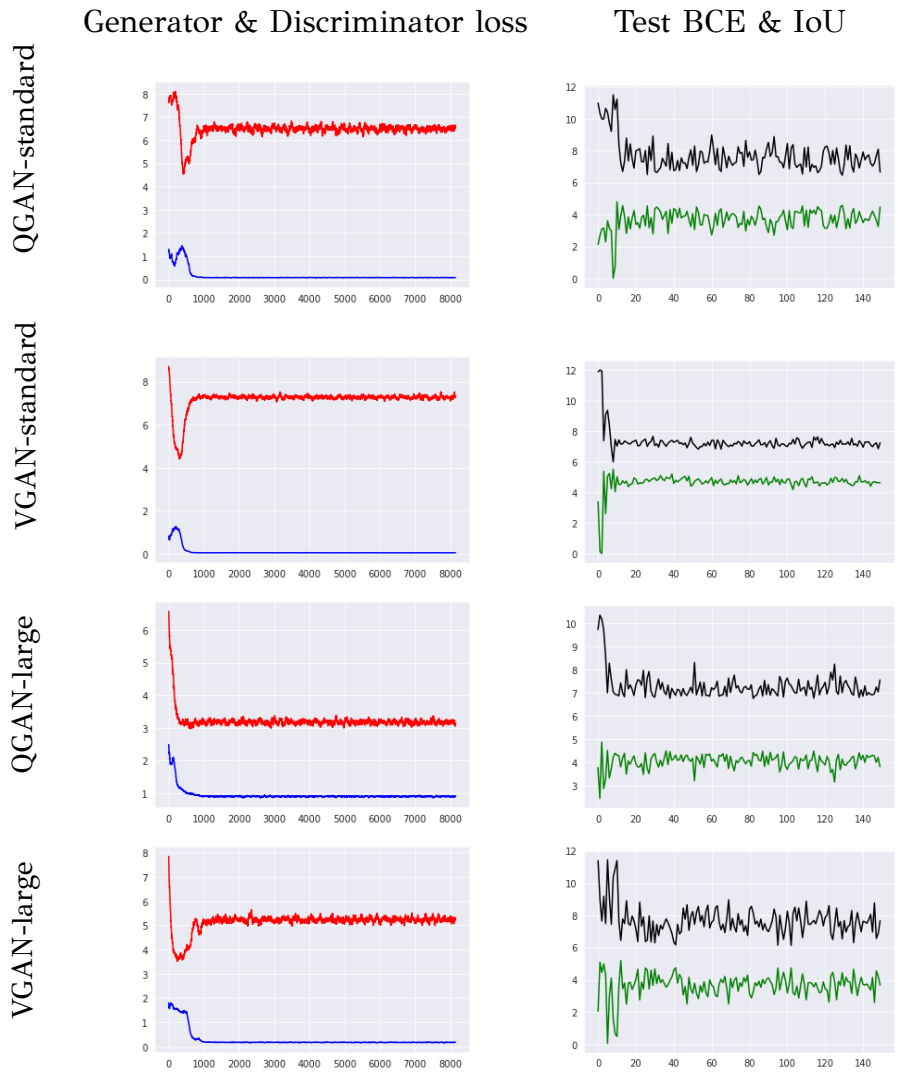


Figure 5.3: Generator loss, Discriminator loss, Test BCE loss and IoU score plots for all models tested in this work. From top row to bottom, we show results for QGAN-standard, VGAN-standard, QGAN-large, VGAN-large. Left column shows Generator and Discriminator loss (red and blue respectively, lower is better for both), and right column shows test BCE and IoU (black and green respectively. Lower BCE is better, higher IoU is better). Generator and Discriminator losses are smoothed with a 100-point uniform convolution kernel and plotted per iteration, test BCE and IoU are plotted per epoch. IoU score is shown multiplied $10\times$ for better visualization.

convolution requires considerably less storage. Judging from the results shown in fig. 5.3 and table 5.1, we can conclude that in all cases performance of the proposed QGAN is comparable with its corresponding non-quaternionic model, and definitely with scores on the same order of magnitude. IoU scores seems somewhat worse, though BCE results are more inconclusive, with QGAN faring slightly better than VGAN with respect to the "Standard" model size. What is definitely noteworthy though, is that QGAN is a considerably less expensive network (in table 5.2 we show

the number of total network parameters for each version of the QGANs and VGANs considered). The number of weights, translated in practice in required storage, is only 25% of the non-quaternionic versions. This means that the proposed QGAN can achieve similar results with the standard GAN, using four times less parameters.

Table 5.1: Numerical results for two variants of the proposed model (QGAN) versus its non-quaternionic counterpart with the same number of neurons (VGAN). Test BCE figures (lower is better) are shown and corresponding IoU scores in parenthesis (higher is better).

Model / Network type	Standard	Large
Quaternionic GAN	6.54(45.4%)	6.91(44.9%)
Vanilla GAN	7.4(51.9%)	6.45(52.0%)

Table 5.2: Comparative table of model sizes, measured in numbers of trainable weights. Number of quaternionic and real weights are shown respectively. In parenthesis, the number of equivalent real weights is shown, in order to ease storage size requirements comparison for the two variants.

Model / Network type	Standard	Large
Quaternionic GAN	$381,426\ (1,525,704)$	$1,516,514\ (6,066,056)$
Vanilla GAN	$6,053,826$	$24,166,274$

5.7 Concluding discussion

We have presented a new variant of Generative Adversarial Networks that uses quaternion-valued neurons and weights, as well as suitable quaternionic variants of convolutional and deconvolutional layers. The proposed model is a conditional GAN, with the generator accepting a color input image and outputing a detection heatmap. We have applied the new model on the task of inscription detection, where we have used a set of byzantine monument text inscriptions as our targets. Quaternion-valued networks such as the proposed one can inherently deal with representing color intercorrelation. The inscriptions themselves are not characterized by color variance; however, the elements that are not part of the inscription very often do (murals, paintings). The proposed network showed that it can be as effective as a real-valued GAN, while being much less expensive in terms of model size. This can be a very important factor, especially in use cases where the resource budget is very constrained (e.g. neural networks running on mobile phones, etc.).

Chapter 6

Conclusions

The subject of this dissertation lies on document image KWS as a recognition-free image retrieval approach, suitable for indexing documents available in various scripts and fonts. In this framework, we have thoroughly studied the recent literature on word spotting along with its key components that make it work. During the course of this thesis, we proposed a number of KWS techniques addressing most of the underlying challenges present in modern and historical documents, mainly focusing on the creation of fast and accurate word image representations. In the following, we summarize the main contributions of the thesis.

In Chapter 2 we carried out an extensive review comprising of more than 250 keyword spotting techniques, underlining crucial points that need to be considered before developing KWS methods. In particular, our main contributions concerning these points can be outlined as follows:

- We highlighted the challenges that derive from the intrinsic nature of original documents which is related to the age of the text, the alphabet and language it is written, as well as the source (typewritten or handwritten) that it came from. Among these challenges, we denote the degradations of historical documents as opposed to good quality modern texts, the variability of handwriting when a document is written my multiple authors, the text cursiveness which prevails especially in Arabic and Indic scripts and the particular writing direction of Chinese characters compared to the Latin alphabet. Moreover, KWS methods need to be robust to the above challenges, while at the same time allowing efficient retrieval. For example, the ability to cast arbitrary queries for a document

collection, where the availability of annotated data to train robust models is not always ensured, is a desired property for KWS. Lastly, the appropriate selection of features and their compact representation learning, highly distinct, yet not to specific to particular writing styles and languages is a must, in order to achieve high KWS performance in heterogeneous documents.

- Basic document image analysis technologies involved to the KWS pipeline, in order to make the task easier for the steps that follow and increase its performance were also examined. Typically, these technologies perform an image preprocessing step which may, for instance, segment the document image at the desired level of abstraction (line, word), binarize the image to remove unnecessary background noise, or even normalize the images before they get fed to the main component of the KWS system. The impact on KWS performance after involving or omitting each distinct preprocessing step is also investigated.

- With respect to the main KWS system architecture, an analytical taxonomy of the various feature extraction and representation learning methodologies, along with the employed matching algorithm was performed. Concerning the image matching procedure, a further categorization was proposed, according to the word, line or document level that image matching is applied.

- A number of methods which are used to improve the retrieved results of a KWS system by exploiting the information of the ranked lists obtained from user queries were presented. Therein, the user is asked to select positive query instances in a supervised fashion, whereas data fusion and re-ranking techniques alleviate the task in a purely unsupervised manner which can further raise the KWS performance.

- In the last part of the chapter we presented the most commonly used datasets and evaluation indices to assess the KWS performance. Through a systematic study of diverse approaches, we suggest a common experimental protocol depending on the type of query (string or example-based), the use of annotated data, and the segmentation level that methods perform, to alleviate evaluation bias among similar is spirit techniques. Results achieved by the state of the art in widely used benchmarks were also reported, further underlining future directions to be explored by trending deep learning methods.

In Chapter 3 we presented two word segmentation-based methods developed for handwritten KWS on modern and historical documents at an early stage of the thesis. Both methods employ a family of shallow-learnt, handcrafted local image features that form variable-length sequences. An overview of the main contributions of this Chapter is as follows:

- A learning-based method for KWS in modern Greek handwritten text written under multi-writer conditions was proposed. The method follows a query-by-word class paradigm which is a variant of example-based techniques. To address intra-class writing style variability, a representative word-class model is built using a subset of images from that class under a series of learning schemes, which iteratively refine its shape to approximate the common class-boundaries of training samples. To achieve boundary level localization we rely on local contour features, dubbed pairs of adjacent segments (PAS), described by a variable-length vector, invariant to translation and scale. Similar descriptors of the training sets are clustered together to form a visual codebook. PAS types (visual words) frequently reoccurring inside training images vote for a particular location and scale of the actual PAS that will form the representative shape of the word-class. Then, a non-rigid point set matching algorithm, which rejects clutter points, is employed to refine the final shape model representation. This representation is then used as a word-class query, able to deform to unknown (during training) writing styles during retrieval. The success of writing-style adaptation is attributed to a statistical model of intra-class deformations using principal component analysis.

- Due to the limitation of the former learning-based method to cast out of sample queries, an improved QBE-based method was developed, which suggests an unsupervised adaptation of local contour features (PAS), able to obtain faster and more accurate retrieval. The contribution of the proposed technique lies on the direct use of these features to retrieve the location and scale of the center of the query's bounding box inside a test image using Hough transform. This acts as an alignment step which initializes the non-rigid point set matching algorithm to deform the query word so as to approach the shape of the test word images. The outcome of this step produces a boundary level localization, typically scored by a weighted sum of four terms reflecting the accuracy of the registration task.

This sum is relaxed to allow for intra-class writing-style variations and is also extended with an extra term to account for false detections obtained from partial matches of the query inside the test image. The proposed method consistently outperformed counter learning-free approaches for a number of heterogenous handwritten documents.

Chapter 4 includes two proposed methodologies for segmentation-based multi-writer KWS. Its main contributions are summarized as follows:

- A fixed-length word image representation for KWS in polytonic Greek documents was proposed. The representation encodes binary attributes of the word image transcription, simulating the occurence or absense of sub-word components at specific splits of the word. To this end, three alternatives are suggested to expand the binary representation's capacity so as to handle the Greek alphabet and its various combinations of diacritic marks thus outperforming a number of related works for KWS in polytonic Greek text.

- The second part of this Chapter presented the theoretical fundamentals behind neural networks including their training and inference procedures. Then we briefly reviewed convolutional neural networks, which is the standard deep learning-based model that we focus for the rest of the Chapter.

- In the third part of this Chapter we proposed a deep learning-based framework, where a convolutional neural network is employed as an extraction model of deep features which are used to adapt a seminal KWS approach on weakly supervised target document collections. Therein, data distribution differs substantially, in terms of target set intra-class variances and number of word-classes, from the source training sets. To this end, spatial transformations of the convolutional feature space aim to prevent the ability of the KWS model to correctly predict transferable representations, so as to adversarially improve its efficiency and robustness to unknown writing styles and unseen word-classes. Numerical experiments of the adaptation of deep features from a low resource document collection to a much more diverse target dataset, where little annotations exist to finetune the original model, confirm the validity of our approach and are on par with the state of the art.

Finally, in Chapter 5 we presented a variation of adversarial learning to solve a

slightly more abstract problem to KWS, which is more related to text spotting in natural images, obtained from Byzantine inscriptions. In this respect, we introduce and discuss quaternion generative adversarial networks, a variant of generative adversarial networks that uses quaternion-valued inputs, weights and intermediate network representations to efficiently encode cross-channel information carried by multichannel signals (e.g. color images) holistically, while at the same time requiring far less computational resources. Standard convolutional and deconvolutional layers are replaced by their quaternionic variants, in both generator and discriminator nets, while activations and loss functions are adapted accordingly. The proposed model is on par with a vanilla conditional generative adversarial network, whereas requiring almost a quarter of the total parameters.

Our future plans focus on extending the proposed deep features, as described in Chapter 4, in a domain adaptation framework, wherein the alignment of corresponding representations between source and target domains will employ adversarial learning along with self-training and iterative retraining techniques. By these means, we aim to transfer pseudo-labels from semi-supervised target collections, to augment the discriminative power of deep features and thus infer more transferable representations.

Bibliography

[1] V. Ferrari, T. Tuytelaars, and L. V. Gool, "Object detection by contour segment networks," in *Proceedings of the* 9^{th} *European Conference on Computer Vision (ECCV)*, 2006, pp. 14–28.

[2] V. Lavrenko, T. M. Rath, and R. Manmatha, "Holistic word recognition for handwritten historical documents," in *Proceedings of the* 1^{st} *International Workshop on Document Image Analysis for Libraries*, 2004, pp. 278–287.

[3] S. Sudholt, L. Rothacker, and G. A. Fink, "Query-by-online word spotting revisited: Using CNNs for cross-domain retrieval," in *Proceedings of the* 14^{th} *IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, 2017, pp. 481–486.

[4] S. Sudholt and G. A. Fink, "PHOCNet: A deep convolutional neural network for word spotting in handwritten documents," in *Proceedings of the* 15^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016, pp. 277–282.

[5] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proceedings of the* 29^{th} *International Conference on Advances in Neural Information Processing Systems (NIPS)*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015, p. 2017–2025.

[6] M. Eden, "Handwriting and pattern recognition," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 160–166, 1962.

[7] C. Tappert, C. Suen, and T. Wakahara, "The state of the art in online handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 8, pp. 787–808, 1990.

[8] M. Liwicki and H. Bunke, "Combining on-line and off-line systems for handwriting recognition," in *Proceedings of the 9^{th} International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, 2007, pp. 372–376.

[9] A. L. Bianne-Bernard, F. Menasri, R. H. Mohamad, C. Mokbel, C. Kermorvant, and L. Likforman-Sulem, "Dynamic and contextual information in HMM modeling for handwritten word recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 2066–2080, 2011.

[10] S. Shetty, H. Srinivasan, and S. Srihari, "Handwritten word recognition using conditional random fields," in *Proceedings of the 9^{th} International Conference on Document Analysis and Recognition (ICDAR)*, vol. 2, 2007, pp. 1098–1102.

[11] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Transactions on Pattern Analysis and Machince Intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[12] X. Zhang and C. L. Tan, "Unconstrained handwritten word recognition based on trigrams using BLSTM," in *Proceedings of the 22^{th} International Conference on Pattern Recognition (ICPR)*, 2014, pp. 2914–2919.

[13] J. Michael, R. Labahn, T. Grüning, and J. Zöllner, "Evaluating sequence-to-sequence models for handwritten text recognition," *CoRR*, vol. abs/1903.07377, 2019. [Online]. Available: http://arxiv.org/abs/1903.07377

[14] A. Ahmad, C. Viard-Gaudin, and M. Khalid, "Lexicon-based word recognition using support vector machine and hidden Markov model," in *Proceedings of the 10^{th} International Conference on Document Analysis and Recognition (ICDAR)*, 2009, pp. 161–165.

[15] S. Prum, M. Visani, and J. Ogier, "Cursive on-line handwriting word recognition using a bi-character model for large lexicon applications," in *Proceedings of the 12^{th} International Conference on Frontiers for Handwriting Recognition (ICFHR)*, 2010, pp. 194–199.

[16] S. Espana-Boquera, M. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, "Improving offline handwritten text recognition with hybrid HMM/ANN mod-

els," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 767–779, 2011.

[17] J. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden Markov modeling for speaker-independent word spotting," in *Proceedings of the* 14^{th} *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, 1989, pp. 627–630.

[18] S. Khoubyari and J. J. Hull, "Keyword location in noisy document images," in *Proceedings of the* 2^{nd} *Annual Symposium on Document Analysis and Information Retrieval*, 1993, pp. 217–231.

[19] F. Chen, L. Wilcox, and D. Bloomberg, "Word spotting in scanned images using hidden Markov models," in *Proceedings of the* 18^{th} *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, 1993, pp. 1–4.

[20] R. Manmatha, C. Han, and E. Riseman, "Word spotting: a new approach to indexing handwriting," in *Proceedings of the* 9^{th} *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1996, pp. 631–637.

[21] P. Keaton, H. Greenspan, and R. Goodman, "Keyword spotting for cursive document retrieval," in *Proceedings of the* 1^{st} *Workshop on Document Image Analysis (DIA)*, 1997, pp. 74–81.

[22] J. A. Rodríguez-Serrano and F. Perronnin, "Handwritten word-spotting using hidden Markov models and universal vocabularies," *Pattern Recognition*, vol. 42, no. 9, pp. 2106–2116, 2009.

[23] K. Khurshid, C. Faure, and N. Vincent, "Fusion of word spotting and spatial information for figure caption retrieval in historical document images," in *Proceedings of the* 10^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, 2009, pp. 266–270.

[24] H. Cao, A. Bhardwaj, and V. Govindaraju, "A probabilistic method for keyword retrieval in handwritten document images," *Pattern Recognition*, vol. 42, no. 12, pp. 3374–3382, 2009.

[25] A. Tarafdar, U. Pal, J. Ramel, N. Ragot, and B. Chaudhuri, "Word spotting in Bangla and English graphical documents," in *Proceedings of the* 22^{nd} *International Conference on Pattern Recognition (ICPR)*, 2014, pp. 3044–3049.

[26] L. Rothacker, D. Fisseler, G. Muller, F. Weichert, and G. A. Fink, "Retrieving cuneiform structures in a segmentation-free word spotting framework," in *Proceedings of the* 3^{rd} *International Workshop on Historical Document Imaging and Processing (HIP)*, 2015, pp. 129–136.

[27] B. Bogacz, N. Howe, and H. Mara, "Segmentation free spotting of cuneiform using part structured models," in *Proceedings of the* 15^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016, pp. 301–306.

[28] V. Romero, A. H. Toselli, J. A. Sànchez, and E. Vidal, "Handwriting transcription and keyword spotting in historical daily records documents," in *Proceedings of the* 12^{th} *IAPR Workshop on Document Analysis Systems (DAS)*, 2016, pp. 275–280.

[29] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke, "A novel word spotting method based on recurrent neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 211–224, 2012.

[30] V. Frinken, A. Fischer, M. Baumgartner, and H. Bunke, "Keyword spotting for self-training of BLSTM NN-based handwriting recognition systems," *Pattern Recognition*, vol. 47, no. 3, pp. 1073 – 1082, 2014.

[31] J. Almazan, A. Gordo, A. Fornes, and E. Valveny, "Word spotting and recognition with embedded attributes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 12, pp. 2552–2566, 2014.

[32] P. Krishnan, K. Dutta, and C. V. Jawahar, "Deep feature embedding for accurate recognition and retrieval of handwritten text," in *Proceedings of the* 15^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016, pp. 289–294.

[33] A. K. Bhunia, P. P. Roy, A. Mohta, and U. Pal, "Cross-language framework for word recognition and spotting of indic scripts," *Pattern Recognition*, vol. 79, pp. 12–31, 2018.

[34] P. Krishnan, K. Dutta, and C. Jawahar, "Word spotting and recognition using deep embedding," in *Proceedings of the* 13^{th} *IAPR International Workshop on Document Analysis Systems (DAS)*, 2018, pp. 1–6.

[35] M. Mhiri, C. Desrosiers, and M. Cheriet, "Word spotting and recognition via a joint deep embedding of image and text," *Pattern Recognition*, vol. 88, pp. 312–320, 2019.

[36] S. Bansal, P. Krishnan, and C. V. Jawahar, "Fused text recogniser and deep embeddings improve word recognition and retrieval," *CoRR*, vol. abs/2007.00166, 2020. [Online]. Available: https://arxiv.org/abs/2007.00166

[37] G. Retsinas, G. Sfikas, and P. Maragos, "WSRNet: Joint spotting and recognition of handwritten words," *CoRR*, vol. abs/2008.07109, 2020. [Online]. Available: https://arxiv.org/abs/2008.07109

[38] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *International Journal of Computer Vision*, vol. 116, no. 1, pp. 1–20, 2016.

[39] F. Su, W. Ding, L. Wang, S. Shan, and H. Xu, "Text proposals based on windowed maximally stable extremal region for scene text detection," in *Proceedings of the* 14^{th} *IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017, pp. 376–381.

[40] A. P. Giotis, G. Sfikas, B. Gatos, and C. Nikou, "A survey of document image word spotting techniques," *Pattern Recognition*, vol. 68, pp. 310–332, 2017.

[41] A. Giotis, D. Gerogiannis, and C. Nikou, "Word spotting in handwritten text using contour-based models," in *Proceedings of the* 14^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2014, pp. 399–404.

[42] A. P. Giotis, G. Sfikas, C. Nikou, and B. Gatos, "Shape-based word spotting in handwritten document images," in *Proceedings of the* 13^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 561–565.

[43] G. Sfikas, A. P. Giotis, G. Louloudis, and B. Gatos, "Using attributes for word spotting and recognition in polytonic greek documents," in *Proceedings of the*

13^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 686–690.

[44] G. Sfikas, A. P. Giotis, G. Retsinas, and C. Nikou, "Quaternion generative adversarial networks for inscription detection in byzantine monuments," in *Proceedings of the* 2^{nd} *Workshop on Pattern Recognition for Cultural Heritage (PatReCH 2020), held in conjunction with the* 25^{th} *International Conference on Pattern Recognition (ICPR)*, 2021, pp. 171–184.

[45] A. Murugappan, B. Ramachandran, and P. Dhavachelvan, "A survey of keyword spotting techniques for printed document images," *Artificial Intelligence Review*, vol. 35, no. 2, pp. 119–136, 2011.

[46] M. Kchaou, S. Kanoun, and J. Ogier, "Segmentation and word spotting methods for printed and handwritten Arabic texts: A comparative study," in *Proceedings of the* 13^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2012, pp. 274–279.

[47] S. Marinai, B. Miotti, and G. Soda, "Digital libraries and document image retrieval techniques: A survey," in *Learning Structure and Schemas from Documents*, ser. Studies in Computational Intelligence, M. Biba and F. Xhafa, Eds. Springer Berlin Heidelberg, 2011, vol. 375, pp. 181–204.

[48] C. Tan, X. Zhang, and L. Li, "Image based retrieval and keyword spotting in documents," in *Handbook of Document Image Processing and Recognition*, D. Doermann and K. Tombre, Eds. Springer London, 2014, pp. 805–842.

[49] A. Sharma and S. K. Pramod, "Adapting off-the-shelf cnns for word spotting & recognition," in *Proceedings of the* 13^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 986–990.

[50] D. Aldavert, M. Rusiñol, R. Toledo, and J. Llados, "Integrating visual and textual cues for query-by-string word spotting," in *Proceedings of the* 12^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 511–515.

[51] ——, "A study of bag-of-visual-words representations for handwritten keyword spotting," *International Journal on Document Analysis and Recognition*, vol. 18, no. 3, pp. 223–234, 2015.

[52] D. Aldavert and M. Rusiñol, "Synthetically generated semantic codebook for bag-of-visual-words based word spotting," in *Proceedings of the* 13^{th} *IAPR International Workshop on Document Analysis Systems (DAS)*, 2018, pp. 223–228.

[53] K. Zagoris, I. Pratikakis, and B. Gatos, "Segmentation-based historical handwritten word spotting using document-specific local features," in *Proceedings of the* 14^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2014, pp. 9–14.

[54] S. En, C. Petitjean, S. Nicolas, and L. Heutte, "A scalable pattern spotting system for historical documents," *Pattern Recognition*, vol. 54, pp. 149–161, 2016.

[55] X. Zhang and C. Tan, "Segmentation-free keyword spotting for handwritten documents based on heat kernel signature," in *Proceedings of the* 12^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 827–831.

[56] A. Fornes, V. Frinken, A. Fischer, J. Almazan, G. Jackson, and H. Bunke, "A keyword spotting approach using blurred shape model-based descriptors," in *Proceedings of the* 10^{th} *Workshop on Historical Document Imaging and Processing*, 2011, pp. 83–90.

[57] U. Roy, N. Sankaran, K. Sankar, and C. Jawahar, "Character n-gram spotting on handwritten documents using weakly-supervised segmentation," in *Proceedings of the* 12^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 577–581.

[58] L. Rothacker, M. Rusiñol, and G. A. Fink, "Bag-of-features HMMs for segmentation-free word spotting in handwritten documents," in *Proceedings of the* 12^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 1305–1309.

[59] L. Rothacker and G. A. Fink, "Segmentation-free query-by-string word spotting with bag-of-features HMMs," in *Proceedings of the* 13^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 661–665.

[60] T. Mondal, N. Ragot, J. Ramel, and U. Pal, "A fast word retrieval technique based on kernelized locality sensitive hashing," in *Proceedings of the* 12^{th} *In-*

ternational Conference on Document Analysis and Recognition (ICDAR), 2013, pp. 1195–1199.

[61] V. Dovgalecs, A. Burnett, P. Tranouez, S. Nicolas, and L. Heutte, "Spot it! Finding words and patterns in historical documents," in *Proceedings of the* 12^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 1039–1043.

[62] T. M. Rath and R. Manmatha, "Word spotting for historical documents," *International Journal on Document Analysis and Recognition*, vol. 9, no. 2-4, pp. 139–152, 2007.

[63] Z. Zhong, W. Pan, L. Jin, H. Mouchère, and C. Viard-Gaudin, "SpottingNet: Learning the similarity of word images with convolutional neural network for word spotting in handwritten historical documents," in *Proceedings of the* 15^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016, pp. 295–300.

[64] A. I. Wagan, S. Bres, and H. Emptoz, "Word spotting in Alice's adventures underground using multi scale integral orientation features," in *Proceedings of the* 9^{th} *IAPR International Workshop on Document Analysis Systems (DAS)*, 2010, pp. 417–424.

[65] G. Kumar, Z. Shi, S. Setlur, V. Govindaraju, and S. Ramachandrula, "Keyword spotting framework using dynamic background model," in *Proceedings of the* 12^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2012, pp. 582–587.

[66] A. H. Toselli, J. Puigcerver, and E. Vidal, "Two methods to improve confidence scores for lexicon-free word spotting in handwritten text," in *Proceedings of the* 15^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016, pp. 349–354.

[67] G. Retsinas, G. Louloudis, N. Stamatopoulos, and B. Gatos, "Keyword spotting in handwritten documents using projections of oriented gradients," in *Proceedings of the* 12^{th} *IAPR Workshop on Document Analysis Systems (DAS)*, 2016, pp. 411–416.

[68] Y. Liang, M. Fairhurst, and R. Guest, "A synthesised word approach to word retrieval in handwritten documents," *Pattern Recognition*, vol. 45, no. 12, pp. 4225 – 4236, 2012.

[69] T. Wilkinson and A. Brun, "Semantic and verbatim word spotting using deep neural networks," in *Proceedings of the* 15^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016, pp. 307–312.

[70] T. Wilkinson, J. Lindström, and A. Brun, "Neural ctrl-f: Segmentation-free query-by-string word spotting in handwritten manuscript collections," in *Proceedings of the* 16^{th} *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4443–4452.

[71] A. Fischer, V. Frinken, H. Bunke, and C. Suen, "Improving HMM-based keyword spotting with character language models," in *Proceedings of the* 12^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 506–510.

[72] S. K. Ghosh and E. Valveny, "Query by string word spotting based on character bi-gram indexing," in *Proceedings of the* 13^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 881–885.

[73] ——, "R-phoc: Segmentation-free word spotting using cnn," in *Proceedings of the* 14^{th} *IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017, pp. 801–806.

[74] G. Retsinas, G. Sfikas, and B. Gatos, "Transferable deep features for keyword spotting," in *Proceedings of the* 25^{th} *European Signal Processing Conference (EUSIPCO)*, 2017, pp. 1–5.

[75] Y. Kessentini, C. Chatelain, and T. Paquet, "Word spotting and regular expression detection in handwritten documents," in *Proceedings of the* 12^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 516–520.

[76] Y. Kessentini and T. Paquet, "Keyword spotting in handwritten documents based on a generic text line HMM and a SVM verification," in *Proceedings of the* 13^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 41–45.

[77] C. Choisy, "Dynamic handwritten keyword spotting based on the NSHP-HMM," in *Proceedings of the* 9^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2007, pp. 242–246.

[78] N. R. Howe, "Part-structured inkball models for one-shot handwritten word spotting," in *Proceedings of the* 12^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 582–586.

[79] ——, "Inkball models for character localization and out-of-vocabulary word spotting," in *Proceedings of the* 13^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 381–385.

[80] F. Wolf, A. Fischer, and G. A. Fink, "Graph convolutional neural networks for learning attribute representations for word spotting," in *Proceedings of the* 16^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2021, pp. 50–64.

[81] F. Daraee, S. Mozaffari, and S. M. Razavi, "Handwritten keyword spotting using deep neural networks and certainty prediction," *Computers & Electrical Engineering*, vol. 92, pp. 107–111, 2021.

[82] J. Puigcerver, A. Toselli, and E. Vidal, "Word-graph-based handwriting keyword spotting of out-of-vocabulary queries," in *Proceedings of the* 22^{nd} *International Conference on Pattern Recognition (ICPR)*, 2014, pp. 2035–2040.

[83] ——, "Querying out-of-vocabulary words in lexicon-based keyword spotting," *Neural Computing and Applications*, vol. 28, pp. 2373–2382, 2016.

[84] P. Riba, J. Llados, and A. Fornes, "Handwritten word spotting by inexact matching of grapheme graphs," in *Proceedings of the* 13^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 781–785.

[85] A. Hast and A. Fornés, "A segmentation-free handwritten word spotting approach by relaxed feature matching," in *Proceedings of the* 12^{th} *IAPR Workshop on Document Analysis Systems (DAS)*, 2016, pp. 150–155.

[86] A. Hast and E. Vats, "Radial line fourier descriptor for historical handwritten text representation," in *Proceedings of the* 26^{th} *International Conference in Central*

Europe on Computer Graphics, Visualization and Computer Vision (WSCG), 2018, pp. 31–40.

[87] M. Villegas, A. H. Toselli, V. Romero, and E. Vidal, "Exploiting existing modern transcripts for historical handwritten text recognition," in *Proceedings of the* 15^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016, pp. 66–71.

[88] G. Fink, L. Rothacker, and R. Grzeszick, "Grouping historical postcards using query-by-example word spotting," in *Proceedings of the* 14^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2014, pp. 470–475.

[89] A. Cheikhrouhou, Y. Kessentini, and S. Kanoun, "Multi-task learning for simultaneous script identification and keyword spotting in document images," *Pattern Recognition*, vol. 113, p. 107832, 2021.

[90] H. Chatbri, P. Kwan, and K. Kameyama, "An application-independent and segmentation-free approach for spotting queries in document images," in *Proceedings of the* 22^{nd} *International Conference on Pattern Recognition (ICPR)*, 2014, pp. 2891–2896.

[91] J. Llados, M. Rusiñol, A. Fornes, D. Fernandez, and A. Dutta, "On the influence of word representations for handwritten word spotting in historical documents," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 26, no. 05, p. 1263002, 2012.

[92] P. Wang, V. Eglin, C. Garcia, C. Largeron, J. Llados, and A. Fornes, "A novel learning-free word spotting approach based on graph representation," in *Proceedings of the* 11^{th} *IAPR International Workshop on Document Analysis Systems (DAS)*, 2014, pp. 207–211.

[93] J. P. Van Oosten and L. Schomaker, "Separability versus prototypicality in handwritten word retrieval," in *Proceedings of the* 13^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2012, pp. 8–13.

[94] T. van Der Zant, L. Schomaker, and K. Haak, "Handwritten-word spotting using biologically inspired features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1945–1957, 2008.

[95] A. Kovalchuk, L. Wolf, and N. Dershowitz, "A simple and fast word spotting method," in *Proceedings of the* 14^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2014, pp. 3–8.

[96] J. Almazan, A. Gordo, A. Fornes, and E. Valveny, "Segmentation-free word spotting with exemplar SVMs," *Pattern Recognition*, vol. 47, no. 12, pp. 3967 – 3978, 2014.

[97] T. Mondal, N. Ragot, J. Y. Ramel, and U. Pal, "Flexible sequence matching technique: Application to word spotting in degraded documents," in *Proceedings of the* 14^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2014, pp. 210–215.

[98] T. Mondal, N. Ragot, J.-Y. Ramel, and U. Pal, "Performance evaluation of DTW and its variants for word spotting in degraded documents," in *Proceedings of the* 13^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 1141–1145.

[99] J. A. Rodríguez-Serrano and F. Perronnin, "A model-based sequence similarity with application to handwritten word spotting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2108–2120, 2012.

[100] S. Sudholt and G. A. Fink, "Evaluating word string embeddings and loss functions for cnn-based word spotting," in *Proceedings of the* 14^{th} *IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017, pp. 493–498.

[101] M. Al-Rawi, E. Valveny, and D. Karatzas, "Can one deep learning model learn script-independent multilingual word-spotting?" in *Proceedings of the* 15^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 260–267.

[102] Y. Leydier, A. Ouji, F. LeBourgeois, and H. Emptoz, "Towards an omnilingual word retrieval system for ancient manuscripts," *Pattern Recognition*, vol. 42, no. 9, pp. 2089–2105, 2009.

[103] K. Terasawa and Y. Tanaka, "Slit style HoG feature for document image word spotting," in *Proceedings of the* 10^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2009, pp. 116–120.

[104] C. Sugawara, T. Miyazaki, Y. Sugaya, and S. Omachi, "Text retrieval for japanese historical documents by image generation," in *Proceedings of the 4^{th} International Workshop on Historical Document Imaging and Processing (HIP)*, 2017, p. 19–24.

[105] A. Abidi, A. Jamil, I. Siddiqi, and K. Khurshid, "Word spotting based retrieval of Urdu handwritten documents," in *Proceedings of the 13^{th} International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2012, pp. 331–336.

[106] M. W. Sagheer, N. Nobile, C. L. He, and C. Y. Suen, "A novel handwritten Urdu word spotting based on connected components analysis," in *Proceedings of the 20^{th} International Conference on Pattern Recognition (ICPR)*, 2010, pp. 2013–2016.

[107] M. Khayyat, L. Lam, and C. Y. Suen, "Learning-based word spotting system for Arabic handwritten documents," *Pattern Recognition*, vol. 47, no. 3, pp. 1021 – 1030, 2014.

[108] N. Li, J. Chen, H. Cao, B. Zhang, and P. Natarajan, "Applications of recurrent neural network language model in offline handwriting recognition and word spotting," in *Proceedings of the 14^{th} International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2014, pp. 134–139.

[109] G. Kumar and V. Govindaraju, "A Bayesian approach to script independent multilingual keyword spotting," in *Proceedings of the 14^{th} International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2014, pp. 357–362.

[110] S. Wshah, G. Kumar, and V. Govindaraju, "Statistical script independent word spotting in offline handwritten documents," *Pattern Recognition*, vol. 47, no. 3, pp. 1039 – 1050, 2014.

[111] S. N. Srihari and G. R. Ball, "Language independent word spotting in scanned documents," in *Proceedings of the 11^{th} International Conference on Asian Digital Libraries (ICADL)*, 2008, pp. 134–143.

[112] L. Huang, F. Yin, Q.-H. Chen, and C.-L. Liu, "Keyword spotting in unconstrained handwritten Chinese documents using contextual word model," *Image and Vision Computing*, vol. 31, no. 12, pp. 958 – 968, 2013.

[113] R. Saabni and A. Bronstein, "Fast keyword searching using 'boostmap' based embedding," in *Proceedings of the* 13^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2012, pp. 734–739.

[114] M. Shah and C. Suen, "Word spotting in gray scale handwritten Pashto documents," in *Proceedings of the* 12^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2010, pp. 136–141.

[115] E. F. Can and P. Duygulu, "A line-based representation for matching words in historical manuscripts," *Pattern Recognition Letters*, vol. 32, no. 8, pp. 1126 – 1138, 2011.

[116] M. Rusiñol, D. Aldavert, R. Toledo, and J. Llados, "Browsing heterogeneous document collections by a segmentation-free word spotting method," in *Proceedings of the* 11^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2011, pp. 63–67.

[117] H. Wei, G. Gao, and Y. Bao, "A method for removing inflectional suffixes in word spotting of Mongolian Kanjur," in *Proceedings of the* 11^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2011, pp. 88–92.

[118] H. Wei, G. Gao, and X. Su, "A multiple instances approach to improving keyword spotting on historical Mongolian document images," in *Proceedings of the* 13^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 121–125.

[119] H. Wei, H. Zhang, and G. Gao, "Representing word image using visual word embeddings and rnn for keyword spotting on historical document images," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2017, pp. 1368–1373.

[120] H. Wei, Y. Kang, and H. Zhang, "Word image representation based on sequence to sequence model with attention mechanism for out-of-vocabulary keyword spotting," in *Proceedings of the* 21^{st} *International Conference on High Performance Computing and Communications; IEEE* 17^{th} *International Conference on Smart City; IEEE* 5^{th} *International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2019, pp. 2224–2231.

[121] H. Wei, J. Zhang, and K. Liu, "A hybrid representation of word images for keyword spotting," in *Proceedings of the* 27^{th} *International Conference on Neural Information Processing (ICONIP)*, 2020, pp. 3–10.

[122] V. Ranjan, G. Harit, and C. Jawahar, "Enhancing word image retrieval in presence of font variations," in *Proceedings of the* 22^{nd} *International Conference on Pattern Recognition (ICPR)*, 2014, pp. 2709–2714.

[123] L. Li, S. Lu, and C. Tan, "A fast keyword-spotting technique," in *Proceedings of the* 9^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2007, pp. 68–72.

[124] K. Zagoris, E. Kavallieratou, and N. Papamarkos, "A document image retrieval system," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 6, pp. 872 – 879, 2010.

[125] S. Bai, L. Li, and C. Tan, "Keyword spotting in document images through word shape coding," in *Proceedings of the* 10^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2009, pp. 331–335.

[126] G. Louloudis, A. Kesidis, and B. Gatos, "Efficient word retrieval using a multiple ranking combination scheme," in *Proceedings of the* 10^{th} *IAPR International Workshop on Document Analysis Systems (DAS)*, 2012, pp. 379–383.

[127] P. Roy, J. Ramel, and N. Ragot, "Word retrieval in historical document using character-primitives," in *Proceedings of the* 11^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2011, pp. 678–682.

[128] A. Papandreou, B. Gatos, and G. Louloudis, "An adaptive zoning technique for efficient word retrieval using dynamic time warping," in *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage (DATeCH)*, 2014, pp. 147–152.

[129] B. Gatos and I. Pratikakis, "Segmentation-free word spotting in historical printed documents," in *Proceedings of the* 10^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2009, pp. 271–275.

[130] J. Sousa, J. Gil, and J. Pinto, "Word indexing of ancient documents using fuzzy classification," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 5, pp. 852–862, 2007.

[131] S. Marinai, "Text retrieval from early printed books," *International Journal on Document Analysis and Recognition*, vol. 14, no. 2, pp. 117–129, 2011.

[132] T. Konidaris, B. Gatos, K. Ntzios, I. Pratikakis, S. Theodoridis, and S. Perantonis, "Keyword-guided word spotting in historical printed documents using synthetic data and user feedback," *International Journal on Document Analysis and Recognition*, vol. 9, no. 2-4, pp. 167–177, 2007.

[133] A. L. Kesidis, E. Galiotou, B. Gatos, and I. Pratikakis, "A word spotting framework for historical machine-printed documents," *International Journal on Document Analysis and Recognition*, vol. 14, no. 2, pp. 131–144, 2011.

[134] Y. Xia, K. Wang, and M. Li, "Chinese keyword spotting using knowledge-based clustering," in *Proceedings of the* 11^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2011, pp. 789–793.

[135] E. Hassan, S. Chaudhury, and M. Gopal, "Word shape descriptor-based document image indexing: a new DBH-based approach," *International Journal on Document Analysis and Recognition*, vol. 16, no. 3, pp. 227–246, 2013.

[136] P. Krishnan and C. Jawahar, "Bringing semantics in word image retrieval," in *Proceedings of the* 12^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 733–737.

[137] R. Shekhar and C. Jawahar, "Document specific sparse coding for word retrieval," in *Proceedings of the* 12^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 643–647.

[138] I. Z. Yalniz and R. Manmatha, "An efficient framework for searching text in noisy document images," in *Proceedings of the* 10^{th} *IAPR International Workshop on Document Analysis Systems (DAS)*, 2012, pp. 48–52.

[139] M. Meshesha and C. V. Jawahar, "Matching word images for content-based retrieval from printed document images," *International Journal on Document Analysis and Recognition*, vol. 11, no. 1, pp. 29–38, 2008.

[140] S. Lu and C. L. Tan, "Retrieval of machine-printed Latin documents through word shape coding," *Pattern Recognition*, vol. 41, no. 5, pp. 1799–1809, 2008.

[141] E. Indermuhle, V. Frinken, A. Fischer, and H. Bunke, "Keyword spotting in online handwritten documents containing text and non-text using BLSTM neural networks," in *Proceedings of the* 11^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2011, pp. 73–77.

[142] H. Zhang, D. H. Wang, and C. L. Liu, "Character confidence based on n-best list for keyword spotting in online Chinese handwritten documents," *Pattern Recognition*, vol. 47, no. 5, pp. 1880 – 1890, 2014.

[143] B. Zhu, A. Shivram, S. Setlur, V. Govindaraju, and M. Nakagawa, "Online handwritten cursive word recognition using segmentation-free MRF in combination with P2DBMN-MQDF," in *Proceedings of the* 12^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 349–353.

[144] P. Krishnan and C. V. Jawahar, "Matching handwritten document images," in *Proceedings of the* 14^{th} *European Conference on Computer Vision (ECCV)*. Springer International Publishing, 2016, pp. 766–782.

[145] M. Kassis and J. El-Sana, "Word spotting using radial descriptor graph," in *Proceedings of the* 15^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016, pp. 31–35.

[146] ——, "Automatic synthesis of historical arabic text for word-spotting," in *Proceedings of the* 12^{th} *IAPR Workshop on Document Analysis Systems (DAS)*, 2016, pp. 239–244.

[147] M. Kulkarni, S. S. Karande, and S. Lodha, "Unsupervised word clustering using deep features," in *Proceedings of the* 12^{th} *IAPR Workshop on Document Analysis Systems (DAS)*, 2016, pp. 263–268.

[148] I. Rabaev, K. Kedem, and J. El-Sana, "Keyword retrieval using scale-space pyramid," in *Proceedings of the* 12^{th} *IAPR Workshop on Document Analysis Systems (DAS)*, 2016, pp. 144–149.

[149] A. Santoro, A. Parziale, and A. Marcelli, "A human in the loop approach to historical handwritten documents transcription," in *Proceedings of the* 15^{th} *In-*

ternational Conference on Frontiers in Handwriting Recognition (ICFHR), 2016, pp. 222–227.

[150] B. Wicht, A. Fischer, and J. Hennebert, "Deep learning features for handwritten keyword spotting," in *Proceedings of the* 23^{rd} *International Conference on Pattern Recognition (ICPR)*, 2016, pp. 3434–3439.

[151] ——, "Keyword spotting with convolutional deep belief networks and dynamic time warping," in *Proceedings of the* 25^{th} *International Conference on Artificial Neural Networks and Machine Learning (ICANN)*, 2016, pp. 113–120.

[152] C. Wieprecht, L. Rothacker, and G. A. Fink, "Word spotting in historical document collections with online-handwritten queries," in *Proceedings of the* 12^{th} *IAPR Workshop on Document Analysis Systems (DAS)*, 2016, pp. 162–167.

[153] L. Gómez, M. Rusiñol, and D. Karatzas, "Lsde: Levenshtein space deep embedding for query-by-string word spotting," in *Proceedings of the* 14^{th} *IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017, pp. 499–504.

[154] K. Zagoris, I. Pratikakis, and B. Gatos, "Unsupervised word spotting in historical handwritten document images using document-oriented local features," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 4032–4041, 2017.

[155] G. Retsinas, N. Stamatopoulos, G. Louloudis, G. Sfikas, and B. Gatos, "Nonlinear manifold embedding on keyword spotting using t-sne," in *Proceedings of the* 14^{th} *IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017, pp. 487–492.

[156] L. Rothacker, S. Sudholt, E. Rusakov, M. Kasperidus, and G. A. Fink, "Word hypotheses for segmentation-free word spotting in historic document images," in *Proceedings of the* 14^{th} *IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017, pp. 1174–1179.

[157] A. Santoro, C. De Stefano, and A. Marcelli, "Assisted transcription of historical documents by keyword spotting: A performance model," in *Proceedings of the* 14^{th} *IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017, pp. 971–976.

[158] M. Stauffer, A. Fischer, and K. Riesen, "Ensembles for graph-based keyword spotting in historical handwritten documents," in *Proceedings of the* 14^{th} *IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017, pp. 714–720.

[159] N. Gurjar, S. Sudholt, and G. A. Fink, "Learning deep representations for word spotting under weak supervision," in *Proceedings of the* 13^{th} *IAPR International Workshop on Document Analysis Systems (DAS)*, 2018, pp. 7–12.

[160] E. Rusakov, L. Rothacker, H. Mo, and G. A. Fink, "A probabilistic retrieval model for word spotting based on direct attribute prediction," in *Proceedings of the* 16^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2018, pp. 38–43.

[161] M. Stauffer, A. Fischer, and K. Riesen, "Keyword spotting in historical handwritten documents based on graph matching," *Pattern Recognition*, vol. 81, pp. 240–253, 2018.

[162] H. Wei, H. Zhang, and G. Gao, "Word image representation based on visual embeddings and spatial constraints for keyword spotting on historical documents," in *Proceedings of the* 24^{th} *International Conference on Pattern Recognition (ICPR)*, 2018, pp. 3616–3621.

[163] G. Retsinas, G. Louloudis, N. Stamatopoulos, and B. Gatos, "Efficient learning-free keyword spotting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1587–1600, 2019.

[164] R. Benabdelaziz, D. Gaceb, and M. Haddad, "Word spotting based on bispace similarity for visual information retrieval in handwritten document images," *International Journal of Computer Vision and Image Processing (IJCVIP)*, vol. 9, no. 3, p. 38–58, 2019.

[165] P. Krishnan and C. V. Jawahar, "Hwnet v2: an efficient word image representation for handwritten documents," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 22, no. 4, p. 387–405, 2019.

[166] M. Mhiri, C. Desrosiers, and M. Cheriet, "Word spotting and recognition via a joint deep embedding of image and text," *Pattern Recognition*, vol. 88, pp. 312–320, 2019.

[167] Y. Serdouk, V. Eglin, S. Bres, and M. Pardoen, "Keyword spotting using siamese triplet deep neural networks," in *Proceedings of the* 15^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 1157–1162.

[168] E. Vats, A. Hast, and A. Fornés, "Training-free and segmentation-free word spotting using feature matching and query expansion," in *Proceedings of the* 15^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 1294–1299.

[169] M. Al-Rawi and E. Valveny, "Compact and efficient multitask learning in vision, language and speech," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 2933–2942.

[170] G. Jie, X. Guo, M. Shang, and J. Sun, "Page-level handwritten word spotting via discriminative feature learning," in *Proceedings of the* 13^{th} *International Conference on Knowledge Science, Engineering and Management (KSEM)*, 2020, pp. 368–379.

[171] F. Westphal, H. Grahn, and N. Lavesson, "Representative image selection for data efficient word spotting," in *Proceedings of the* 14^{th} *IAPR International Workshop on Document Analysis Systems (DAS)*, 2020, pp. 383–397.

[172] T. Wilkinson and C. Nettelblad, "Bootstrapping weakly supervised segmentation-free word spotting through HMM-based alignment," in *Proceedings of the* 17^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2020, pp. 49–54.

[173] F. Wolf, K. Brandenbusch, and G. A. Fink, "Improving handwritten word synthesis for annotation-free word spotting," in *Proceedings of the* 17^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2020, pp. 61–66.

[174] F. Wolf and G. A. Fink, "Annotation-free learning of deep representations for word spotting using synthetic data and self labeling," *CoRR*, vol. abs/2003.01989, 2020. [Online]. Available: https://arxiv.org/abs/2003.01989

[175] P. Zhao, W. Xue, Q. Li, and S. Cai, "Query by strings and return ranking word regions with only one look," in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2020.

[176] A. Amanatiadis, K. Zagoris, and I. Pratikakis, "Word spotting as a service for handwritten documents," in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE)*, 2021, pp. 1–2.

[177] L. Rothacker, F. Wolf, and G. A. Fink, "Annotation-free word spotting with bag-of-features HMMs," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 35, no. 04, p. 2153001, 2021.

[178] G. Retsinas, G. Louloudis, N. Stamatopoulos, G. Sfikas, and B. Gatos, "An alternative deep feature approach to line level keyword spotting," in *Proceedings of the* 32^{nd} *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 650–12 658.

[179] G. Retsinas, G. Sfikas, N. Stamatopoulos, G. Louloudis, and B. Gatos, "Exploring critical aspects of cnn-based keyword spotting. a phocnet study," in *Proceedings of the* 13^{th} *IAPR International Workshop on Document Analysis Systems (DAS)*, 2018, pp. 13–18.

[180] G. Retsinas, G. Sfikas, G. Louloudis, N. Stamatopoulos, and B. Gatos, "Compact deep descriptors for keyword spotting," in *Proceedings of the* 16^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2018, pp. 315–320.

[181] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[182] E. Ataer and P. Duygulu, "Matching Ottoman words: An image retrieval approach to historical document indexing," in *Proceedings of the* 6^{th} *ACM International Conference on Image and Video Retrieval (CIVR)*, 2007, pp. 341–347.

[183] A. Abidi, I. Siddiqi, and K. Khurshid, "Towards searchable digital Urdu libraries - a word spotting based retrieval approach," in *Proceedings of the* 11^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2011, pp. 1344–1348.

[184] M. Rusiñol and J. Llados, "Boosting the handwritten word spotting experience by including the user in the loop," *Pattern Recognition*, vol. 47, no. 3, pp. 1063 – 1072, 2014.

[185] P. P. Roy, U. Pal, and J. Lladós, "Query driven word retrieval in graphical documents," in *Proceedings of the* 9^{th} *IAPR International Workshop on Document Analysis Systems (DAS)*, 2010, pp. 191–198.

[186] J. Sauvola and M. Pietikainen, "Adaptive document image binarization," *Pattern Recognition*, vol. 33, pp. 225–236, 2000.

[187] R. F. Moghaddam and M. Cheriet, "Application of multi-level classifiers and clustering for automatic word spotting in historical document images," in *Proceedings of the* 10^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2009, pp. 511–515.

[188] H. Cao, V. Govindaraju, and A. Bhardwaj, "Unconstrained handwritten document retrieval," *International Journal on Document Analysis and Recognition*, vol. 14, no. 2, pp. 145–157, 2011.

[189] K. Khurshid, C. Faure, and N. Vincent, "Word spotting in historical printed documents using shape and sequence comparisons," *Pattern Recognition*, vol. 45, no. 7, pp. 2598 – 2609, 2012.

[190] Z. Shi and V. Govindaraju, "Historical document image enhancement using background light intensity normalization," in *Proceedings of the* 17^{th} *International Conference on Pattern Recognition*, 2004, pp. 473–476.

[191] H. Cao and V. Govindaraju, "Handwritten carbon form preprocessing based on markov random field," in *Proceedings of the* 20^{th} *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–7.

[192] B. Gatos, I. Pratikakis, and S. Perantonis, "Adaptive degraded document image binarization," *Pattern Recognition*, vol. 39, no. 3, pp. 317 – 327, 2006.

[193] N. Howe, "A Laplacian energy for document binarization," in *Proceedings of the* 11^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2011, pp. 6–10.

[194] E. Vats, A. Hast, and P. Singh, "Automatic document image binarization using bayesian optimization," in *Proceedings of the* 4^{th} *International Workshop on Historical Document Imaging and Processing (HIP)*, 2017, p. 89–94.

[195] S. Cao and V. Govindaraju, "Template-free word spotting in low-quality manuscripts," in *Proceedings of the* 6^{th} *International Conference on Advances in Pattern Recognition (ICAPR)*, 2007, pp. 45–53.

[196] Y. Leydier, F. L. Bourgeois, and H. Emptoz, "Text search for medieval manuscript images," *Pattern Recognition*, vol. 40, no. 12, pp. 3552– 3567, 2007.

[197] M. Rusiñol, D. Aldavert, R. Toledo, and J. Llados, "Efficient segmentation-free keyword spotting in historical document collections," *Pattern Recognition*, vol. 48, no. 2, pp. 545 – 555, 2015.

[198] X. Zhang and C. L. Tan, "Handwritten word image matching based on heat kernel signature," *Pattern Recognition*, vol. 48, no. 11, pp. 3346 – 3356, 2015.

[199] J. Calvo-Zaragoza, G. Vigliensoni, and I. Fujinaga, "Pixel-wise binarization of musical documents with convolutional neural networks," in *Proceedings of the* 15^{th} *IAPR International Conference on Machine Vision Applications (MVA)*, 2017, pp. 362–365.

[200] Q. N. Vo, S. H. Kim, H. J. Yang, and G. Lee, "Binarization of degraded document images based on hierarchical deep supervised network," *Pattern Recognition*, vol. 74, pp. 568–586, 2018.

[201] I. Pratikakis, K. Zagoris, G. Barlas, and B. Gatos, "ICDAR2017 competition on document image binarization (DIBCO 2017)," in *Proceedings of the* 14^{th} *IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017, pp. 1395–1403.

[202] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proceedings of the* 18^{th} *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Cham: Springer International Publishing, 2015, pp. 234–241.

[203] C. Tensmeyer and T. Martinez, "Document image binarization with fully convolutional neural networks," in *Proceedings of the* 14^{th} *IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017, pp. 99–104.

[204] X. Peng, H. Cao, and P. Natarajan, "Using convolutional encoder-decoder for document image binarization," in *Proceedings of the* 14^{th} *IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017, pp. 708–713.

[205] J. Calvo-Zaragoza and A.-J. Gallego, "A selectional auto-encoder approach for document image binarization," *Pattern Recognition*, vol. 86, pp. 37–47, 2019.

[206] F. Westphal, N. Lavesson, and H. Grahn, "Document image binarization using recurrent neural networks," in *Proceedings of the* 13^{th} *IAPR International Workshop on Document Analysis Systems (DAS)*, 2018, pp. 263–268.

[207] S. He and L. Schomaker, "DeepOtsu: Document enhancement and binarization using iterative deep learning," *Pattern Recognition*, vol. 91, pp. 379–390, 2019.

[208] V. Papavassiliou, T. Stafylakis, V. Katsouros, and G. Carayannis, "Handwritten document image segmentation into text lines and words," *Pattern Recognition*, vol. 43, no. 1, pp. 369 – 377, 2010.

[209] N. Nikolaou, M. Makridis, B. Gatos, N. Stamatopoulos, and N. Papamarkos, "Segmentation of historical machine-printed documents using adaptive run length smoothing and skeleton segmentation paths," *Image and Vision Computing*, vol. 28, no. 4, pp. 590 – 604, 2010.

[210] M. Feldbach and K. Tönnies, "Robust line detection in historical church registers," in *Proceedings of the* 23^{rd} *DAGM Symposium on Pattern Recognition*, B. Radig and S. Florczyk, Eds. Springer Berlin Heidelberg, 2001, pp. 140–147.

[211] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis, "Text line and word segmentation of handwritten documents," *Pattern Recognition*, vol. 42, no. 12, pp. 3169 – 3183, 2009.

[212] G. Seni and E. Cohen, "External word segmentation of off-line handwritten text lines," *Pattern Recognition*, vol. 27, no. 1, pp. 41 – 52, 1994.

[213] T. Varga and H. Bunke, "Tree structure for word extraction from handwritten text lines," in *Proceedings of the* 8^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2005, pp. 352–356.

[214] S. Banerjee, G. Harit, and S. Chaudhury, "Word image based latent semantic indexing for conceptual querying in document image databases," in *Proceedings of the 9^{th} International Conference on Document Analysis and Recognition (ICDAR)*, vol. 2, 2007, pp. 1208–1212.

[215] A. Bhardwaj, D. Jose, and V. Govindaraju, "Script independent word spotting in multilingual documents," in *Proceedings of the 2^{nd} Workshop on Cross Lingual Information Access (CLIA)*, 2008, pp. 48–54.

[216] P. Wang, V. Eglin, C. Garcia, C. Largeron, J. Llados, and A. Fornes, "A coarse-to-fine word spotting approach for historical handwritten documents based on graph embedding and graph edit distance," in *Proceedings of the 22^{nd} International Conference on Pattern Recognition (ICPR)*, 2014, pp. 3074–3079.

[217] Z. Shi, S. Setlur, and V. Govindaraju, "A steerable directional local profile technique for extraction of handwritten Arabic text lines," in *Proceedings of the 10^{th} International Conference on Document Analysis and Recognition (ICDAR)*, 2009, pp. 176–180.

[218] F. M. Wahl, K. Y. Wong, and R. G. Casey, "Block segmentation and text extraction in mixed text/image documents," *Computer Graphics and Image Processing*, vol. 20, no. 4, pp. 375 – 390, 1982.

[219] M. Kassis and J. El-Sana, "Word spotting using radial descriptor," in *Proceedings of the 14^{th} International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2014, pp. 387–392.

[220] M. Khayyat, L. Lam, C. Y. Suen, F. Yin, and C. L. Liu, "Arabic handwritten text line extraction by applying an adaptive mask to morphological dilation," in *Proceedings of the 10^{th} IAPR International Workshop on Document Analysis Systems (DAS)*, 2012, pp. 100–104.

[221] F. Yin and C.-L. Liu, "Handwritten Chinese text line segmentation by clustering with distance metric learning," *Pattern Recognition*, vol. 42, no. 12, pp. 3146 – 3157, 2009.

[222] C.-L. Liu, M. Koga, and H. Fujisawa, "Lexicon-driven segmentation and recognition of handwritten character strings for Japanese address reading," *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 11, pp. 1425–1437, 2002.

[223] T. Wilkinson and A. Brun, "A novel word segmentation method based on object detection and deep learning," in *Proceedings of the* 11^{th} *International Symposium on Advances in Visual Computing (ISVC)*, 2015, pp. 231–240.

[224] K. Chen, M. Seuret, M. Liwicki, J. Hennebert, and R. Ingold, "Page segmentation of historical document images with convolutional autoencoders," in *Proceedings of the* 13^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 1011–1015.

[225] S. Dey, A. Nicolaou, J. Lladós, and U. Pal, "Evaluation of word spotting under improper segmentation scenario," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 22, p. 361–374, 2019.

[226] P. Wang, V. Eglin, C. Garcia, C. Largeron, and A. McKenna, "A comprehensive representation model for handwriting dedicated to word spotting," in *Proceedings of the* 12^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 450–454.

[227] A. Fischer, A. Keller, V. Frinken, and H. Bunke, "HMM-based word spotting in handwritten documents using subword models," in *Proceedings of the* 20^{th} *International Conference on Pattern Recognition (ICPR)*, 2010, pp. 3416–3419.

[228] U.-V. Marti and H. Bunke, "Using a statistical language model to improve the performance of a HMM-based cursive handwriting recognition system," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 01, pp. 65–90, 2001.

[229] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proceedings of the* 13^{th} *European Conference on Computer Vision (ECCV)*, 2014, pp. 346–361.

[230] A. Kumar, C. Jawahar, and R. Manmatha, "Efficient search in document image collections," in *Proceedings of the* 9^{th} *Asian Conference on Computer Vision (ACCV)*, vol. 4843, 2007, pp. 586–595.

[231] J. A. Rodríguez-Serrano and F. Perronnin, "Synthesizing queries for handwritten word image retrieval," *Pattern Recognition*, vol. 45, no. 9, pp. 3270 – 3276, 2012.

[232] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," in *Proceedings of the* 27^{th} *International Conference on Neural Information Processing Systems*, 2014, p. 3581–3589.

[233] J. A. Rodríguez-Serrano and F. Perronnin, "Local gradient histogram features for word spotting in unconstrained handwritten documents," in *Proceedings of the* 11^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2008, pp. 1–6.

[234] ——, "Score normalization for HMM-based word spotting using a universal background model," in *Proceedings of the* 11^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2008, pp. 5–8.

[235] G. Sfikas, G. Retsinas, and B. Gatos, "Zoning aggregated hypercolumns for keyword spotting," in *Proceedings of the* 15^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016, pp. 283–288.

[236] J. Almazan, A. Gordo, A. Fornés, and E. Valveny, "Handwritten word spotting with corrected attributes," in *Proceedings of the* 14^{th} *IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 1017–1024.

[237] S. Ghosh and E. Valveny, "A sliding window framework for word spotting based on word attributes," in *Proceedings of the* 7^{th} *Iberian Conference on Pattern Recognition and Image Analysis (PRAI)*, 2015, pp. 652–661.

[238] B. Gatos, A. Kesidis, and A. Papandreou, "Adaptive zoning features for character and word recognition," in *Proceedings of the* 11^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2011, pp. 1160–1164.

[239] D. Fernández, J. Lladós, and A. Fornés, "Handwritten word spotting in old manuscript images using a pseudo-structural descriptor organized in a hash structure," in *Proceedings of the* 5^{th} *Iberian Conference on Pattern Recognition and Image Analysis (PRIA)*, 2011, pp. 628–635.

[240] J. Almazan, A. Gordo, A. Fornes, and E. Valveny, "Efficient exemplar word spotting," in *Proceedings of the* 23^{rd} *British Machine Vision Conference (BMVC)*, 2012, pp. 67.1–67.11.

[241] T. Mondal, N. Ragot, J.-Y. Ramel, and U. Pal, "Exemplary sequence cardinality: An effective application for word spotting," in *Proceedings of the* 13^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 1146–1150.

[242] R. Jain and C. V. Jawahar, "Towards more effective distance functions for word image matching," in *Proceedings of the* 9^{th} *IAPR International Workshop on Document Analysis Systems (DAS)*, 2010, pp. 363–370.

[243] R. Saabni, "Efficient word image retrieval using earth movers distance embedded to wavelets coefficients domain," in *Proceedings of the* 12^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 314–318.

[244] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the* 27^{th} *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.

[245] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[246] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the* 18^{th} *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, C. Schmid, S. Soatto, and C. Tomasi, Eds., vol. 2, 2005, pp. 886–893.

[247] J. Almazan, A. Fornés, and E. Valveny, "Deformable HOG-based shape descriptor," in *Proceedings of the* 12^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 1022–1026.

[248] T. M. Rath and R. Manmatha, "Word image matching using dynamic time warping," in *Proceedings of the* 16^{th} *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003, pp. 521–527.

[249] V. Frinken, A. Fischer, H. Bunke, and R. Manmatha, "Adapting BLSTM neural network based keyword spotting trained on modern data to historical docu-

ments," in *Proceedings of the* 12^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, IEEE Computer Society, Washington, DC, USA, 2010, pp. 352–357.

[250] A. Toselli and E. Vidal, "Fast HMM-Filler approach for key word spotting in handwritten documents," in *Proceedings of the* 12^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 501–505.

[251] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[252] A. K. Jain, N. K. Ratha, and S. Lakshmanan, "Object detection using Gabor filters," *Pattern Recognition*, vol. 30, no. 2, pp. 295 – 309, 1997.

[253] J. T. Favata and G. Srikantan, "A multiple feature/resolution approach to handprinted digit and character recognition," *International Journal of Imaging Systems and Technology*, vol. 7, no. 4, pp. 304–311, 1996.

[254] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision*, 2004, pp. 1–22.

[255] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the* 19^{th} *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 2169–2178.

[256] F. Perronnin, J. Sanchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Proceedings of the* 11^{th} *European Conference on Computer Vision (ECCV)*. Springer-Verlag, 2010, pp. 143–156.

[257] J. Sanchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the Fisher vector: Theory and practice," *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222–245, 2013.

[258] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Fisher vector faces in the wild," in *Proceedings of the* 24^{th} *British Machine Vision Conference (BMVC)*, 2013, pp. 8.1–8.12.

[259] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios, "Boostmap: A method for efficient approximate similarity rankings," in *Proceedings of the* 17^{th} *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004, pp. 268–275.

[260] T. Adamek, N. E. O'Connor, and A. F. Smeaton, "Word matching using single closed contours for indexing handwritten historical documents," *International Journal on Document Analalysis and Recognition*, vol. 9, no. 2, pp. 153–165, 2007.

[261] S. Colutto and B. Gatos, "Efficient word recognition using a pixel-based dissimilarity measure," in *Proceedings of the* 11^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2011, pp. 1110–1114.

[262] H. A. Glucksman, "Classification of mixed-font alphabets by characteristic loci," in *Proceedings of the IEEE Computer Society Conference*, 1967, p. 138–141.

[263] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proceedings of the* 23^{rd} *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 3360–3367.

[264] S. Sudholt and G. A. Fink, "A modified isomap approach to manifold learning in word spotting," in *Proceedings of the German Conference on Pattern Recognition*, 2015, pp. 529–539.

[265] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[266] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," *CoRR*, vol. abs/1406.2227, 2014. [Online]. Available: http://arxiv.org/abs/1406.2227

[267] Y. Bengio, N. Léonard, and A. C. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *CoRR*, vol. abs/1308.3432, 2013. [Online]. Available: http://arxiv.org/abs/1308.3432

[268] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE Computer Society*, vol. 77, no. 2, pp. 257–286, 1989.

[269] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech & Language*, vol. 16, no. 1, pp. 69–88, 2002.

[270] F. Moreno-Noguer, "Deformation and illumination invariant feature point descriptor," in *Proceedings of the* 24^{th} *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1593–1600.

[271] A. Hast and A. Marchetti, "Putative match analysis: A repeatable alternative to ransac for matching of aerial images," in *Proceedings of the* 7^{th} *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2012, pp. 341–344.

[272] R. B. Girshick, "Fast R-CNN," *CoRR*, vol. abs/1504.08083, 2015. [Online]. Available: http://arxiv.org/abs/1504.08083

[273] T. Wilkinson and A. Brun, "A novel word segmentation method based on object detection and deep learning," in *Proceedings the the* 11^{th} *International Symposium on Visual Computing (ISVC) in Advances in Visual Computing*, 2015, pp. 231–240.

[274] L. Neumann and J. Matas, "Real-time lexicon-free scene text localization and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1872–1885, 2016.

[275] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.

[276] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the* 29^{th} *IEEE Computer Society Conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 2921–2929.

[277] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, 2011.

[278] K. Riesen and H. Bunke, "Approximate graph edit distance computation by means of bipartite graph matching," *Image and Vision Computing*, vol. 27, no. 7, pp. 950 – 959, 2009.

[279] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the* 22^{nd} *IEEE Computer Society Conference on computer vision and pattern recognition (CVPR)*, 2009, pp. 248–255.

[280] Y. Yang, F. Nie, D. Xu, J. Luo, Y. Zhuang, and Y. Pan, "A multimedia retrieval framework based on semi-supervised ranking and relevance feedback," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 723–742, 2012.

[281] B. He, "Rocchio's formula," in *Encyclopedia of Database Systems*. Springer US, 2009, pp. 2447–2447.

[282] G. Giacinto and F. Roli, "Instance-based relevance feedback in image retrieval using dissimilarity spaces," in *Case-Based Reasoning on Images and Signals*, ser. Studies in Computational Intelligence. Springer Berlin Heidelberg, 2008, vol. 73, pp. 419–436.

[283] F. Wolf, P. Oberdiek, and G. A. Fink, "Exploring confidence measures for word spotting in heterogeneous datasets," *CoRR*, vol. abs/1903.10930, 2019. [Online]. Available: http://arxiv.org/abs/1903.10930

[284] R. Shekhar and C. Jawahar, "Word image retrieval using bag of visual words," in *Proceedings of the* 10^{th} *IAPR International Workshop on Document Analysis Systems (DAS)*, 2012, pp. 297–301.

[285] R. Nuray and F. Can, "Automatic ranking of information retrieval systems using data fusion," *Information Processing and Management*, vol. 42, no. 3, pp. 595 – 614, 2006.

[286] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *CoRR*, vol. abs/1507.05717, 2015. [Online]. Available: http://arxiv.org/abs/1507.05717

[287] I. Pratikakis, K. Zagoris, B. Gatos, G. Louloudis, and N. Stamatopoulos, "ICFHR 2014 competition on handwritten keyword spotting (H-KWS 2014)," in *Proceedings of the* 14^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2014, pp. 814–819.

[288] J. Puigcerver, A. Toselli, and E. Vidal, "ICDAR2015 competition on keyword spotting for handwritten documents," in *Proceedings of the* 13^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 1176–1180.

[289] I. Pratikakis, K. Zagoris, B. Gatos, J. Puigcerver, A. H. Toselli, and E. Vidal, "ICFHR2016 handwritten keyword spotting competition (H-KWS 2016)," in *Proceedings of the* 15^{th} *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016, pp. 613–618.

[290] U. V. Marti and H. Bunke, "The IAM-database: an English sentence database for offline handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 39–46, 2002.

[291] W. Pantke, M. Dennhardt, D. Fecker, V. M*ä*rgner, and T. Fingscheidt, "An historical handwritten Arabic dataset for segmentation-free word spotting - HADARA80P," in *Proceedings of the* 14^{th} *International Conference on Frontiers in Handwriting Recognition (ICHFR)*, 2014, pp. 15–20.

[292] S. Yao, Y. Wen, and Y. Lu, "HoG based two-directional dynamic time warping for handwritten word spotting," in *Proceedings of the* 13^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 161–165.

[293] V. Ferrari, F. Jurie, and C. Schmid, "From images to shape models for object detection," *International Journal of Computer Vision*, vol. 87, no. 3, pp. 284–303, 2010.

[294] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 530–549, 2004.

[295] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, no. 2-3, pp. 114–141, 2003.

[296] T. Cootes, E. Baldock, and J. Graham, "An introduction to active shape models," *Image Processing and Analysis*, vol. 328, pp. 223–248, 2000.

[297] B. Gatos, I. Pratikakis, K. A.L., and S. Perantonis, "Efficient off-line cursive handwritten word recognition," in *Proceedings of the* 10^{th} *International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, 2006, pp. 121–125.

[298] G. Horrocks, *Greek: A History of the Language and its Speakers*. John Wiley & Sons, 2009.

[299] A. Kesidis, E. Galiotou, B. Gatos, A. Lampropoulos, I. Pratikakis, I. Manolessou, and A. Ralli, "Accessing the content of greek historical documents," in *Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data*. ACM, 2009, pp. 55–62.

[300] K. Ntzios, B. Gatos, I. Pratikakis, T. Konidaris, and S. J. Perantonis, "An old greek handwritten OCR system based on an efficient segmentation-free approach," *International Journal of Document Analysis and Recognition*, vol. 9, no. 2-4, pp. 179–192, 2007.

[301] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222–245, 2013.

[302] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Proceedings of the* 22^{nd} *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 1778–1785.

[303] J. A. Rodríguez-Serrano and F. Perronnin, "Label embedding for text recognition," in *Proceedings of the* 24^{rd} *British Machine Vision Conference (BMVC)*, 2013, pp. 5.1–5.12.

[304] M. Douze, A. Ramisa, and C. Schmid, "Combining attributes and fisher vectors for efficient image retrieval," in *Proceedings of the* 24^{th} *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 745–752.

[305] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proceedings of the* 27^{th} *IEEE*

Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1090–1097.

[306] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*. Springer, 2009, vol. 2, no. 1.

[307] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proceedings of the* 20^{th} *International Conference on Advances in Neural Information Processing Systems (NIPS)*, 2007, pp. 1177–1184.

[308] A. Protopapas, M. Tzakosta, A. Chalamandaris, and P. Tsiakoulis, "IPLR: An online resource for Greek word-level and sublexical information," *Language resources and evaluation*, vol. 46, no. 3, pp. 449–459, 2012.

[309] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[310] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the* 25^{th} *Advances in Neural Information Processing Systems (NIPS)*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1106–1114.

[311] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the* 3^{rd} *International Conference on Learning Representations (ICLR)*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[312] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the* 3^{rd} *International Conference on Learning Representations (ICLR)*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[313] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the* 32^{nd} *International Conference on Machine Learning (ICML)*, 2015, p. 448–456.

[314] 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, 2014.

[315] A. Shrivastava, A. Gupta, and R. B. Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the* 29^{th} *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 761–769.

[316] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the* 16^{th} *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2223–2232.

[317] Y. Zhang, N. Wang, and S. Cai, "Adversarial sliced Wasserstein domain adaptation networks," *Image and Vision Computing*, vol. 102, p. 103974, 2020.

[318] A. K. Bhunia, A. Das, A. K. Bhunia, P. S. R. Kishore, and P. P. Roy, "Handwriting recognition in low-resource scripts using adversarial learning," in *Proceedings of the* 32^{nd} *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4762–4771.

[319] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the* 27^{th} *International Conference on Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 2672–2680.

[320] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[321] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[322] E. Kordatos, D. Exarchos, C. Stavrakos, A. Moropoulou, and T. Matikas, "Infrared thermographic inspection of murals and characterization of degradation in historic monuments," *Construction and Building Materials*, vol. 48, pp. 1261–1265, 2013.

[323] A. Rhoby, "Text as art? byzantine inscriptions and their display," *Writing Matters: Presenting and Perceiving Monumental Inscriptions in Antiquity and the Middle Ages. Berlin: de Gruyter*, pp. 265–83, 2017.

[324] H. Leung and S. Haykin, "The complex backpropagation algorithm," *IEEE Transactions on signal processing*, vol. 39, no. 9, pp. 2101–2104, 1991.

[325] T. Nitta, "A quaternary version of the back-propagation algorithm," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 5, 1995, pp. 2753–2756.

[326] T. Parcollet, Y. Zhang, M. Morchid, C. Trabelsi, G. Linarès, R. D. Mori, and Y. Bengio, "Quaternion convolutional neural networks for end-to-end automatic speech recognition," *CoRR*, vol. abs/1806.07789, 2018. [Online]. Available: http://arxiv.org/abs/1806.07789

[327] X. Zhu, Y. Xu, H. Xu, and C. Chen, "Quaternion convolutional neural networks," in *Proceedings of the* 15^{th} *European Conference on Computer Vision (ECCV)*, 2018, pp. 631–647.

[328] T. Parcollet, M. Morchid, and G. Linares, "A survey of quaternion neural networks," *Artificial Intelligence Review*, vol. 53, no. 4, pp. 2957–2982, 2020.

[329] Z. Raisi, M. A. Naiel, P. W. Fieguth, S. Wardell, and J. S. Zelek, "Text detection and recognition in the wild: A review," *CoRR*, vol. abs/2006.04305, 2020. [Online]. Available: https://arxiv.org/abs/2006.04305

[330] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, no. 07, pp. 1480–1500, 2015.

[331] M. Liao, B. Shi, and X. Bai, "Textboxes++: A single-shot oriented scene text detector," *IEEE Transactions on Image Processing*, vol. 27, no. 8, p. 3676–3690, 2018.

[332] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao, "Scene text detection via holistic, multi-channel prediction," *CoRR*, vol. abs/1606.09002, 2016. [Online]. Available: http://arxiv.org/abs/1606.09002

[333] M. Liao, Z. Zhu, B. Shi, G. Xia, and X. Bai, "Rotation-sensitive regression for oriented scene text detection," *CoRR*, vol. abs/1803.05265, 2018. [Online]. Available: http://arxiv.org/abs/1803.05265

[334] T. Parcollet, M. Morchid, and G. Linarès, "Quaternion convolutional neural networks for heterogeneous image processing," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8514–8518.

[335] T. A. Ell and S. J. Sangwine, "Hypercomplex fourier transforms of color images," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 22–35, 2007.

[336] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are GANs created equal? a large-scale study," in *Proceedings of the* 32^{nd} *International Conference on Neural Information Processing Systems (NIPS)*, 2018, pp. 700–709.

[337] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CoRR*, vol. abs/1611.07004, 2016. [Online]. Available: http://arxiv.org/abs/1611.07004

[338] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, 2010.

Author's Publications

C1 A. P. Giotis, D. P. Gerogiannis, and C. Nikou, "Word Spotting in Handwritten Text Using Contour-based Models," in Proceedings of the 14^{th} International Conference on Frontiers in Handwriting Recognition (ICFHR), September 2014, pp. $399 - 404$.

C2 G. Sfikas, A. P. Giotis, G. Louloudis and B. Gatos, "Using Attributes for Word Spotting and Recognition in Polytonic Greek Documents," in Proceedings of the 13^{th} International Conference on Document Analysis and Recognition (ICDAR), August 2015, pp. 686–690.

C3 A. P. Giotis, G. Sfikas, C. Nikou, and B. Gatos, "Shape-based word spotting in handwritten document images," in In Proceedings of the 13^{th} International Conference on Document Analysis and Recognition (ICDAR), August 2015, pp. 561–565.

C4 G. Sfikas, A. P. Giotis, G. Retsinas, C. Nikou, "Quaternion Generative Adversarial Networks for Inscription Detection in Byzantine Monuments", In Proceedings of the 2^{nd} International Workshop on Pattern Recognition for Cultural Heritage (PatReCH 2020), held in conjunction with the 25^{th} International Conference on Pattern Recognition (ICPR), January 2021, pp. 171–184.

C5 A. P. Giotis, G. Sfikas, C. Nikou, "Adversarial deep features for weakly supervised KWS", To be submitted in Dec. 2021.

J1 A. P. Giotis, G. Sfikas, C. Nikou and B. Gatos. A Survey of Document Image Word Spotting Techniques. Pattern Recognition, Vol. 68, pp. 310-332, 2017.

Short Biography

Angelos Giotis received his B.Sc. and M.Sc. degrees in Computer Science from the Department of Computer Science and Engineering, University of Ioannina, Greece in 2010 and 2012, respectively. He is a Ph.D. candidate at the same department. He worked as a research associate at the Institute of Informatics and Telecommunications of the National Center for Scientific Research "Demokritos" (Athens, Greece) from 2014 to 2015, in the field of document image processing using keyword spotting techniques to index historical manuscripts. Since 2018, he is working as a research associate and project manager in computer vision projects concerning visual image tracking, as well as age and gender recognition of customers in retail industry using deep learning from RGB and thermal images. His main research interests lie on computer vision, pattern recognition and machine learning with emphasis on document analysis and text understanding.