Gender and age estimation without facial information from still images

A Thesis

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

by

Georgia Chatzitzisi

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN DATA AND COMPUTER SYSTEMS ENGINEERING

WITH SPECIALIZATION
IN DATA SCIENCE AND ENGINEERING

University of Ioannina

July 2020

Examining Committee:

- **Christophoros Nikou**, Professor, Department of Computer Science and Engineering, University of Ioannina (Supervisor)
- **Aristidis Likas**, Professor, Department of Computer Science and Engineering, University of Ioannina
- **Konstantinos Blekas**, Associate Professor, Department of Computer Science and Engineering, University of Ioannina

Dedication

To my family,
for all that I am and hope to be.

Acknowledgements

I would like to express my sincere gratitude to both my supervisor Prof. Christophoros Nikou and co-supervisor Asst. Prof. Michalis Vrigkas for their invaluable guidance and professionalism over the course of this thesis. Their constant encouragement and careful monitoring extensively helped me to stay motivated and diligent. I can confidently say that their scientific approach has been a constant source of inspiration.

I would also like to thank PhD candidate Angelos Giotis for his willing help whenever I had trouble during my experimental setup.

Thanks are also extended to all the academic faculty of the Dept. of Computer Science and Engineering in Ioannina for consistently sharing their knowledge during both my undergraduate and graduate studies. I would specially thank Prof. Aristidis Likas and Assoc. Prof. Konstantinos Blekas for serving in the examining committee of this thesis.

I am also thankful to my colleagues for the creative collaboration and the long stimulating conversations in and out of class. A special expression of gratitude goes to my friends for their continuous support and patience during this journey.

Above all, my heartfelt gratitude is kept for my family, my parents Thomas and Vaya, my brother Nikos and my twin sister Eva. It is their endless support and love that carried me thus far.

Table of Contents

List of Figures

List of Tables

Abstract

Georgia Chatzitzisi, M.Sc. in Data and Computer Systems Engineering, Department of Computer Science and Engineering, School of Engineering, University of Ioannina, Greece, July 2020.
Gender and age estimation without facial information from still images.
Advisor: Christophoros Nikou, Professor.

For many computer vision applications, such as image understanding and human identification, recognizing the gender and age of humans is an essential yet challenging problem. In this thesis, the task is performed on pedestrian images, which are usually captured in-the-wild with no near face-frontal information. In addition, images of humans are acquired under different illumination conditions, yielding poor visual quality, and different camera viewing angles, representing the pedestrian in arbitrary body poses. Moreover, another difficulty in the problem originates from the underlying class imbalance in real examples, especially for the age estimation problem. The first scope of the thesis is to examine how different loss functions in convolutional neural networks (CNN) perform under the class imbalance problem. The loss functions include the cross entropy, which equally weighs each of the classes, the focal loss, focusing on the misclassified examples, and their weighted variants, which weigh the loss function according to the prior class distribution. For this purpose, as a backbone, we employ a commonly used CNN architecture, the Residual Network (ResNet). On top of that, we attempt to benefit from appearance-based attributes, which are inherently present in the available data. We incorporate this knowledge in an autoencoder, which we attach to our baseline CNN in order for the combined model to jointly learn the features and increase the classification accuracy. Finally, all of our experiments are evaluated on the publicly available PETA, RAP v2 and PA100k datasets.

Εκτεταμένη Περίληψη

Γεωργία Χατζητζήση, Μ.Δ.Ε. στη Μηχανική Δεδομένων και Υπολογιστικών Συστημάτων, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πολυτεχνική Σχολή, Πανεπιστήμιο Ιωαννίνων, Ιούλιος 2020.
Αναγνώριση του φύλου και της ηλικίας σε εικόνες χωρίς την πληροφορία προσώπου.
Επιβλέπων: Χριστόφορος Νίκου, Καθηγητής.

Σε πολλές εφαρμογές της υπολογιστικής όρασης, όπως στην περιγραφή εικόνων και στην ταυτοποίηση ανθρώπων, η αναγνώριση του φύλου και της ηλικίας είναι ιδιαίτερα καθοριστική, εν τούτοις αποτελεί ένα πρόβλημα με αρκετές προκλήσεις. Η παρούσα εργασία ασχολείται με εικόνες πεζών, οι οποίες συχνά στερούνται τη σημαντική πληροφορία του προσώπου. Επίσης, οι εικόνες πεζών συχνά προκύπτουν από διαφορετικές συνθήκες φωτεινότητας, οι οποίες παρέχουν φτωχή οπτική ποιότητα και διαφορετικές γωνίες προβολής, οι οποίες οδηγούν σε αυθαίρετες στάσεις σώματος. Επιπλέον, άλλη μία δυσκολία στο πρόβλημα προέρχεται από την δυσαναλογία των κατηγοριών (class imbalance), ιδιαίτερα στο πρόβλημα της εκτίμησης της ηλικίας. Σε αυτή την εργασία, αρχικά, εξετάζουμε πώς διαφορετικές συναρτήσεις κόστους συμπεριφέρονται στα Συνελικτικά Νευρωνικά Δίκτυα (CNN) υπό το πρόβλημα της δυσαναλογίας των κατηγοριών. Σε αυτές τις συναρτήσεις κόστους περιλαμβάνονται η διασταυρωμένη εντροπία (cross entropy), η οποία εξίσου σταθμίζει κάθε κατηγορία, η εστιακή συνάρτηση κόστους (focal loss), η οποία επικεντρώνεται στα εσφαλμένα ταξινομημένα παραδείγματα και στις σταθμισμένες παραλλαγές τους, οι οποίες λαμβάνουν υπ' όψιν την εκ των προτέρων (prior) κατανομή των κατηγοριών. Για το σκοπό αυτό, σαν κορμό χρησιμοποιούμε ένα ευρέως γνωστό Συνελικτικό Νευρωνικό Δίκτυο, το Υπολειπόμενο Δίκτυο (Residual Network). Επιπλέον, αξιοποιούμε την ύπαρξη γνωρισμάτων εμφάνισης, που υπάρχει ήδη στα διαθέσιμα δεδομένα. Εισάγουμε αυτή τη γνώση σε έναν αυτοκωδικοποιητή (autoencoder), τον

οποίο συνδυάζουμε με το Συνελικτικό Νευρωνικό Δίκτυο, ώστε να γίνει μια από κοινού μάθηση των χαρακτηριστικών, που πιθανώς να οδηγήσει σε καλύτερη ακρίβεια κατηγοριοποίησης. Τέλος, όλα τα πειράματα αξιολογούνται στις βάσεις δεδομένων PETA, RAP v2 και PA100k.

Chapter 1

Introduction

1.1 Outline

Gender and age classification has become close-related to many computer vision problems, such as automatic image description, person retrieval and person identification. It has been studied in the literature over the last decade and recently has gained much more interest thanks to the large availability of data. It has been extensively applied to facial images, where face-frontal information is available; similarly, it has been applied to pedestrian images, where a full-body picture is provided; in most cases though, the approaches working with pedestrian images conduct a multi-label classification, where alongside the gender and age other appearance-related attributes are also predicted. Traditional methods, either pertaining to facial images or pedestrian images, use hand-crafted rules to infer the gender and age (or the other multi-label attributes), but these methods cannot adequately capture the interdependence among the attributes. Most recent approaches reoriented towards convolutional neural networks, which directly extract meaningful features for the particular dataset

they are trained on.

In this thesis, we exclusively work with pedestrian images, which are commonly utilized for surveillance systems and require reliable and accurate decisions. To the best of our knowledge, this thesis is the first work to study the classification of solely the gender and age attributes from pedestrian images. In practice, numerous factors affect the classification performance and make the task of gender and age classification far from trivial. Primarily, datasets with gender and age annotations are usually captured in-the-wild, where often no near-frontal information is available. Also, images are taken under different illumination conditions and different camera viewing angles, providing poor visual quality. In some cases, part of the pedestrian's body is occluded by other pedestrians or obstacles or the image background is cluttered. Unfortunately, there exists no single classification approach that can handle successfully all these scenarios.

Following the current trend with the convolutional neural networks, we conduct all of the experiments with the ResNet architecture as the backbone. ResNet is a quite deep convolutional network and has been successfully used in many research problems. Its power comes from its special architecture, which comprises of skip or shortcut connections to jump over the stacked convolutional layers. The motivation behind the shortcut connections is two-fold. First, they speed up training by reducing the impact of vanishing gradients, as there are fewer layers to propagate through, and second, they allow the model to learn an identity function, which ensures that the top layer will preserve what the model previously learned and perform at least as good as the layer below.

Another concern about convolutional neural networks is that they require datasets to be composed of balanced class distributions. However, datasets with gender and age labels are inherently imbalanced. Class imbalance refers to the skewed ratio of the class distributions and can have a negative impact on the model's performance. When the class imbalance is of high degree, most of the times the model predicts the majority class and fails to adequately capture the minority class. To examine how a loss function affects the performance of a model, we study the performance of four different loss functions. The first one is the cross entropy loss, which is commonly used as the default loss function in almost every classification model. Cross-entropy loss penalizes equally each class ignoring the discrepancy between the actual class and the predicted probability. The second loss function is the focal loss, which reshapes the

standard cross entropy loss, such that it dynamically down-weights the contribution of the examples, whose confidence in the correct class increases. The next two are the weighted counterparts of the aforementioned loss functions, which assign an appropriate weight to each class based on the prior distribution of that particular class.

On top of that, we additionally build a model to benefit from the appearance-based attributes present in the available data. Having the ResNet architecture as the baseline, an autoencoder is added in parallel and the whole network is trained end-to-end. We consider that this combined model can learn more powerful relationships among the attributes and potentially lead in a better performance.

1.2 Roadmap

The rest of this thesis is structured as follows: Chapter 2 provides all the theoretical background on which this thesis is based upon, starting from neural networks, to autoencoders and finally to convolutional neural networks. Chapter 3 presents the related work relevant to the task of the gender and age classification. Chapter 4 demonstrates in detail the stages of our methodology and provides a comparative experimental evaluation on three popular datasets. Finally, Chapter 5 summarizes the conclusions and provides interesting future guidelines.

Chapter 2

Deep Neural Networks

2.1 Neural Networks

The human brain has long served as a source of inspiration in many research areas. To better capture the intelligence of the human brain, there have been huge efforts to provide computers with properties of the biological plausible mechanisms. Because the complexity involved in this task makes the design of artificial neural systems by hand impractical, simplified models of neural processing are instead being developed. An information processing paradigm that is loosely modeled after the neuronal structure of the mamalian cortex but on much smaller scale is the (Artificial) Neural Network. Although initially research had been concerned with the ability to simulate brain's skills and operate in a human-like way, neural networks soon reoriented towards improving empirical results.

A neural network involves a network of simple processing elements (artificial neurons or units) carefully arranged in a series of layers. The layer that receives external data is the input layer and the layer on the opposite side that produces the ultimate result is the output layer. In between, there may be zero or more hidden layers, which

in a sense form the majority of the artificial brain. In traditional neural networks, the layers are fully connected: units in successive layers are densely pairwise connected, while units in the same layer share no connections. A simple neural network can be seen in Fig. 2.1.

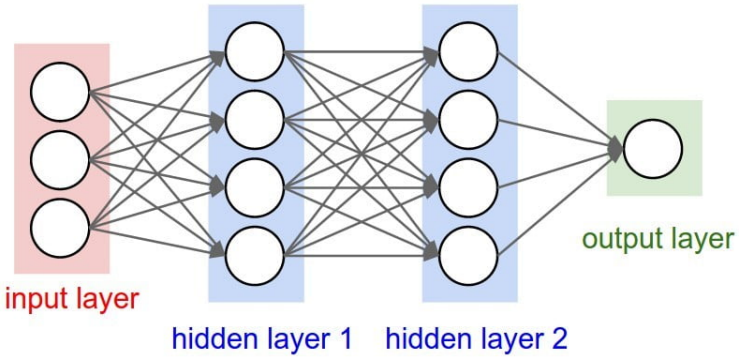


Figure 2.1: A feed-forward neural network[1].

When the input layer is fed with an input vector $\mathbf{x} \in R^d$, the whole neural network is set into action, triggering every layer $k = 1, \cdots, L$ to compute the activation:

$$\boldsymbol{h}^{(k)}(\boldsymbol{x}) = \boldsymbol{g}\left[\boldsymbol{b}^{(k)} + \boldsymbol{W}^{(k)}\boldsymbol{h}^{(k-1)}(\boldsymbol{x})\right], \tag{2.1}$$

where $\boldsymbol{g}(\cdot)$ is an activation function, $\mathbf{W}^{(k)}$ is a weight matrix, which determines the strength of the network's connections in layer k, $\mathbf{b}^{(k)}$ is a bias vector, which determines how easily each unit in layer k fires and $\mathbf{h}^{(k-1)}$ is the activation output vector of the previous layer $k-1$.

Activation functions introduce non-linear properties, facilitating the process of learning more complex mappings from data. For the hidden layers, the choice for which activation function to use is not obvious and requires considerable experience, since the optimal values are not known in advance. Traditional choices are the sigmoid and tanh, but the vanishing gradient problem has made them fall out of popularity. The most recent trend is the ReLU, whose most dominant benefit is the sparsity of activations, forcing units that produce negative values to not activate at all. The output units' activation function provides an easier choice. If the neural network faces a classification problem, the sigmoid or the softmax activation function is used whether it's a binary or multi-class classification respectively. For regression problems, where the outputs are continuous, the linear activation function is more appropriate.

[1]Image taken from `https://cs231n.github.io/neural-networks-1/`.

By adjusting the weight and bias matrices, the neural network can progressively improve its prediction accuracy. The parameter adaptation involves a loss function, which quantifies how the distribution of the predictions matches the distribution represented by the observed data. A loss function is often expressed as the negative log-likelihood of the conditional distribution $p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$ and is decomposed as a sum over training examples:

$$J(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} -\log p(\boldsymbol{y}_i|\boldsymbol{x}_i;\boldsymbol{\theta}))\,, \tag{2.2}$$

where $\boldsymbol{X} = \{x_i, y_i\}_{i=1}^{N}$ is the training dataset of size N of pairs $\boldsymbol{x}_i$ (input vector) and $\boldsymbol{y}_i$ (the corresponding target vector) and $\boldsymbol{\theta}$ are the parameters of the model (including weights and biases). The learning procedure starts with randomly assigned parameters. Then, the training set of data is fed forward repeatedly, and the parameters are modified until the output matches closely with the target values.

Parameter updates are calculated using the gradient descent algorithm by the update rule:

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^{t} - \eta\nabla_{\theta}J(\boldsymbol{\theta}^{t})\,, \tag{2.3}$$

where $\boldsymbol{\theta}^{t}$ indicates the parameters at iteration t and η is the learning rate, representing the step size at the descent direction. The gradient, $\nabla_{\theta}J(\boldsymbol{\theta}^{t})$, is usually calculated over a randomly selected subset of the training data, called a **mini-batch**. The algorithm being responsible for the gradient calculation is called **backpropagation**. It essentially calculates the gradient by taking the loss term found at the output layer at the end of the forward pass, and propagates the error backwards, all the way to the input layer. The gradient descent via backpropagation forces the network to learn the parameters in order to accomplish the desired behavior. On the same time, what the neural network is trying to achieve, is to improve the generalization ability, which refers to the performance difference of the model when evaluated on previously seen data (training data), versus data it has never seen before (testing data).

Earlier versions of neural networks were shallow, composed of one input layer, one output layer, and at most one hidden layer in between. Deep Neural Networks (DNNs) are distinguished from the more commonplace single hidden layer networks by their depth, which is determined by the number of hidden layers. DNNs use multiple hidden layers in order to progressively extract more representative features among the input data. Deep layers identify lower-level features first, and top layers use those features to gradually identify higher-level features by recombining the

identified features from previous layers. By this means, they build a hierarchy of internal representations, which allows them to comprehend the input data better.

2.2 Autoencoders

An autoencoder (AE) is a learning technique, which is used in an unsupervised manner for the task of representation learning. Specifically, an autoencoder is a special type of neural network, which imposes a bottleneck layer in the network, such that a compressed representation is extracted from the original input. The bottleneck layer is also referred to as code or latent representation. Alongside the stage of compression, a reconstruction stage is learned, where the autoencoder tries to generate from the reduced encoding a representation as close as possible to its original input. Therefore, an autoencoder is constituted by two main parts: an encoder that maps the input into the reduced representation, and a decoder that maps the reduced representation to a reconstruction of the original input. If the input features are each independent of one another, this compression and subsequent reconstruction would be a very difficult task. However, if some sort of structure exists within the data (i.e correlations between input features), this structure can be learned and consequently leveraged when forcing the input through the network's bottleneck.

The simplest form of an autoencoder is a feed-forward neural network – having an input layer, an output layer and one or more hidden layers connecting them – where the output layer is forced to have the same number of neurons as the input layer. A simple autoencoder can be seen in Fig. 2.2. As there are many advantages to using deep feed-forward networks, encoder and decoder can individually benefit from deeper architectures.

Formally stated, the encoder and decoder can be defined as transitions ϕ and ψ, such that:

$$\phi : \mathcal{X} \rightarrow \mathcal{F} \,,$$

$$\psi : \mathcal{F} \rightarrow \mathcal{X} \,,$$

$$\phi, \psi = argmin_{\phi,\psi} ||\boldsymbol{X} - (\psi \circ \phi)\boldsymbol{X}||^2 \,.$$

The transitions ϕ and ψ are identical neural networks, in which the weights and biases are initialized randomly, and then updated iteratively during training through

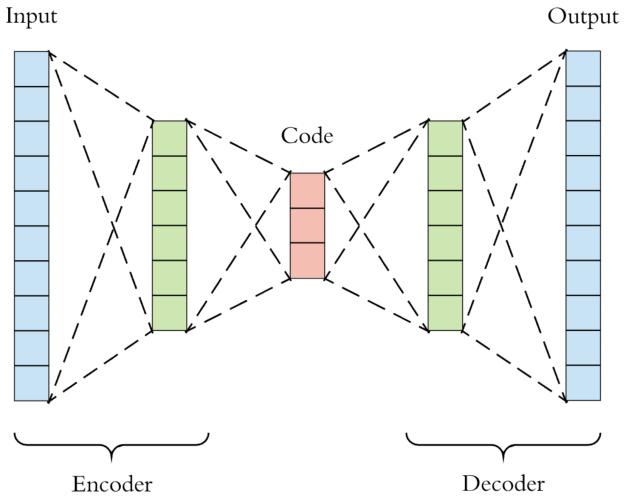


Figure 2.2: A simple autoencoder[2].

backpropagation. The learning process is described as minimizing the reconstruction error:

$$\mathcal{L}(\boldsymbol{x}, \hat{x}) = ||\boldsymbol{x} - \hat{\boldsymbol{x}}||^2 ,$$

which essentially penalizes the reconstruction $\hat{\boldsymbol{x}}$ for being dissimilar from the original input $\boldsymbol{x}$.

Should the feature space $\mathcal{F}$ have lower dimensionality than the input space $\mathcal{X}$, the feature vector $\phi(\boldsymbol{x})$ can be regarded as a compressed representation of the input $\boldsymbol{x}$. This is the case of an undercomplete autoencoder. If the hidden layers are larger than (overcomplete autoencoders) or equal to the input layer, or the hidden units are given enough capacity, an autoencoder can potentially learn the identity function and become useless. Remember that, its ultimate goal is to capture only the salient features of the data and learn rich representations, not to perfectly memorize the input. Hence, the ideal autoencoder balances between, being (1) sensitive enough to the input to accurately build a reconstruction, and (2) insensitive enough to the input for avoiding the input memorization. Therefore, one should be able to tailor the code dimension and the model capacity on the basis of the complexity of the data distribution to be modeled.

The ability to train an autoencoder successfully and yield better compression, is alternatively achieved with variants, called regularized autoencoders. Rather than limiting the autoencoder's capacity by keeping the encoder and decoder parts shallow, regularized autoencoders impose some constraints on the loss function, which

[2]Image taken from `https://towardsdatascience.com/generating-images-with-autoencoders-77fd3a8dd368/`
.

force them to reconstruct the input approximately, preserving only the most relevant aspects of the data. These constraints include sparsity of the representation (sparse autoencoders), smallness of the derivative of the representation (contractive autoencoder), and robustness to noise (denoising autoencoders).

The idea of autoencoders has been popular in the field of neural networks for decades, and the first applications date back to the '80s. Their most traditional application is dimensionality reduction or feature learning, but more recently they have become more widely used for learning generative models of data (variational autoencoders). Autoencoders are effectively used for solving many applied problems nowadays, from face recognition to acquiring the semantic meaning of words in text sequences.

2.3 Convolutional Neural Networks

Convolutional neural networks are a type of neural network specialized in processing data that has a known grid-like topology, such as image data - they can be thought of as a 2D grid of pixel values. They were inspired by the way biological cortical neurons process information and encode image features. Although, its first demonstrations date back to the 1980s, until mid-2010s, research stagnated, as computers lacked sufficient computational resources to process the huge amount of computations derived from images. The next decade followed a significant raise in computational power, which came along with an immense increase on volume, speed and different sources of data, allowing deep convolutional networks to demonstrate a compelling performance. Recently, they have been established as very effective methods and have made prominent contributions across a broad spectrum of applications, ranging from computer vision, pattern recognition, natural language processing and machine translation.

Convolutional networks have also introduced some great novelties like parameter sharing, local connectivity and pooling layers, which will be described in the next subsections.

2.3.1 Local connectivity

Traditional fully connected networks, as the name suggests, use a fully connection pattern, i.e every unit in a particular layer is connected to every unit in the previous layer. However, when dealing with high dimensional input volumes, such as images, even with a shallow neural network, the interconnections are increasing abruptly and this pattern tends to become impractical.

Convolutional networks address this issue by enforcing a local connectivity pattern between units in adjacent layers. Each unit in a convolutional layer is connected to a small number of units in the previous layer via a weight matrix called filter (see Fig. 2.3 (left)). This encourages a sparse connection scheme, which significantly reduces the number of the network's parameters. The spatial extent of this connectivity is called the filter size or the receptive field. The filter is convolved with the input image for every possible receptive field, each time sliding the filter by a number of units at a time, until the entire image is covered. For every convolution performed, the resulted value is stored in a 2D array, called the feature map.

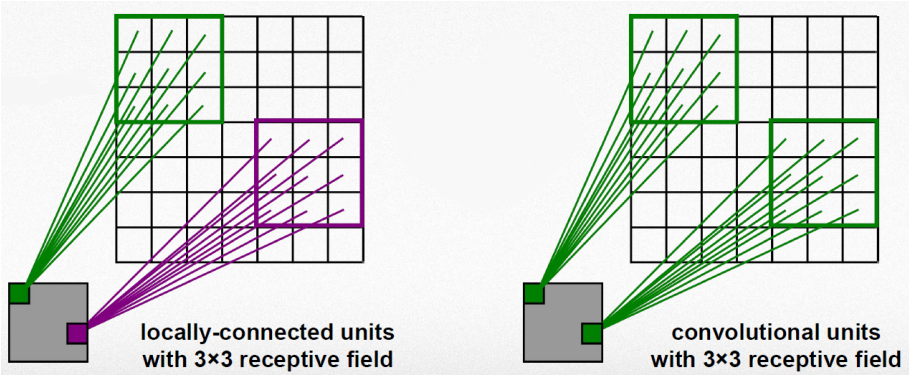


Figure 2.3: Local connectivity (left) and parameter sharing (right).

2.3.2 Parameter sharing

As it was mentioned before, during convolution, a filter is slided through the input volume to produce a feature map. Each number in this filter remains the same for every receptive field it is connected to in this sliding procedure (see Fig. 2.3 (right)). This means that, rather than learning a separate set of parameters for every possible location, only one set is learned. To detect a richer set of representations though, multiple feature maps need to be obtained and thus, multiple filters have to progressively be applied. For example, at the very beginning of a convolutional network, the

filters are basic, usually detecting horizontal or vertical lines. As the information is propagated through the layers, filters become more complex and abstract, detecting more sophisticated patterns, by combining the features obtained from the previous layers. This is known as feature hierarchy.

In addition, parameter sharing provides the network with a property, called equivariance to translation. This indicates that the detected features of a translated image will be moved by the same amount at the resulted feature map. Therefore, the features will be detected in all possible locations.

2.3.3 Non-linearity layer

Most CNNs share the same characteristic: each convolutional layer is directly followed by a nonlinear activation layer acting on the generated feature maps. Since the convolution operation on its own is a linear operation, introducing non-linearity increases the nonlinear properties of the decision function and makes the model capable of learning more complex representations. Because traditional activation functions, like the sigmoid and tanh, saturate and deprive deeper layers from receiving useful gradient information during backpropagation, a revolutionary contribution in deep learning is to utilize the Rectified Linear Unit (ReLU). ReLU applies the non-saturating function $f(x) = \max(0, x)$, which is capable of outputting a true zero value. Zero activation values lead to a sparse representation, which is a desirable property for deep networks, as it can accelerate learning and simplify the model. Also, ReLUs are trivial to implement, requiring only a max function, unlike the sigmoid and tanh activation functions that require the use of an exponential calculation.

2.3.4 Pooling layer

Feature maps summarize the presence of the features with regard to the filters applied by the convolutional layer. However, even small displacements of a feature's position might result in a different feature map. A common approach to addressing this sensitivity includes down-sampling the feature maps, in order to make them more robust to small transformations and distortions. The output then, would be the same no matter the position of the feature within its neighborhood, making the network **local translation invariant**.

The pooling layer operates independently on every feature map and resizes it

spatially. It may include local pooling, which acts in small non-overlapping regions of a feature map (2x2, 3x3 etc) or global pooling, which reduces each feature map to a single value. Pooling also involves selecting a pooling operation, e.g max pooling, average pooling, sum pooling etc. The extracted element is then used to the corresponding position of the subsequent layer, called the pooling layer. The pooling layer serves to progressively reduce the spatial size of the representation and the number of parameters in the network.

2.3.5 Details of a full CNN architecture

A CNN typically consists of two parts, the feature extraction part and the classification part. The feature extraction part stacks convolutional and pooling layers, so as to learn richer representations of the given dataset. These representations are learned during backpropagation w.r.t a loss function that needs to be minimized in order to provide a better discrimination across the classes. In the classification part, the CNN uses the learned features from the previous part to classify the data. This part is fed with the vectorized output of the last convolutional layer and passes this output through a classifier. Depending on the task, the top classifier can be a binary classifier, a softmax layer, a linear or kernel SVM etc.

Another concept that is useful when it comes to training deep convolutional networks is the weight initialization. Weights are often initialized from pre-trained models, which have been previously trained on large datasets. This is known as transfer learning and saves a lot of time, which would be required when training the whole network from scratch. The pre-trained weights are often used as a starting point and the training is resumed on the new dataset to match the new dataset's requirements. Typically, the ImageNet dataset is used, since it is large enough to create features that demonstrate a strong ability to generalize well. Over the years, there have been proposed several CNN architectures for the feature extraction part, such as ResNet, DenseNet, GoogleNet etc and for most of them it is available a pre-trained counterpart.

2.3.6 Regularization

One of the major aspects in training deep neural networks is for the fitted model to be able to both accurately capture the regularities in the training data, but also to generalize well on unseen data. Unfortunately it is typically impossible to do both

simultaneously, and there are often cases where, a model is able to deliver accurate results on the training data, but provides poor results when evaluated on the testing data.

There are two sources of error that prevent models from generalizing beyond the training data. When the model fails to capture the important regularities within the data, is said to have high bias and is underfitting. Contrary, when it perfectly memorizes even noisy or unrepresentative cases, it is said to have high variance and is overfitting. The **bias-variance dilemma** is the conflict in trying to balance both bias and variance, such that training and testing error would be at a minimum. To address this problem, we try to reach the sweet spot using the concept of regularization.

Here, we demonstrate some of the most common regularization strategies including early stopping, dropout and data augmentation.

Early Stopping

When training a neural network, one crucial question is about when should the training stop or how many epochs to use. An epoch is the pass of each example in the training set once. If too few epochs are used, the model might underfit; if too many epochs are used, it might overfit.

Early stopping attempts to remove the need to manually set the number of epochs by estimating and monitoring the generalization error on a usually small held-out portion of the training data, called the validation set. The optimizer is halted to the point where the generalization performance degrades or stops improving (Fig. 2.4). Although this technique requires constant evaluations of the current model on the validation set, which can be computationally expensive, it is useful for either speeding the learning procedure or improving the generalization performance, whichever is more important in the particular situation.

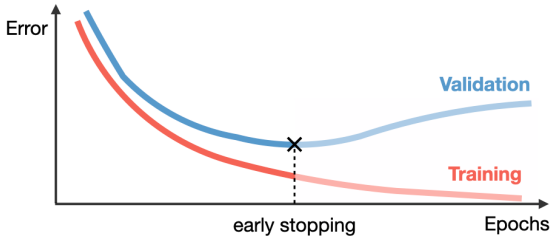


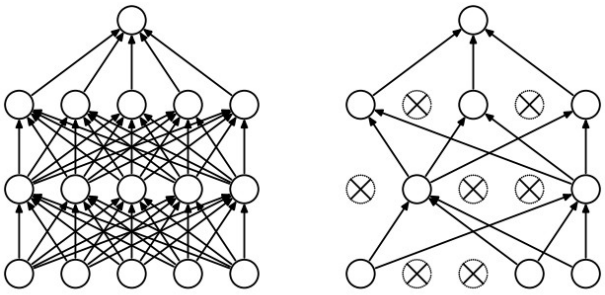
Figure 2.4: Illustation of early-stopping[3].

[3]Image taken from `https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-deep-learning-tips-and-tricks`.

The partition for the validation set should be done carefully, because it might lead in an improper stopping decision or it might deprive the model from valuable information. To prevent such scenarios, cross-validation techniques are being used, where in one round of cross-validation the training set is partitioned into complementary subsets, performing the training in one of them and the evaluating procedure on the other. To reduce the variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are combined over the rounds to estimate the final predictive model.

Dropout

Dropout is a very effective, yet simple, way of performing model averaging, which involves training more than one neural networks on the same dataset. The final prediction, then is made by combining the predictions of the ensemble of the trained models to yield better predictive performance.



(a) Standard Neural Net (b) After applying dropout.

Figure 2.5: **Left**: A standard neural network with 2 hidden layers **Right**: An example of a network produced by applying dropout on the network on the left. Crossed neurons have been dropped out [4].

More technically, at each training stage, individual units are randomly ignored with probability p, so that a reduced neural network is left; incoming and outgoing edges to a dropped-out unit are also removed (see Fig. 2.5). At testing phase, the entire neural network is considered and each activation is reduced by a factor of p to account for the missing activations during training.

In fully connected layers, since all the weights are learned together, it is likely for some units to develop co-dependency among each other, which curbs the individual power of each unit. In such a scenario, only a fraction of the connections is trained

[4] Image taken from [?]

and the rest stops participating. Dropout has the effect of making the training process noisy, breaking situations where the neural network layers co-adapt, which in turn makes the model more robust.

Data augmentation

Deep learning models require big datasets to obtain a good generalization ability. However, assembling enormous datasets can be a very daunting task due to the manual effort of collecting and labeling data. Data augmentation has been developed under the assumption that new artificial instances can be extracted from the original dataset. The earliest demonstrations showing the effectiveness of data augmentation come from simple image transformations such as flipping (horizontal or vertical), cropping, rotation, translation, shearing etc. Additional augmentations that have proven to be effective are color space augmentations - which result in lighting alternations in the image color channels - and noise injections - which consist of injecting random pixel values to the original pixel values (usually drawn from a Gaussian distribution). Another interesting augmentation technique is random erasing, which works by randomly selecting a patch of an image and masking it with random pixel values. It ultimately forces the model to learn more descriptive features about an image, preventing it from overfitting to a certain visual feature.

An important consideration with respect to the augmentations listed above is the safety of their application, which refers to its likelihood of preserving the label post-transformation. For example, in an object detection application, where the model tries to approximate the position of an object, a translation to the original image causes the position of the object to alter. The same geometric transformation should also be applied to the target position values such that in the translated image the translated target values match the new object position. Or, when decreasing the pixel values of an image to simulate a darker environment, it may become impossible to detect the object in the image. Such scenarios would require refined labels post-augmentation and constructing refined labels for every non-safe data augmentation is a computationally expensive process.

Therefore, the choice of what type of augmentation to apply can be tricky and time-consuming to tune by hand for a new dataset or task, and can have a large effect on both the model's generalization ability and efficiency.

Chapter 3

Related Work

3.1 Introduction

Over the last decade, the rate of image availability has grown at a nearly exponential rate. This new-found wealth of data has empowered computer scientists to tackle problems in computer vision that were previously either irrelevant or intractable. An example is the automatic gender and age classification task, which has become closely-connected to an increasing number of applications. Particularly, for many practical applications, relying on humans to supply demographic information from images is not feasible. Hence, there has been a growing interest in automatic extraction of demographic information from images, either they are facial images or images of pedestrians captured in real world scenarios.

Applications that benefit from this technology have a broad scope and the potential to make a large impact. Particularly, the rise of social media platforms has led to an abundance of facial image uploads on the web. The objective has extended from detecting and counting the faces that appear in an image to classifying the characteristics of these faces. These characterists include gender and age, which both perceive to be of the most distinguishable traits across people. Social media platforms, like

Facebook and Instagram, could use the gender and age information to better infer the context of an image or to provide a targeted advertisement and/or recommendation. In security and access control, an automatic age estimation system can be used to prevent underage people from purchasing alcohol or cigarettes from vending machines or deny children access to inappropriate web content.

Gender and age classification from facial images is an inherently challenging problem. The main reason lies in the nature of the data associated with facial images. While general object classification tasks can often have access to hundreds of thousands, or even millions of images, datasets with gender and/or age labels are considerably smaller in size. This is the case because in order to have labels for such images, it is required to have access to the personal information of the subjects appearing in the images. Namely, the gender and the date of birth are required, with the last one to be a rarely-released piece of information. Additionally, in real-world applications, images can be subject to various lighting, angle, focus, occlusion, etc conditions, hindering the systems from achieving good performance.

Another application domain that benefits from automatic gender and age classification is authorization and surveillance monitoring in malls, customs, banks etc. In a large-scale surveillance system, gender and age information could be useful for security officers searching for suspects or terrorists.

Images for surveillance applications are taken at far distance and often are lacking face-frontal information. Thus, gender and age recognition has to be performed using the full body appearance, with the absence of critical face or close-shot visual information. The inherent visual ambiguity and the poor quality of visual features originating from the far view field make the problem of gender and age classification even more challenging. There are also variations originating from the camera such as variations in illumination and camera viewing angle.

In the following two sections, we review the approaches that have studied the problem of gender and age classification. In Section 3.2 we focus on approaches that use facial images and in Section 3.3 on approaches that use pedestrian images.

3.2 Related work applied to facial images

The problem of automatically extracting the gender and age information from facial images has received increasing attention in recent years and many methods have been proposed. Most have employed classification schemes particularly for gender or age, but few of these examine the problem of predicting both the attributes simultaneously.

3.2.1 Gender classification

Several studies have shown that humans can determine a person's gender easily and accurately using only facial information. However, this is not a trivial task for machine-based gender recognition systems and remains as a challenge to date. Some facial features have semantic structures that may mislead the classification process. In the case where the facial features are quite ambiguous, the visual information surrounding the face in an image are of high importance in making a more accurate classification. Vision-based gender classification methods are usually based on extracting features from the given face image and then use these features to train a classifier that outputs the predicted gender. Such methods can be divided into two main categories: geometric-based and appearance-based.

The geometric-based techniques extract and utilize facial landmark information from the given images to predict the gender [1, 2, 3]. These models maintain a certain geometric relationship between different face parts and discard facial texture information in the whole modeling process. Thus, they are highly sensitive to imaging geometry and face alignment.

On the other hand, appearance-based methods rely on extracting features from either or both the whole face image (holistic features) and regions of the face image (local features). Some earlier researchers extracted pixel intensity values as well and then fed these values to the classifiers [4]. Li et al. [5] introduced a method based on five individual facial features in addition to the hair and clothing of the person and then used multiple SVMs to infer the gender. Shan et al. [6] employed Local Binary Patterns (LBPs) to describe faces, an AdaBoost classifier to select the discriminative LBP features and a SVM applied to the boosted LBP features to classify the gender. The subspace transformation was also performed to either reduce dimensions or explore parts of the underlying structure of the raw image values [7]. More recently, Geetha et al. [8] proposed a method to extract texture features from three discrimi-

nating levels (global, directional and regional) and then fed a kernel-based SVM for the classification stage.

CNNs have been applied in gender classification as well. Mansatet et al. [9] introduced a Local-DNN which was trained on local features obtained from overlapping patches. In order to explore how face images behave under occlusions, Juefei-Xu et al. [10] utilized multiple levels of blurring to train a deep CNN in a progressive way.

3.2.2 Age classification

Age classification from facial images remains a challenging problem for a number of reasons. The face aging process generally follows some common aging modes. During the growth stage of children, the biggest change is the shape change caused by the growth of the skull. The aging process in adulthood is mainly reflected in changes in facial skin texture such as the appearance and deepening of wrinkles, loose skin, increased spots, etc. However, due to the complex facial features and slow aging process, the degree of aging depends not only on age development, but also due to intrinsic (gender, race, genes, etc) or extrinsic factors (living habits, health status, environment, etc). In addition, the collection of face age images is very burdensome. The existing public face age datasets have many problems such as an imbalance in age, gender, and ethnicity, which makes it difficult to meet the requirements of most research work. The above reasons mean that the research on face age estimation still faces great challenges and it is an active research topic.

Age estimation is a special pattern recognition task where age labels can be viewed as a class or a set of sequential values. When age labels are viewed as classes, age estimation is approached as a classification problem, whereas when age labels are viewed as sequential chronological series, regression approach is used for age estimation. Hybrid approaches can also be employed, where both classification and regression techniques are integrated, mostly hierarchically, to find the relationship between extracted feature vectors and age labels. Despite our focus on age group classification rather than precise age estimation (i.e regression), here we include methods designed for either task.

An early approach to age estimation was by Kwon et al. [11], who used anthropometric models to extract facial age features. Once facial features (e.g eyes, nose, mouth, chin, etc) were localized, the distance ratios between them were measured.

Using these measurements, they roughly divided ages into three age categories, according to hand-crafted rules based on craniofacial development theory. Lanitis et al. [12] applied the Active Appearance Model (AAM) to provide an aging pattern representation. Based on this representation, a quadratic regression function was used for age estimation. Later, in [13], the aging process was simulated using AMM for the same individual with a series of age-asceding facial images, so that specific models associated with different people's aging processes could be constructed. Geng et. al. [14] proposed a concept of aging patterns subspace (AGES) to interpret the long-term aging subspace of a person. Since the available images for a specific person are typically very limited, many researchers focused on developing non-personalized approaches instead. Guo et al. [15] introduced manifold learning with locally adjusted robust regressors to learn a common aging trend for each age. Gunay et al. [16] applied Local Binary Patterns (LBPs) to small regions of the face image and concatenated the spatial LBP histograms from the different regions into a feature vector to be used as a face descriptor. Guo et al. [17] proposed Biological Inspired Features (BIF) in order to model a face image as a hierarchy of increasingly sophisticated representations.

In recent years, deep learning technologies, such as CNNs, have been gradually applied to age estimation and have achieved better results than manually designed features. Yi et al. [18] introduced a relatively shallow CNN architecture and a multi-scale analysis strategy to end-to-end learn the age label of a face image. Wang et al. [19] trained a deeper CNN for extracting features from different layers. Niu et al. [20] formulated the age estimation problem as an ordinal regression problem using a series of binary classification tasks, which were jointly optimized by a multiple output CNN architecture. Hu et al. [21] presented a CNN architecture, in which they incorporated the KL divergence to insert age difference information for each pair of images. Chen et al. [22] proposed a ranking-CNN framework, in which a series of basic CNNs were employed and their binary outputs were aggregated. A separate CNN for each ordinal age group was learned, allowing each sub-CNN to capture different patterns for different age groups.

3.2.3 Age and gender classification

Several hybrid methods predicting age and gender simultaneously with other facial attributes or not have also been reported in the literature. A combined framework for

age and gender was introduced in [23]. The model was a viewpoint invariant derived from local scale-invariant features, which were probalistically quantified in terms of their occurrence, appearance, geometry and association with visual traits of interest. In [24], Eidinger et al. employed LBP descriptor variations and a dropout-SVM classifier, in which different features were randomly omitted from the linear-SVM classification process. Levi et al. [25] were the first to use a CNN architecture for the problem of age and gender classification with a relative shallow architecture. Rodriguez et al. [26] introduced the visual attention mechanism to discover the most informative and reliable parts in a face image for improving age and gender classification. Dual et al. [27] integrated a CNN for feature extraction and an Extreme Learning Machine (ELM) for classifying the intermediate results.

3.3 Related work applied to pedestrian images

In surveillance scenarios, any information that can be extracted from an image is crucial and can prove to be useful in identifying a suspect. Hence, together with age and gender, datasets also provide other attributes related to pedestrian clothing and appearance. Predicting jointly all attributes adds an extra difficulty due to multi-factor variations. There exists large intra-class diversity among different images for the same attribute. Also, because images are captured in-the-wild and are unconstrained, there is an inherent visual ambiguity. For example, part of the pedestrian's body is often occluded by obstacles, or some of the attributes to be predicted may be partly visible due to the pedestrian's posture. All approaches presented here perform a multi-attribute classification by attempting to predict all possible attributes in an image.

Early approaches to attribute recognition involved heavy use of hand-crafted features. Features included color histograms [28], HOG, textures and ensembles of localized features (ELF)[29]. A common formula for the task of attribute recognition was to use these hand-crafted features in combination with a linear SVM. However, these early approaches suffer from several problems. First, it is difficult to craft a set of features that perform well on such a wide variety of attributes in a wide variety of situations. Second, the SVM optimizes each attribute independently and lacks a way to learn relationships among attributes. For example, attributes such as 'female' are closely related to other attributes such as 'long hair' or 'skirt', yet the SVM is not

capable of learning this kind of relationship. The thing is, the performance would be significantly improved by leveraging a classifier that can benefit from the interdependence of these three attributes. Finally, a large class imbalance definitely hinders the classifier's performance as the hyperplanes are overwhelmed by a large ratio of negative to positive samples.

Later approaches reoriented towards CNNs, which showed that end-to-end learning could mitigate some of the limitations associated with SVMs and hand-crafted features. Features in CNNs are extracted directly from the training data, hence there is no need to hand-craft features for each dataset and attribute. The feature extractors and the classifier parameters are optimized together in an end-to-end fashion for the particular dataset and set of attributes. Moreover, a considerable advantage of CNNs over the traditional SVMs is their ability to learn relationships among attributes.

The remainder of this section gives an overview of the widely known approaches that use, solely or partially, a CNN architecture. These approaches are usually divided into three categories, namely part-based, attention-based and relation-based. Starting from approaches that utilize a simple CNN in the next paragraph, we subsequently refer to more advanced approaches belonging to the three aforementioned categories.

Sudowe et al. [30] (ACN) describe a pedestrian attribute CNN which is trained with one loss per attribute. A similar approach with individual attribute losses which are manually restricted to relevant body parts is described by Zhu et al. [31]. Li te al. [32] (DeepMar) train a CNN with a single, weighted loss which includes all attributes and applies weights based on each attribute's label imbalance. These algorithms all take the whole image as input and conduct multi-task learning for pedestrian attribute recognition. They all attempt to learn more robust feature representations using feature sharing, end-to-end training or multi-task learning. The main advantage of these models is that they are simple, intuitive and highly efficient, characteristics particularly important for practical applications. However, the performance of such models is still limited due to the lack of consideration of fine-grained features.

3.3.1 Part-based methods

It is yet another popular idea to make use of part-based information to jointly utilize global and fine-grained local features. The localization of body parts is achieved via an external part localization module, such as body-part detection, pose estimation,

poselets or region proposals. Several approaches have been developed towards this direction. Zhang et al. [33] explore poselets for part localization, by decomposing the objects into their canonical poses, and incorporate these normalized parts into a CNN model to capture pose-normalized representations. Zhu et al. [34] divide the whole image into 15 rigid patches and fuse features from different patches. In [35], Yu et al. propose a CNN based approach which relies on multi-level features to recognize and localize pedestrian attributes. Li et al. [36] explore the deformable body structure knowledge, i.e. human pose, and body parts localization, using image-level supervision, to adaptively locate informative image regions. Liu et al. [37] explore attribute regions in a weakly supervised manner while they assign attribute regions to some fixed proposals. Tang et al. [38] localize the attribute-specific regions at multiple feature levels and apply a feature pyramid architecture to enhance the attribute localization and region-based feature learning in a mutually reinforcing manner.

Part-based methods rely either on predefined rigid parts or on sophisticated part localization mechanisms, which are less robust to pose variations and require extra computational resources. Moreover, most of them just fuse the part-based features with global features, which still fail to indicate the attribute-region correspondence.

3.3.2 Attention-based methods

Visual attention mechanism has also been introduced in pedestrian attribute recognition, but the existing works are still limited. These methods usually generate attention masks from certain layers and then multiply them to the corresponded feature maps so as to extract the attentive features. Sarfraz et al. [39] introduce a model with view guidance to make view-specific attribute predictions in order to overcome the variance of patterns in different angles. Liu et al. [40] combine a plain CNN architecture with an attentive feature network comprising of multi-directional attention modules applied to different semantic levels. Zhu et al. [41] propose a spatial regularization network to associate image regions to each attribute. In [42], Sarafianos et al. extract and aggregate visual attention masks at different scales and establish a weighted-variant of the focal loss to handle both under-represented or uncertain attributes.

Although with attention-based methods recognition accuracy has been improved, these methods are attribute-agnostic and fail to take the attribute-specific information

into consideration. It is ambiguous which mask encodes a given attribute's location, and there is no specific mechanism that guarantees the correspondences between attributes and attention masks.

3.3.3 Relation-based methods

Other approaches are regarded as relation-based and exploit semantic relations to assist attribute recognition. Wang et al. [43] propose a CNN-RNN based framework to exploit the interdependence and correlation among attributes. Zhao et al. [44] divide the attributes into several groups and attempt to explore the intra-group and inter-group relationships. In [45], Sarafianos et al. leverage curriculum learning, by learning first the strongly correlated attributes in a multi-task learning setup and then use transfer learning to additionally learn the weakly-correlated attributes.

Relation-based methods require manually defined rules, e.g. prediction order, attribute groups, which are hard to determine in real applications.

Chapter 4

Gender and age estimation without facial information from still images

4.1 Methodology

In this work, we focus on recognizing the gender and age attributes, which are physical, adhered human characteristics belonging to the soft biometrics. Gender and age are of the most recognizable human attributes and are established by humans with the aim of distinguishing individuals e.g. in suspect descriptions. Since, they are meaningful semantic representations understood both by humans and computers, we strive to build a powerful framework to address the challenge of automatically recognizing these two attributes. Our method relies on still images of pedestrians without the presence of clear-shot face-frontal information. Such images are usually captured in real surveillance scenarios and lack good visual quality. We opt for a three-stage strategy; we first only consider the problem of gender classification, then the problem

of age classification and finally the problem of multi-label classification, where we try to predict both attributes simultaneously. The main challenge we focused on is the class imbalanced distributions, which are inherently present in the available datasets. For all experiments, we use the ResNet architecture as the backbone to investigate how four different loss functions perform under the class imbalance problem. Finally, we build a model, adding an autoencoder on top of the ResNet, which we feed with appearance-based attributes. We consider that a combined model can leverage this additional information to make more accurate predictions. More details about the network architectures, the loss functions and the evaluation process are provided in the following sections.

4.1.1 The ResNet architecture

As the backbone of our experiments, we choose the ResNet50 architecture, which, in [46], has been shown to outperform other CNN variants. ResNet is a very deep convolutional network and was implemented to support the idea that increasing the network's depth does not work by simply stacking convolutional layers. Deep networks with stacked convolutional layers are hard to train because of the notorious vanishing gradient problem, which indicates that as the network goes deeper its performance gets saturated or even starts degrading rapidly.

ResNet is composed of stacked residual blocks, which have a double convolution residual leg and a direct input-to-output shortcut connection. At the end of every such block, the features from both branches are merged. By this means, the training of one or more layers is skipped, which speeds up the training procedure. At the same time, shortcut connections enforce the model to learn an identity function, which ensures that the top layer will preserve what the model previously learned and perform at least as good as the layer below. In addition, ResNet was the first architecture to introduce a heavy use of batch normalization. Batch normalization normalizes the output of an activation layer by subtracting the batch mean and dividing by the batch standard deviation. It can be thought of as performing pre-processing to the activations and can increase the stability of the network. The full ResNet50 architecture is depicted in Fig.4.1.

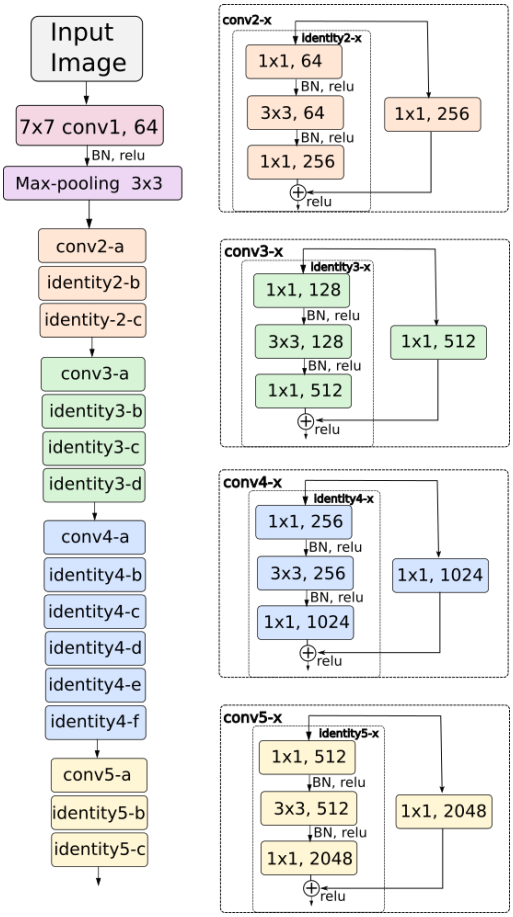


Figure 4.1: The ResNet50 architecture.

4.1.2 Gender classification

Consider there are N pedestrian images $x_i, i = 1 \cdots N$, labeled with the gender attribute $y_i \in \{0, 1\}$. Formally stated, the problem of gender recognition is a binary classification problem, in which given an input pedestrian image, we try to recognize the pedestrian's gender, with a 1 denoting a female and a 0 a male. The features extracted from the ResNet are pooled and passed through a binary classifier to determine the pedestrian's gender. Our approach employs a global average pooling, which takes the average of each of the feature maps obtained from the ResNet. Fig. 4.2(a) illustrates this model's architecture. The output of the model is one neuron with the sigmoid activation function, representing the probability of the pedestrian being a "female". Thresholding this value at 0.5, we obtain the final prediction.

The choice of the appropriate loss function is affected by the class imbalance, as the ratio of class 0 over class 1 is often relatively large. In the presence of class imbalance, the loss due to the frequent class can dominate total loss and cause instability. Hence, in order to see how different loss functions perform under the class imbalance

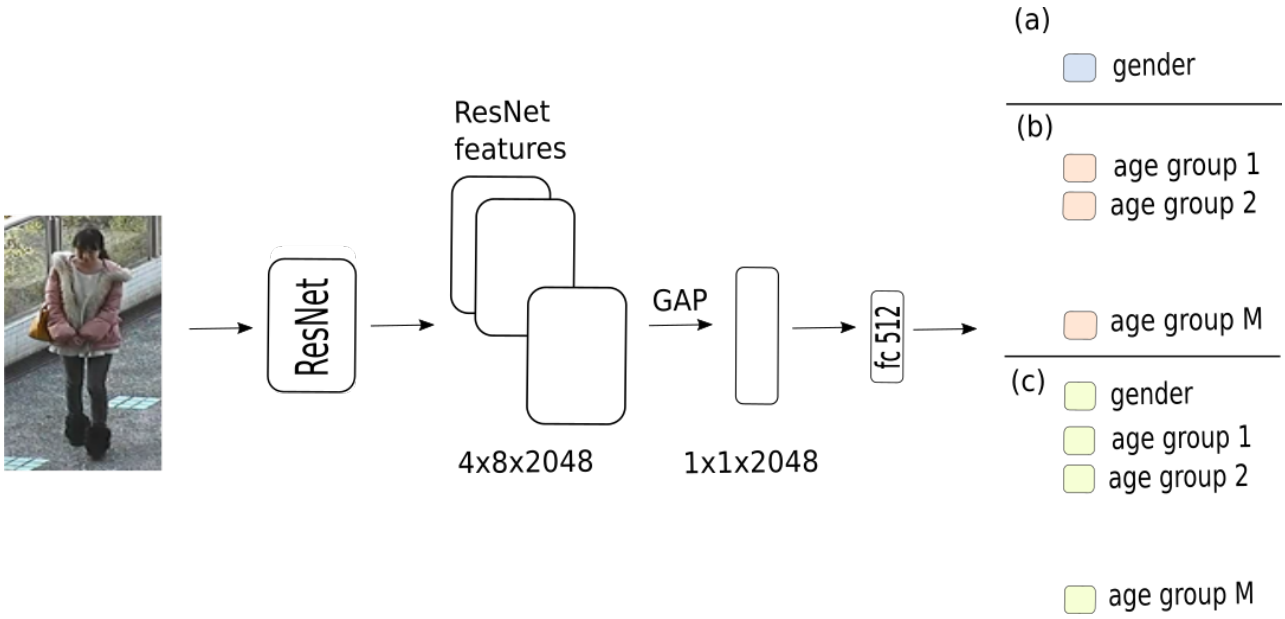


Figure 4.2: The model for (a) gender classification, (b) age classification and (c) multi-label classification.

problem, we explore the performance of four different loss functions. The first one is the standard binary cross entropy, formulated as:

$$L_{bce} = -y \log \hat{y} - (1-y) \log(1-\hat{y}) \tag{4.1}$$

$$= \begin{cases} -\log \hat{y} & \text{if} \quad y = 1 \\ -\log(1-\hat{y}) & \text{if} \quad y = 0 \end{cases}, \tag{4.2}$$

where y and $\hat{y}$ are the ground truth and predicted labels respectively. Such a loss function ignores completely the class imbalance, assigning the same weight to the two classes. Aiming to alleviate this problem, we employ a weighted-variant of the binary cross entropy, called the binary focal loss, defined as:

$$L_{bfl} = -y\,(1-\hat{y})^{\gamma} \log \hat{y} - (1-y)\,\hat{y}^{\gamma}\, \log(1-\hat{y}) \tag{4.3}$$

$$= \begin{cases} -(1-\hat{y})^{\gamma}\, \log \hat{y} & \text{if} \quad y = 1 \\ -\hat{y}^{\gamma}\, \log(1-\hat{y}) & \text{if} \quad y = 0 \end{cases}, \tag{4.4}$$

where $\gamma \geq 0$ is a focusing parameter. Focal loss is a cross-entropy loss that weighs the contribution of each example to the loss based on the classification error. When an example is classified correctly, its contribution to the loss decreases, as the modulating factor $(1-\hat{y})^{\gamma} \to 0$ if $y = 1$ (or $\hat{y}^{\gamma} \to 0$ if $y = 0$). When an example is misclassified, the

modulating factor $(1-\hat{y}_i)^\gamma \to 1$ if $y = 1$ (or $\hat{y}^\gamma \to 1$ if $y = 0$) and the loss is unaffected. With this strategy, the loss is made to implicitly focus on the problematic cases by extending the range in which an example receives low loss. For instance, with $\gamma = 2$, an example classified with $\hat{y} = 0.9$ would have $100\times$ lower loss and with $\hat{y} = 0.968$ it would have $1000\times$ lower loss compared with cross entropy. Finally, we also employ two variants of the binary cross entropy and the binary focal loss. The two variants are the weighted binary cross entropy and the weighted binary focal loss and are respectively defined as:

$$L_{wbce} = -w \left[y \log \hat{y} + (1-y) \log(1-\hat{y}) \right], \tag{4.5}$$

$$L_{wbfl} = -w \left[y\,(1-\hat{y})^\gamma \log \hat{y} + (1-y)\,\hat{y}^\gamma \log(1-\hat{y}) \right], \tag{4.6}$$

$$w = \begin{cases} \frac{1}{1-p_f} & \text{if} \quad y = 0 \\ \frac{1}{p_f} & \text{if} \quad y = 1 \end{cases}, \tag{4.7}$$

where w is the loss weight according to the gender label and p_f is the proportion of the females in the training set.

For the problem of gender recognition, we additionally develop a model that can benefit from annotations already present in the available data. Specifically, instead of treating an image independently, we consider inference with the help from additional attributes. We claim that introducing this kind of information into a model, gender prediction would be performed with more confidence. For example, all datasets provide attributes related to pedestrian appearance (e.g long hair, short hair etc), upper and lower body clothing style (casual or formal, t-shirt, jeans, skirt etc) and accessories (hat, backpack etc). We incorporate these attributes in a binary vector, in which a 1 indicates the presence and a 0 indicates the absence of that particular attribute. Hence, each pedestrian image x_i is assigned with a L-length binary vector y_i, where $y_{il} \in \{0, 1\}$ denotes the presence of the l-th attribute in x_i. Then, we use an autoencoder to learn the "compressed" representation of the original attribute input vector. Since the input to the autoencoder is a small-length vector, we wanted to keep the autoencoder's architecture simple. Hence, the autoencoder is a one-hidden-layer neural network, with the size of the "bottleneck" layer and the size of the output layer to be the same as the size of the input vector ($= L$). The problem that the autoencoder

is trying to solve is a multi-label classification problem hence, we use the sigmoid activation function for each of the output neurons. We also employ the binary cross entropy loss of Eq. (4.2) and the binary focal loss of Eq. (4.4) slightly modified to account for all L attributes:

$$L_{ae} = -\sum_{l=1}^{L} \Big[y_l \log \hat{y}_l + (1 - y_l) \log(1 - \hat{y}_l) \Big] , \tag{4.8}$$

$$L_{ae} = -\sum_{l=1}^{L} \Big[y \, (1 - \hat{y})^{\gamma} \log \hat{y} + (1 - y) \, \hat{y}^{\gamma} \, \log(1 - \hat{y}) \Big] , \tag{4.9}$$

where L is the number of attributes and y_l, $\hat{y}_l$ are the ground truth and predicted labels for the l-th attribute. The features from the autoencoder's bottleneck layer are concatenated with the features obtained from the ResNet's last fully connected layer to form a new model. At the top, we add a binary classifier and we train this combined model, which we call ResNet+AE, with the best performing loss function from the sinlge-ResNet architecture. The illustration of the ResNet+AE model is depicted in Fig. 4.3(a). The only difference here is that the final prediction is affected both by the features obtained from the ResNet and the autoencoder. This combined model is trained end-to-end and the overall loss is a combination of the autoencoder's loss and the loss arising from the ResNet:

$$L_{combined} = L_{ae} + L_{ResNet} , \tag{4.10}$$

where L_{ae} is one of the Eq. (4.8), (4.9) and L_{ResNet} is one of the Eq. (4.2), (4.4), (4.5), (4.6), whichever performs the best in the case of the gender classification.

4.1.3 Age classification

At the second stage, we study the problem of age recognition, where the model should predict 1 of M classes corresponding to M age categories. Formally stated, the problem of age recognition is a multi-class classification problem, in which given an input pedestrian image, we try to recognize the pedestrian's age range. The age label vector is a one-hot vector y, and each element of that vector is represented as y_m, $m = 1, \cdots, M$ and $y_m \in \{0, 1\}$. For example, an age label vector [0,0,1,0,0] means that the pedestrian to whom corresponds this label vector belongs to the third age group in a 5-class problem. We again employ the ResNet architecture, with the

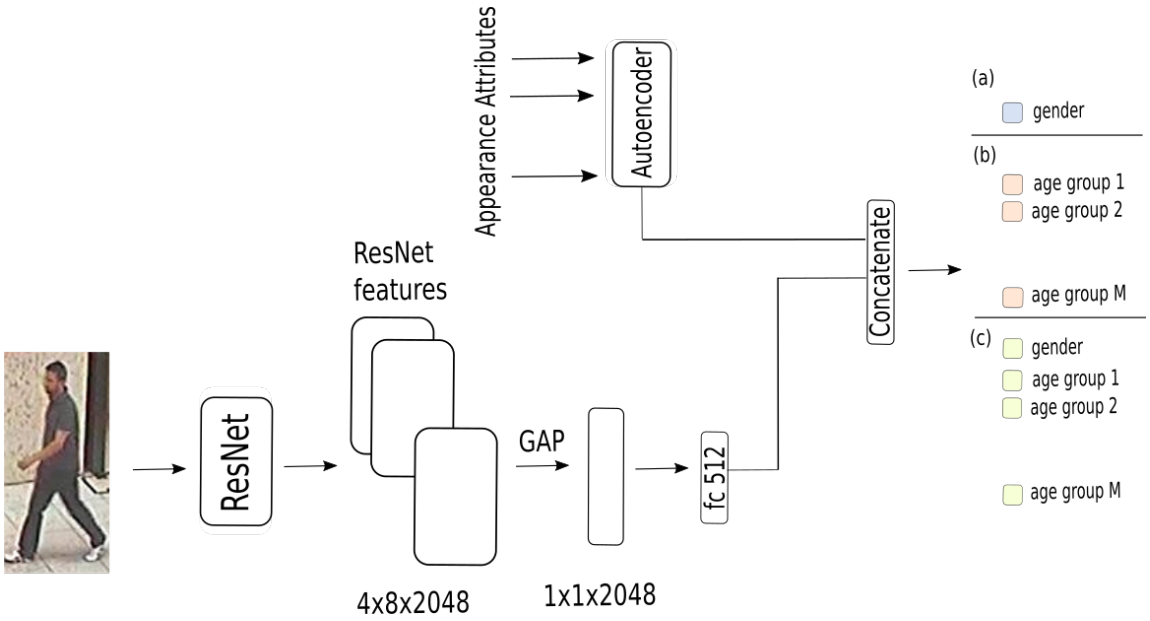


Figure 4.3: The ResNet+AE model for (a) gender classification, (b) age classification and (c) multi-label classification.

difference that the top classifier now predicts one of M possible classes. The M output neurons use the softmax activation function, so as to model a probability distribution consisting of M probabilities. The age range to be predicted by the model is the age range that belongs to the neuron with the greatest probability. The model for this case is depicted in Fig 4.2(b).

For the problem of age classification, we adopt the categorical cross-entropy loss, formulated as:

$$L_{cce} = -\sum_{i=1}^{M} y_i \log \hat{y}_i \,. \tag{4.11}$$

where M is the number of classes, and y_i, $\hat{y}_i$ are the one-hot encoded ground truth and predicted labels for the i-th class. Since the groundtruth labels are one-hot encoded only the positive class keeps its term in the loss, discarding the elements of the summation which are zero due to zero target labels. In addition to the categorical cross-entropy loss, we also explore the performance of the categorical focal loss and their weighted variants, which can be extended to the multi-class case easily:

$$L_{cfl} = -\sum_{i=1}^{M} y_i \,(1-\hat{y}_i)^{\gamma} \log \hat{y}_i \,, \tag{4.12}$$

$$L_{wcce} = -\sum_{i=1}^{M} w_i \, y_i \, \log \hat{y} \,, \tag{4.13}$$

$$L_{wcfl} = \sum_{i=1}^{M} -w_i \, y \, (1-\hat{y})^{\gamma} \, \log \hat{y} \, , \tag{4.14}$$

$$w_i = \frac{n_{\text{argmax}_{i \in \{1,\cdots,M\}} n_i}}{n_i} \, , \tag{4.15}$$

where the weighting factor w_i in Eq. (4.13), (4.19) is the weight loss assigned to the age group i, n_i is the number of examples of the i-th age group in the training set and $n_{\text{argmax}_{i \in \{1,\cdots,M\}} n_i}$ is the number of examples of the most representative class.

Similarly with the task of gender recognition stated above, we conduct an experiment with the combined model for the problem of age recognition, which is depicted in 4.3(b). The overall loss is the summation of the loss originating from the autoencoder and the loss originating from the ResNet and it is in the form of Eq. (4.10), where L_{ae} is one of the Eq. (4.8), (4.9) and L_{ResNet} is one of the Eq. (4.11), (4.12), (4.13), (4.19), whichever performs the best in the age classification case.

4.1.4 Multi-label classification

At the third and final experiment, we consider the multi-label recognition problem, in which both attributes, gender and age, should be predicted simultaneously. The multi-label recognition problem is more challenging than the gender and age recognition when treated separately. Now, each pedestrian image is labeled with a $(M+1)$-length vector, with the first element referring to the pedestrian's gender and the remaining M referring to the pedestrian's age range. For example, the target vector [0,0,0,0,0,1] indicates a male who belongs to the fifth age group. Fig. 4.2(c) depicts the model for this case.

For the multi-label recognition problem we use the sigmoid activation function for the $M+1$ output neurons and conduct experiments with the four loss functions, which for the multi-label case are reformulated as:

$$L_{bce} = -\sum_{i=1}^{M+1} y_i \log \hat{y}_i + (1-y_i) \log(1-\hat{y}_i) \, , \tag{4.16}$$

$$L_{bfl} = -\sum_{i=1}^{M+1} y_i \, (1-\hat{y}_i)^{\gamma} \log \hat{y}_i + (1-y_i) \, \hat{y}_i^{\gamma} \, \log(1-\hat{y}_i) \, , \tag{4.17}$$

$$L_{wbce} = -\sum_{i=1}^{M+1} w_i \left[y_i \, \log \hat{y}_i + (1-y_i) \, \log(1-\hat{y}_i) \right] , \tag{4.18}$$

$$L_{wbfl} = -\sum_{i=1}^{M+1} w_i \left[y_i \left(1-\hat{y}_i\right)^{\gamma} \log \hat{y}_i + \left(1-y_i\right) \hat{y}_i{}^{\gamma} \log(1-\hat{y}_i) \right], \tag{4.19}$$

$$w_i = \begin{cases} e^{p_i} & \text{if} \quad y = 0 \\ e^{1-p_i} & \text{if} \quad y = 1 \end{cases}, \tag{4.20}$$

where y_i, $\hat{y}_i$ are the ground truth and predicted labels for the i-th attribute, w_i is the loss weight assigned to attribute i and p_i is the proportion of the positive labels for the attribute i in the training set. Similar to the previous cases, we conduct an experiment with the combined model as well, which for the multi-label case is depicted in Fig. 4.3(c).

4.2 Evaluation details

4.2.1 Experimental setup

All of our experiments were conducted in a platform equipped with an AMD Ryzen 5 1600 3.6GHz processor (6 cores, 12 threads), with 32GBs of DDR4 RAM at 2400MHz and 2 Nvidia Titan Xp GPUs (with 3840 CUDA cores and 12GBs GDDR5X memory, each).

4.2.2 Datasets

PEdesTrian Attribute (PETA) [47] dataset merges 10 publicly small-scale pedestrian datasets and consists of 19000 images, each annotated with 61 binary attributes. The binary attributes cover an exhaustive set of characteristics of interest, including demographics (gender and age range), appearance (e.g. hair style), upper and lower body clothing style (casual or formal) and accessories (backpack, muffler etc). The list of attributes we are using for the PETA dataset is shown in Tab. 4.1. Image resolutions are ranging from 17x39 to 169x365 pixels. Sample images are depicted in Fig. 4.4. PETA dataset is randomly partitioned into three parts, of which 9500 for training, 1900 for validation and 7600 for testing. Images are all captured from far view field and they exhibit large differences in terms of lighting conditions, camera viewing

angles, image resolutions, background complexity and indoor/outdoor environment. Another thing to notice is that images in PETA are annotated based on person ID, i.e. the images belonging to the same person are annotated with the same attribute set, no matter if the attributes are visible in that image or not. Moreover, most images show the whole pedestrian's body, whereas in many real-world scenarios people are often partially visible due to occlusion with neighboring objects or other people.

RAP v2 (Richly Annotated Pedestrian) [48] dataset is collected from a realistic high-definition surveillance network at an indoor shopping mall. It has in total 84928 images and image resolutions are ranging from 36x92 to 344x554. Sample images are depicted in Fig. 4.5. Each image is annotated with 69 binary attributes; we chose only the appearance-related subset, which can be seen in Tab. 4.1. RAP v2 dataset provides 5 random partitions, though for the experiments we used only the first. Also, the images are independently annotated, which means that the images belonging to the same identity may have different attribute annotations due to the viewing angle variations, occlusions etc. This is particularly important in indoor surveillance scenes where a huge number of pedestrian images have some occlusions in various degrees. However, RAP v2 has less scenario heterogeneity than PETA, since PETA is derived from a mixture of different surveillance scenes with different qualities and environments.

PA-100K (Pedestrian Attribute) [40] dataset consists of 100000 pedestrian images with resolutions ranging from 50x100 to 758x454. Sample images are depicted in Fig. 4.6. Each image in PA100k is annotated with 26 attributes and the list of appearance-related attributes we have opted for that dataset is shown in Tab. 4.1. Images are randomly split into three partitions with a ratio of 8:1:1. PA-100K dataset was conducted by images captured from real outdoor surveillance cameras, which makes it more challenging than PETA and PAR v2 both in terms of its size and its complexity.

4.2.3 Evaluation metrics

Gender recognition For the problem of gender recognition, we use five metrics, namely accuracy, precision, recall, F1 score and mean accuracy. Accuracy quantifies the fraction of predictions the model got right, which is defined by:

$$Acc = \frac{TP + TN}{N_s}, \tag{4.21}$$

Table 4.1: Appearance-based attributes for each dataset.

Dataset	Attributes
PETA	accessoryHeadphone, carryingBabyBuggy, carryingBackpack, hairBald, footwearBoots, lowerBodyCapri, carryingOther, carryingShoppingTrolley, carryingUmbrella, lowerBodyCasual, upperBodyCasual, carryingFolder, lowerBodyFormal, upperBodyFormal, accessoryHairBand, accssoryHat, lowerBodyHotPants, upperBodyJacket, lowerBodyJeans, accessoryKerchief, footwearLeatherShoes, upperBodyLogo, hairLong, lowerBodyLongSkirt, upperBodyLongSleeve, lowerBodyPlaid, lowerBodyThinStripes, carryingLuggageCase, carryingMessengerBag, accessoryMuffler, accessoryNothing, carryingNothing, upperBodyNoSleeve, upperBodyPlaid, carryingPlasticBags, footwearSandals, footwearShoes, hairShort, lowerBodyShots, upperBodyShortSleeve, lowerBodyShortSkirt, footwearSneakers, footwearStocking, upperBodyThinStripes, upperBodySuit, carryingSuitcase, lowerBodySuits, accessorySunglasses, upperBodySweater, upperBodyThickStripes, lowerBodyTrousers, upperBodyTshirt, upperBodyOther, upperBodyVNeck
RAP v2	bodyFatter, bodyFat, bodyNormal, bodyThin, BodyThiner, baldHead, longHair, hat, glasses, sunglasses, muffler, mask, shirt, sweater, vest, tshirt, cotton, jacket, suitUp, upperBodyTight, shortSleeve, upperBodyOther, longTrousers, shorts, skirt, shortSkirt, longSkirt, dress, jeans, tightTrousers, leatherShoes, sportsShoes, boots, clothShoes, sandals, casualShoes, OtherShoes, backpack, shoulderBag, handBag, waistBag, box, plasticBag, paperBag, handTrunk, carryingBaby, carryingOther
PA100k	hat, glasses, handbag, shoulderBag, backpack, holdObjectsInFront, ShortSleeve, LongSleeve, UpperStride, UpperLogo, UpperPlaid, UpperSplice, LowerStripe, LowerPattern, LongCoat, Trousers, Shorts, Skirt-Dress, boots

Figure 4.4: Sample images from the PETA dataset.

Figure 4.5: Sample images from the RAP v2 dataset.

Figure 4.6: Sample images from the PA100k dataset.

where TP is the number of true positives, that is the number of positive samples (females) that the model predicts correctly as positives, TN is the number of true negatives, that is the number of negative samples (males) that the model predicts correctly as negatives and N_s is the number of samples. In the case of gender recog-

nition, accuracy is the number of correctly predicted males and females over all samples. Precision, recall and F1 score are defined as:

$$precision = \frac{TP}{TP + FP}\,, \tag{4.22}$$

$$recall = \frac{TP}{TP + FN}\,, \tag{4.23}$$

$$f1 = \frac{2 \cdot precision \cdot recall}{precision + recall}\,, \tag{4.24}$$

where TP is the number of true positives as before, FP is the false positives, that is the number of samples that the model falsely predicted as positives (females) and FN is the false negatives, that is the number of samples that the model falsely predicted as negatives (males). Intuitively, precision is the fraction of true positives among the predicted positives and recall is the fraction of the total amount of true positives which actually the model predicts; F1 score is the harmonic mean of the precision and recall. Finally, we use the mean accuracy (mAcc) metric, which is calculated by:

$$mAcc = \frac{1}{2}\Big(\frac{TP}{P} + \frac{TN}{N}\Big)\,, \tag{4.25}$$

which is the mean of the ratio of correctly predicted positives and the ratio of correctly predicted negatives.

Age classification For the problem of age classification, we similarly use the accuracy, precision, recall and F1 score, slightly modified, since age classification is a multi-class problem. In this case, accuracy quantifies how often predictions match the true labels by checking to see if the index of the maximal true label is equal to the index of the maximal predicted label:

$$Acc = \frac{\sum_{i=1}^{N_s} \mathbb{1}\big(\text{argmax}(y_i) = \text{argmax}(\hat{y}_i)\big)}{N_s}\,, \tag{4.26}$$

where $\mathbb{1}(\cdot)$ is the indicator function, which equals one if its argument is true and equals zero otherwise, y_i, $\hat{y}_i$ is the one-hot encoded true and predicted labels respectively and N_s is the total number of samples. Precision, recall and F1 score are now calculated for each age group separately and an average over all M age groups is taken to provide the mean precision (mPrecision), mean recall (mRecall) and mean F1 score (mF1):

$$mPrecision = \frac{1}{M}\sum_{i=1}^{M}\frac{TP_i}{TP_i + FP_i}\,, \tag{4.27}$$

$$mRrecall = \frac{1}{M}\sum_{i=1}^{M}\frac{TP_i}{TP_i + FN_i}\,, \tag{4.28}$$

$$mF1 = \frac{2 \cdot mPrecision \cdot mRecall}{mPrecision + mRecall}\,, \tag{4.29}$$

where M is the number of age groups, TP_i, FP_i and TN_i is the true positives, false positives and true negatives for the i-th age group, respectively.

Multi-label recognition For the problem of multi-label recognition, accuracy, precision, recall and F1 score are calculated per-sample and are defined as:

$$Acc = \frac{1}{N_s}\sum_{i=1}^{N_s}\frac{TP_i + TN_i}{N_s}\,, \tag{4.30}$$

$$precision = \frac{1}{N_s}\sum_{i=1}^{N_s}\frac{TP_i}{TP_i + FP_i}\,, \tag{4.31}$$

$$recall = \frac{1}{N_s}\sum_{i=1}^{N_s}\frac{TP_i}{TP_i + FN_i}\,, \tag{4.32}$$

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall}\,, \tag{4.33}$$

where N_s is the number of samples, TP_i, FP_i and FN_i is the true positives, false positives and false negatives of the i-th sample, respectively. Finally, mean accuracy is calculated per-label via the formula:

$$mA = \frac{1}{2(M+1)}\sum_{i=1}^{M+1}\left(\frac{TP_i}{P_i} + \frac{TN_i}{N_i}\right), \tag{4.34}$$

where TP_i, FP_i and FN_i is the true positives, false positives and false negatives calculated independently for the i-th label.

4.3 Experimental results

Before presenting the evaluation results, we should give some information regarding the settings we used throughout the experiments.

It should be noted that we use the pre-trained ResNet50 architecture, which has already been trained on the ImageNet dataset. We followed this tactic as the ResNet50 is a relatively deep network and training it from scratch would require long training times and would probably lead to overfitting issues.

All images were pre-processed and resized to 256×128 since pedestrians walking are usually rectangular. Also, in order to avoid overfitting we employed some of the commonly used data augmentation techniques. These include horizontal flips, random width and height shifts and random crops. All the data augmentations were performed on-line, i.e. each image was randomly augmented by the one or combination of the listed methods in each iteration separately.

As for the optimizer, we used the mini-batch stohastic gradient descent with momentum set to 0.9. The batch size is 50 samples per iteration. Also, in all our experiments we used a learning rate reduction scheme, where the learning rate was multiplied by 0.5 when the validation error was on a plateau and was not improving for two consecutive epochs. Finally, when the validation error was not improving for five consecutive epochs, we used early stopping to terminate the training process.

Finally, preliminary experiments on the ResNet+AE model showed that adding a dropout layer with a small dropout probability (e.g 0.1) can act as a regularizer, hence a dropout layer is added right after the feature concatenation layer.

4.3.1 PETA dataset

Gender classification

Table 4.2 compares the performance of the four loss functions described in Sec. 4.1.2 for the task of gender classification. Although in the PETA dataset, the gender distribution is nearly balanced (see Fig. 4.7), it can be seen that both weighted loss functions outperform their un-weighted counterparts. Specifically, WBCE performs 0.44% better in terms of the F1 score and 0.36% better in terms of the mAcc metric compared to BCE. Similarly, WBFL is by 1.8% better in terms of the F1 score and by 1.41% better in terms of the mAcc compared to BFL. Comparing the weighted loss functions, WBCE outperforms WBFL by 3.48% in the F1 score and by 3.16% in mAcc and subsequently it is used to train the ResNet+AE model.

The proposed ResNet+AE model leverages the appearance-based attributes in the gender classification scheme, achieving 90.71% and 91.53% in the F1 score and mAcc, respectively, outperforming the single-ResNet architecture with any of the loss functions.

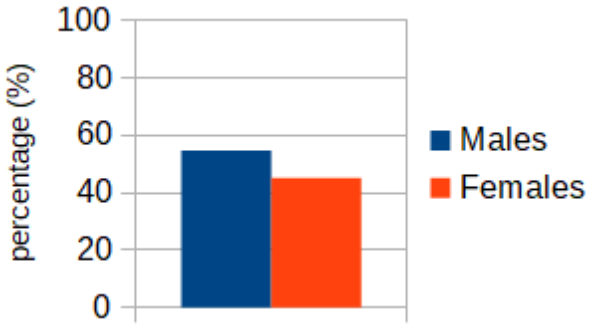


Figure 4.7: The distribution of the gender attribute in the PETA dataset.

Table 4.2: Gender: Performance comparison of the four loss functions and the ResNet+AE model on the PETA dataset (in %).

	Prec	Rec	F1	mAcc	Acc
ResNet-BCE	88.80	86.07	87.42	88.57	88.81
ResNet-WBCE	**87.86**	**87.89**	**87.86**	**88.93**	**89.03**
ResNet-BFL	85.51	79.87	82.58	84.36	84.79
ResNet-WBFL	84.57	84.18	84.38	85.77	85.92
ResNet+AE	**91.67**	**89.79**	**90.71**	**91.53**	**91.70**

Age classification

The age category distribution in the PETA dataset can be seen in Fig. 4.8. There are five age classes to be predicted, <16, 16-30, 31-45, 46-60 and >60, with distributions of 0.9%, 49.77%, 32.92%, 10.24%, 6.17% respectively. Hence, it is apparent that the age attribute suffers from a severe class imbalance.

Table 4.3 compares the performance of the four loss functions described in Sec. 4.1.3. More details can be found on Appendix A on Fig. A.1, which reports the metrics for each age group separately. Although the weighted loss functions balance each example according to the class it belongs, giving more focus on the under-represented classes, they do not seem to improve none of the metrics. We consider that this behavior is likely caused by poor features, since it is difficult for the ResNet to provide representative features given that there is no near-face information and sometimes the pedestrian is standing backwards. In addition, since the optimization

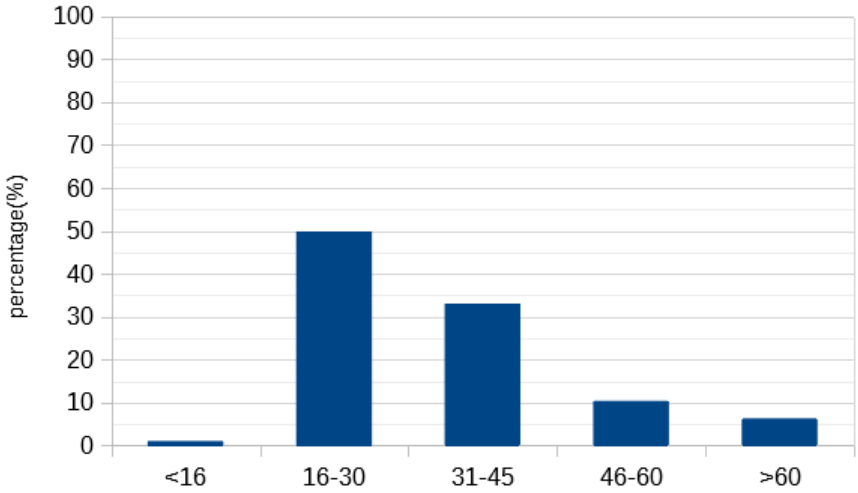


Figure 4.8: The distribution of the age categories in the PETA dataset.

Table 4.3: Age: Performance comparison of the four loss functions and the ResNet+AE model on the PETA dataset (in %).

	mPrec	mRec	mF1	mAcc	Acc
ResNet-CCE	85.55	66.76	73.19	–	77.29
ResNet-WCCE	67.37	70.53	68.72	–	70.76
ResNet-CFL	84.01	**68.03**	**73.85**	–	**76.89**
ResNet-WCFL	54.23	64.64	57.80	–	64.04
ResNet+AE	**80.06**	**72.75**	**75.84**	–	**79.61**

method is performed in batches, it is not guaranteed that there are examples for each age group in each batch, hence the model is overwhelmed by the majority class and cannot ensure good discriminations among the five age categories. The categorical focal loss performs slightly better than the categorical cross entropy by 0.66% in terms of the mF1 score and subsequently it is used to train the ResNet+AE model.

The proposed ResNet+AE model outperforms the single ResNet architecture, achieving 75.84% in terms of the mF1 score. This means that there is some sort of interdependence among the appearance-based attributes, which helps the ResNet+AE model to yield a better age classification performance.

Multi-label classification

Table 4.4 summarizes the performance of the four different loss functions described in Sec. 4.1.4 for the task of multi-label classification, where the model classifies both the gender and the age attributes. Since the gender attribute is nearly balanced (see Fig. 4.7), the heavy imbalance of the age attribute (see Fig. 4.8) overwhelms the distribution to be modeled. However, the performance is not degraded despite the fact that the model now has to predict both attributes simultaneously. The weighted loss functions manage to achieve better results compared to their un-weighted counterparts. More specifically, WBCE is 0.87% and 1.23% better in F1 score and mAcc respectively compared to the plain BCE. Similarly, WBFL performs better by 1.66% in F1 score and by 1.14% in mAcc compared to plain BFL. The best among the four loss functions is the WBCE achieving 79.4% and 82.82% in F1 score and mAcc respectively and this loss function is used to consequently train the ResNet+AE model.

Table 4.4: Multi-label: Performance comparison of the four loss functions and the ResNet+AE model on the PETA dataset (in %).

	Prec	Rec	F1	mAcc	Acc
ResNet-BCE	79.20	77.88	78.53	81.59	91.40
ResNet-WBCE	**79.22**	**79.58**	**79.40**	**82.82**	**91.09**
ResNet-BFL	76.90	75.95	76.42	80.64	90.47
ResNet-WBFL	77.64	78.52	78.08	81.78	90.37
ResNet+AE	**80.02**	**80.80**	**80.41**	**84.49**	**91.54**

The proposed ResNet+AE model outperforms the single-ResNet architecture, achieving 80.41% in F1 score and 84.49% in mAcc.

4.3.2 RAP v2 dataset

Gender classification

The gender distribution in the RAP v2 dataset is quite imbalanced given that the number of the males are over twice the number of the females (see Fig. 4.9).

Table 4.5 compares the performance of the four loss functions described in Sec. 4.1.2.

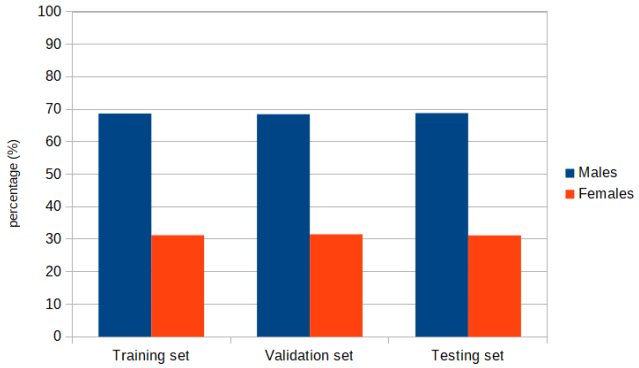


Figure 4.9: The distribution of the gender attribute in the RAP v2 dataset.

WBCE performs 0.42% better in terms of the F1 score and 0.95% better in terms of the mAcc metric compared to BCE. BFL is by 0.6% better in terms of the F1 score compared to WBFL but WBFL is 0.25% better in terms of the mAcc compared to BFL. Nevertheless, WBCE outperforms WBFL by 2.87% in the F1 score and 2.47% in mAcc and this is the loss function of choice for the ResNet+AE model.

The proposed ResNet+AE model performs comparably well achieving 91.72% and 94.12% in F1 score and mAcc respectively but does not outperform the single-ResNet architecture with the WBCE loss function.

Table 4.5: Gender: Performance comparison of the four loss functions and the ResNet+AE model on the RAP v2 dataset (in %).

	Prec	Rec	F1	mAcc	Acc
ResNet-BCE	93.18	91.81	92.49	94.38	95.35
ResNet-WBCE	**91.00**	**94.91**	**92.91**	**95.33**	**95.49**
ResNet-BFL	92.90	91.82	92.36	94.32	95.26
ResNet-WBFL	89.46	94.17	91.76	94.57	94.73
ResNet+AE	91.16	92.30	91.72	94.12	94.81

Age classification

The age category distribution in the RAP v2 dataset can be seen in Fig. 4.10. There are five age classes to be predicted with distributions of 0.92%, 40.44%, 54.89%, 3.53% and 0.22% respectively (in the training set). The distribution is heavily unbalanced with the second and third age categories to be more represented compared to the rest.

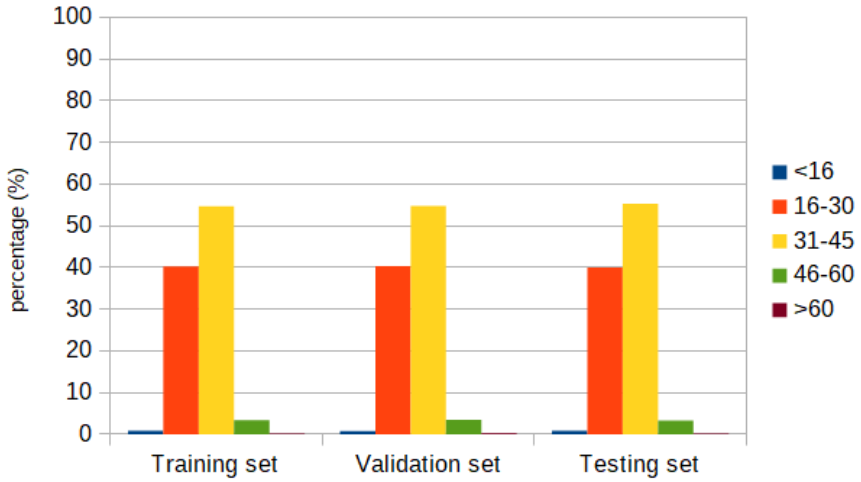


Figure 4.10: The distribution of the age categories in the RAP v2 dataset.

Table 4.6 compares the performance of the four loss functions described in Sec. 4.1.3. More details can be found on Appendix A on Fig. A.2, which reports the metrics for each age group separately. Similar with the PETA dataset, the weighted loss functions do not improve the performance compared to their un-weighted counterparts. CFL is the best performing loss function, which outperforms the CCE by 8.36% in the mF1 score, and it is used to consequently train the ResNet+AE model.

The proposed ResNet+AE model demonstrates inferior performance, achieving 36.27% in the mF1 score, which indicates that the performance is degraded. Therefore, the combined model cannot leverage the appearance-based attributes for the age classification, and the single-ResNet architecture with the CFL is the best performing model.

Multi-label classification

Table 4.7 summarizes the performance of the four different loss functions when the model classifies both the gender and the age attributes. It can be seen that the weighted loss functions perform slightly better than the unweighted counterparts.

Table 4.6: Age: Performance comparison of the four loss functions and the ResNet+AE model on the RAP v2 dataset (in %).

	mPrec	mRec	mF1	mAcc	Acc
ResNet-CCE	41.73	29.97	31.45	–	65.71
ResNet-WCCE	26.27	49.92	24.51	–	39.36
ResNet-CFL	**48.46**	**36.82**	**39.81**	–	**64.82**
ResNet-WCFL	26.00	49.09	23.30	–	37.31
ResNet+AE	41.57	34.30	36.27	–	64.94

Specifically, WBCE performs 0.26% better in terms of the F1 score and 4.3% better in terms of the mAcc compared to BCE, and WBFL performs 0.18% better in terms of the F1 score and 2.09% better in terms of the mAcc compared to BFL. Overall, in the single ResNet architecture, WBCE performs 1.51% better in terms of the F1 score, but WBFL performs 0.6% better in terms of the mAcc. We chose the WBFL as the best performing loss function, as mAcc is a label-based metric and is more important metric in the multi-label classification case.

Table 4.7: Multi-label: Performance comparison of the four loss functions and the ResNet+AE model on the RAP v2 dataset (in %).

	Prec	Rec	F1	mAcc	Acc
ResNet-BCE	71.08	70.63	70.85	63.56	88.40
ResNet-WBCE	70.71	71.52	71.11	67.86	88.25
ResNet-BFL	69.31	69.53	69.42	66.37	87.97
ResNet-WBFL	**68.82**	**70.40**	**69.60**	**68.46**	**87.73**
ResNet+AE	67.63	68.30	67.96	67.76	87.29

Concerning the combined ResNet+AE model, although it is quite similar in performance compared to the most of the single-ResNet architectures, it does not outperform the single-ResNet case with the WBFL loss function.

4.3.3 PA100k dataset

Gender classification

Table 4.8 compares the performance of the four loss functions described in Sec. 4.1.2. Similar with the PETA dataset, the gender distribution is nearly balanced in the PA100k dataset, as well (see Fig. 4.11). Both weighted loss functions outperform

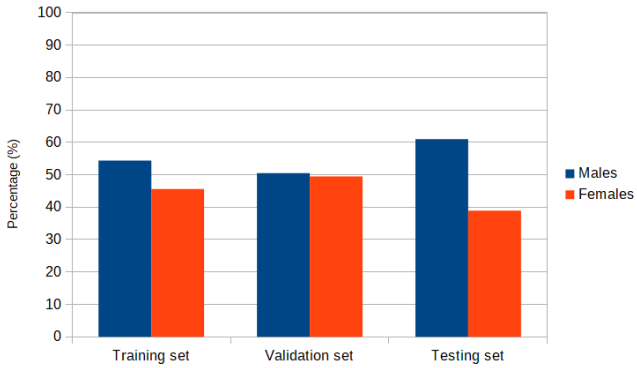


Figure 4.11: The distribution of the gender attribute in the PA100k dataset.

their un-weighted counterparts. Specifically, WBCE performs 0.26% better in terms of the F1 score and 0.30% better in terms of the mAcc metric compared to BCE. Similarly, WBFL is by 1.82% better in terms of the F1 score and by 1.52% better in terms of the mAcc compared to BFL. Comparing the weighted loss functions, WBCE outperforms WBFL by 2.87% in the F1 score and 2.47% in mAcc, therefore it is used to consequently train the ResNet+AE model.

Table 4.8: Gender: Performance comparison of the four loss functions and the ResNet+AE model on the PA100k dataset (in %).

	Prec	Rec	F1	mAcc	Acc
ResNet-BCE	82.48	84.45	83.45	86.50	86.95
ResNet-WBCE	**81.32**	**86.24**	**83.71**	**86.80**	**86.92**
ResNet-BFL	79.49	78.57	79.02	82.81	83.75
ResNet-WBFL	79.99	81.70	80.84	84.33	84.91
ResNet+AE	60.42	66.61	63.36	69.38	69.99

The proposed ResNet+AE model has an inferior performance of 63.36% and

69.38% in F1 score and mAcc respectively. We believe that this happens due to the more generic appearance-based attributes in the PA100k dataset. The model cannot find any important relationships among the attributes to provide better results and its performance is significantly degraded.

Age classification

The age category distribution in the PA100k dataset can be seen in Fig. 4.12. There are three age classes to be predicted with distributions of 5.19%, 93.4% and 1.41% respectively (in the training set). The distribution is heavily unbalanced with the middle age category to be overly represented compared to the rest.

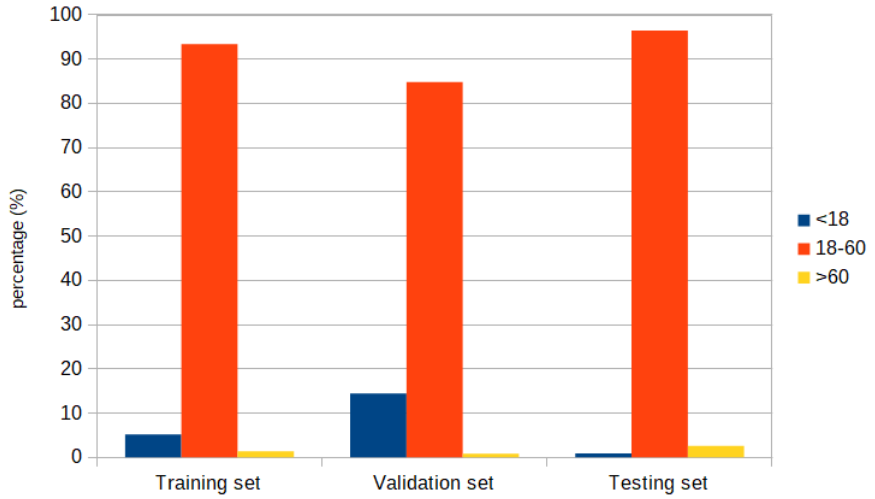


Figure 4.12: The distribution of the age categories in the PA100k dataset.

Table 4.9 compares the performance of the four loss functions described in Sec. 4.1.3. More details can be found on Appendix A on Fig. A.3, which reports the metrics for each age group separately. Here, the weighted loss functions perform better compared to their un-weighted counterparts. Specifically, WCCE is better by 12.4% compared to the CCE and WCFL is better by 1.78% compared to CFL in terms of the mF1 score, hence we selected the WCCE to consequently train the ResNet+AE model.

The proposed ResNet+AE model demonstrates inferior performance, achieving only 26.57% in the mF1 score, which indicates that the appearance-based attributes are relatively vague and cannot be leveraged for the age classification.

Table 4.9: Age: Performance comparison of the four loss functions and the ResNet+AE model on the PA100k dataset (in %).

	mPrec	mRec	mF1	mAcc	Acc
ResNet-CCE	33.06	33.59	33.21	–	96.2
ResNet-WCCE	**42.75**	**63.73**	**45.61**	–	**85.11**
ResNet-CFL	33.63	33.63	33.29	–	96.32
ResNet-WCFL	36.46	51.30	35.07	–	75.15
ResNet+AE	33.99	35.65	26.57	–	54.18

Multi-label classification

Table 4.10 summarizes the performance of the four different loss functions when the model classifies both the gender and age attributes. It can be seen that precision, recall and F1 score are quite high, since they are instance-based metrics. The mAcc metric is severely affected by the large class imbalance of the age attribute (see Fig. 4.12), which overwhelms the overall performance. BCE and WBCE are quite close in performance, but plain BCE is slightly better by 0.92% in terms of the F1 score and 0.38% in terms of the mAcc. Similarly, BFL and WBFL are quite close as well, with the BFL to be better by 0.69% in terms of the F1 score and WBFL to be better by 0.15% in terms of the mAcc. Overall, BCE is the best performing loss function and it used to consequently train the ResNet+AE model.

The proposed ResNet+AE model achieves 88.48% in F1 score and 53.61% in mAcc, but does not outperform the single ResNet case with the BCE loss function.

Table 4.10: Multi-label: Performance comparison of the four loss functions and the ResNet+AE model on the PA100k dataset (in %).

	Prec	Rec	F1	mAcc	Acc
ResNet-BCE	**92.90**	**93.06**	**92.98**	**58.79**	**94.30**
ResNet-WBCE	92.11	92.02	92.06	58.41	93.67
ResNet-BFL	92.51	90.14	91.31	56.36	92.54
ResNet-WBFL	91.18	90.06	90.62	56.51	91.95
ResNet+AE	89.36	87.63	88.48	53.61	89.53

4.4 Ablation Studies

In order to get an intuition about whether the appearance-based attributes provide useful information, we trained the autoencoder of the ResNet+AE model on its own for several epochs and used its features to perform the gender and age classification. For the autoencoder's loss function, we tested both the binary cross entropy and the focal loss to choose the one that performs the best. In most cases, the autoencoder with the focal loss was trained faster than that with the cross entropy and provided a better reconstruction. Hence, the autoencoder part of the ResNet+AE model is trained exclusively with the focal loss to comply with one loss function across all cases. The results are shown in Tables 4.11, 4.12 and 4.13 for the gender classification on the PETA, RAP v2 and PA100k datasets respectively. Similarly, for the age classification, the results are shown in Tables 4.14, 4.15 and 4.16.

Table 4.11: Evaluating the autoencoder for the gender classification on the PETA dataset (in %).

	Prec	Rec	F1	mAcc	Acc
AE-BCE+gender	90.18	86.64	88.37	89.43	89.70
AE-BFL+gender	90.67	85.68	88.10	89.21	89.55

Moreover, in an attempt to investigate how the choice of the appearance-based

Table 4.12: Evaluating the autoencoder for the gender classification on the RAP v2 dataset (in %).

	Prec	Rec	F1	mAcc	Acc
AE-BCE+gender	97.19	79.18	87.27	89.07	92.80
AE-BFL+gender	95.93	80.73	87.68	89.59	92.93

Table 4.13: Evaluating the autoencoder for the gender classification on the PA100k dataset (in %).

	Prec	Rec	F1	mAcc	Acc
AE-BCE+gender	67.88	60.16	63.79	71.00	73.39
AE-BFL+gender	70.08	54.70	61.44	69.89	73.25

Table 4.14: Evaluating the autoencoder for the age classification on the PETA dataset (in %).

	mPrec	mRec	mF1	mAcc	Acc
AE-BCE+age	60.47	47.23	49.98	–	66.10
AE-BFL+age	59.87	50.28	52.87	–	66.47

Table 4.15: Evaluating the autoencoder for the age classification on the RAP v2 dataset (in %).

	mPrec	mRec	mF1	mAcc	Acc
AE-BCE+age	45.35	25.86	25.40	–	63.97
AE-BFL+age	32.81	26.44	26.24	–	64.01

Table 4.16: Evaluating the autoencoder for the age classification on the PA100k dataset (in %).

	mPrec	mRec	mF1	mAcc	Acc
AE-BCE+age	55.47	49.27.23	51.87	–	96.03
AE-BFL+age	32.16	33.01	32.57	–	95.55

attributes affect the performance of the ResNet+AE model, we selected only a subset of the available attributes, which we intuitively considered as the most useful for the task of gender and age classification. This subset is the underlined attribute set seen in Table 4.1. This model has the same architecture as the ResNet+AE model with the difference that the autoencoder's input is smaller in size, and it is called ResNet+AE(less attributes). We performed this experiment on the RAP v2 and the PA100k datasets, since the full set of attributes on these datasets did not improve the performance compared to the single-ResNet architecture.

For the problem of gender classification, the results are shown in Tables 4.17 and 4.18 for the RAP v2 and PA100k datasets repsectively. On the RAP v2 dataset, the ResNet+AE(less attributes) model has a comparable performance to its counterpart with the full set of attributes (ResNet+AE) of Table 4.5, achieving 91.66% in F1 score and 93.97% in mAcc, but again does not outperform the single-ResNet architecture with the WBCE loss function. On the PA100k dataset, the ResNet+AE(less attributes) model demonstrates a better performance compared to its counterpart with the full set of attributes of Table 4.8, achieving 76.63% in F1 score and 80.83% in mAcc, but does not outperform the single-ResNet architecture with the WBCE loss function.

Table 4.17: Gender: Performance of the ResNet+AE(less attributes) model on the RAP v2 dataset (in %).

	Prec	Rec	F1	mAcc	Acc
ResNet+AE (less attributes)	91.53	91.79	91.66	93.97	94.79

For the problem of age classification, the results are shown on Tables 4.19 and 4.20

Table 4.18: Gender: Performance of the ResNet+AE(less attributes) model on the PA100k dataset (in %).

	Prec	Rec	F1	mAcc	Acc
ResNet+AE (less attributes)	74.40	79.00	76.63	80.63	81.23

for the RAP v2 and PA100k datasets respectively. On the RAP v2 dataset, the performance of the ResNet+AE(less attributes) model is even more degraded compared to the ResNet+AE model with the full set of attributes of Table 4.6, achieving 33.28% in mF1 score. On the PA100k dataset, the performance of the the ResNet+AE(less attributes) model is a little better compared to the ResNet+AE model, but the performance is overall not satisfying.

Table 4.19: Age: Performance of the ResNet+AE(less attributes) model on the RAP v2 dataset (in %).

	mPrec	mRec	mF1	mAcc	Acc
ResNet+AE (less attributes)	38.60	31.29	32.88	–	62.68

Table 4.20: Age: Performance of the ResNet+AE(less attributes) model on the PA100k dataset (in %).

	mPrec	mRec	mF1	mAcc	Acc
ResNet+AE (less attributes)	35.40	47.47	28.46	–	55.27

For the problem of multi-label classification, the results are shown in Tables 4.21 and 4.22 for the RAP v2 and PA100k datasets respectively. On both datasets, the ResNet+AE(less attributes) model is comparable to its counterpart with the full set of attributes, but does not outperform the sinlge-ResNet model.

Table 4.21: Multi-label: Performance of the ResNet+AE(less attributes) model on the RAP v2 dataset (in %).

	Prec	Rec	F1	mAcc	Acc
ResNet+AE (less attributes)	66.36	66.64	66.50	64.01	86.38

Table 4.22: Multi-label: Performance of the ResNet+AE(less attributes) model on the PA100k dataset (in %).

	Prec	Rec	F1	mAcc	Acc
ResNet+AE (less attributes)	88.17	86.94	87.55	52.39	88.43

4.5 Qualitative results

The following figures depict some indicative misclassified examples for each dataset. The predictions are obtained from the best performing models, which are in bold in the tables in Sec. 4.3.

For the gender classification, the failure cases indicate that on average false predictions are made when the image is of low illumination or blurry. Also, in most failure cases the human is captured from behind or from the side, hence there is less visual information for the model to provide an accurate inference. For the age classification, it can be seen that in most failure cases the model predicts adjacent to the true age categories, which implies that it is not far from predicting the true category. It is evident though that it is hard for the model to make accurate discriminations among the age categories from pedestrian images, and particularly when there is a large degree of class imbalance.

Figure 4.13: Females predicted as males from the PETA dataset.

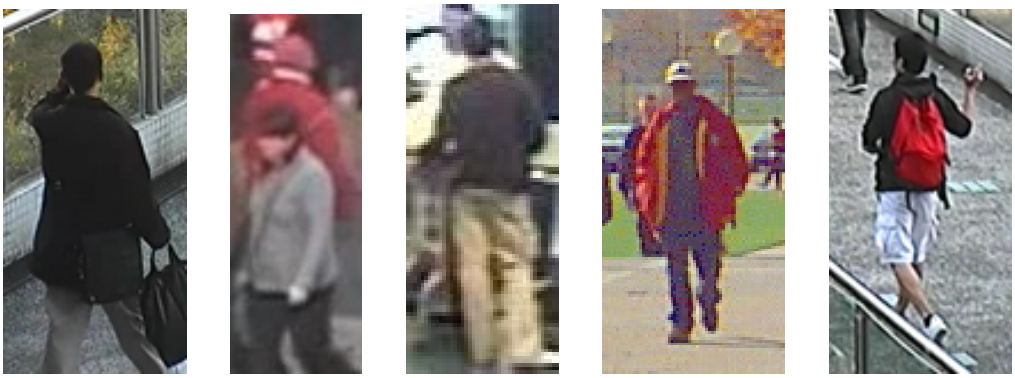


Figure 4.14: Males predicted as females from the PETA dataset.

Figure 4.15: Misclassified examples from the PETA dataset for the age classification problem. True (T) and predicted (P) classes are shown in green and red respectively at the bottom right corner of the image.

Figure 4.16: Females predicted as males from the RAP v2 dataset.

Figure 4.17: Males predicted as females from the RAP v2 dataset.

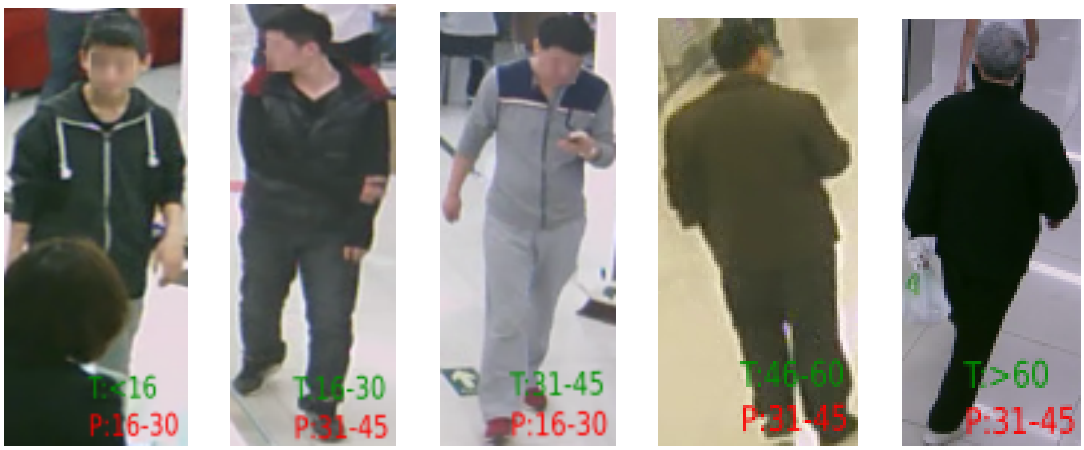


Figure 4.18: Misclassified examples from the RAP v2 dataset for the age classification problem. True (T) and predicted (P) classes are shown in green and red respectively at the bottom right corner of the image.

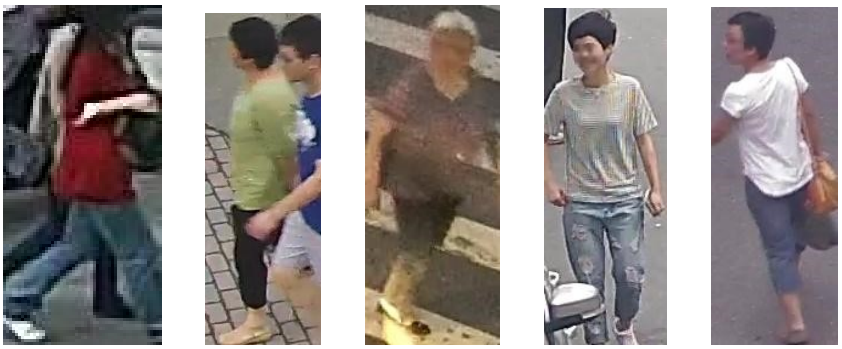


Figure 4.19: Females predicted as males from the PA100k dataset.

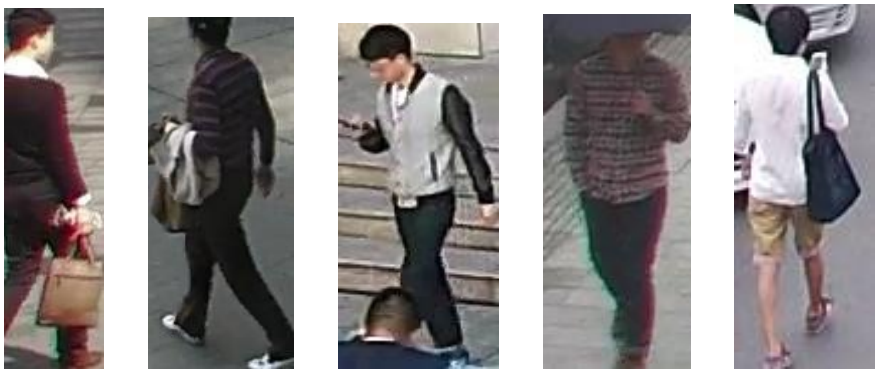


Figure 4.20: Males predicted as females from the PA100k dataset.

Figure 4.21: Misclassified examples from the PA100k dataset for the age classification problem. True (T) and predicted (P) classes are shown in green and red respectively at the bottom right corner of the image.

Chapter 5

Conclusion

In this thesis, we studied the problem of gender and age classification from pedestrian images. Pedestrian images are captured in-the-wild, they might contain only part of the human body, and often are taken at far distance. The far view field produces images with different pedestrian postures, different camera viewing angles and illuminations, which need to be addressed by the model implementation. In addition, the class imbalance which characterizes the datasets with pedestrian images makes the task even more challenging. Having said this, we focused on examining how different loss functions - including the cross-entropy, focal loss and their weighted counterparts - perform under the class imbalance problem. We used the ResNet architecture as the backbone of our experiments in order to study the performance of the loss functions. In addition, we tested another model, which concatenates the features from the ResNet and the features from an autoencoder, which is trained in parallel with appearance-based attributes.

Taken into consideration the experimental results, the gender classification is an easier task, as the ResNet can extract representative features to make an accurate classification. Age classification is a more challenging problem, since age categories are heavily imbalanced and with no near-face information, the ResNet cannot provide a proper discrimination across the categories. The multi-label classification is also a challenging task, as the age category imbalance overwhelms the distribution to be modeled. As for the combined model, reasonable classification accuracy can be obtained when the appearance-based attributes involve some sort of relationship.

However, this greatly depends on the appearance-based attributes, which need to provide useful patterns and be well represented in each dataset.

There are several research directions to be followed in future work. First, it would be useful to study the correlation among subsets of the appearance-based attributes and use only the subset that is more beneficial for each case. Another research direction is to experiment with deeper architectures for the autoencoder and use different fusion techniques for the ResNet and autoencoder's features.

Bibliography

[1] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool, "Real-time facial feature detection using conditional regression forests," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2578–2585, IEEE, 2012.

[2] S. Z. Gilani and A. Mian, "Perceptual differences between men and women: A 3d facial morphometric perspective," in *2014 22nd International Conference on Pattern Recognition (ICPR)*, pp. 2413–2418, IEEE, 2014.

[3] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 121–135, 2017.

[4] B. Moghaddam and M. hsuan Yang, "Learning gender with support faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 707–711, 2002.

[5] B. Li, X.-C. Lian, and B.-L. Lu, "Gender classification by combining clothing, hair and facial component classifiers," *Neurocomputing*, vol. 76, no. 1, pp. 18–27, 2012.

[6] C. Shan, "Learning local binary patterns for gender classification on real-world face images," *Pattern Recognition Letters*, vol. 33, no. 4, pp. 431–437, 2012.

[7] C.-B. Ng, Y. H. Tay, and B.-M. Goi, "A review of facial gender recognition," *Pattern Analysis and Applications*, vol. 18, 11 2015.

[8] A. Geetha, M. Sundaram, and B. Vijayakumari, "Gender classification from face images by mixing the classifier outcome of prime, distinct descriptors," *Soft Computing*, vol. 23, no. 8, pp. 2525–2535, 2019.

[9] J. Mansanet, A. Albiol, and R. Paredes, "Local deep neural networks for gender recognition," *Pattern Recognition Letters*, vol. 70, pp. 80–86, 2016.

[10] F. Juefei-Xu, E. Verma, P. Goel, A. Cherodian, and M. Savvides, "Deepgender: Occlusion and low resolution robust facial gender classification via progressively trained convolutional neural networks with attention," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*, pp. 68–77, 2016.

[11] Y. H. Kwon *et al.*, "Age classification from facial images," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 762–767, 1994.

[12] A. Lanitis, C. Taylor, and T. Cootes, "Toward automatic simulation of aging effects on face images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, pp. 442–455, 05 2002.

[13] X. Geng, Z.-H. Zhou, and K. Smith-Miles, "Automatic age estimation based on facial aging patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2234–2240, 2007.

[14] X. Geng, Z.-H. Zhou, Y. Zhang, G. Li, and H. Dai, "Learning from facial aging patterns for automatic age estimation," in *Proceedings of the 14th ACM international conference on Multimedia*, pp. 307–316, 2006.

[15] G. Guo, Y. Fu, C. R. Dyer, and T. S. Huang, "Image-based human age estimation by manifold learning and locally adjusted robust regression," *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1178–1188, 2008.

[16] A. Günay Yılmaz and V. Nabiyev, "Automatic age classification with lbp," pp. 1 – 4, 11 2008.

[17] G. Guo, G. Mu, Y. Fu, and T. S. Huang, "Human age estimation using bio-inspired features," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 112–119, IEEE, 2009.

[18] D. Yi, Z. Lei, and S. Z. Li, "Age estimation by multi-scale convolutional network," in *Asian Conference on Computer Vision*, pp. 144–158, Springer, 2014.

[19] X. Wang, R. Guo, and C. Kambhamettu, "Deeply-learned feature for age estimation," in *IEEE Winter Conference on Applications of Computer Vision*, pp. 534–541, IEEE, 2015.

[20] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua, "Ordinal regression with multiple output cnn for age estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4920–4928, 2016.

[21] Z. Hu, Y. Wen, J. Wang, M. Wang, R. Hong, and S. Yan, "Facial age estimation with age difference," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3087–3097, 2016.

[22] S. Chen, C. Zhang, and M. Dong, "Deep age estimation: From classification to ranking," *IEEE Transactions on Multimedia*, vol. 20, no. 8, pp. 2209–2222, 2017.

[23] M. Toews and T. Arbel, "Detection, localization, and sex classification of faces from arbitrary viewpoints and under occlusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, pp. 1567–1581, 2008.

[24] E. Eidinger, R. Enbar, and T. Hassner, "Age and gender estimation of unfiltered faces," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2170–2179, 2014.

[25] G. Levi and T. Hassner, "Age and gender classification using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*, pp. 34–42, 2015.

[26] P. Rodríguez, G. Cucurull, J. M. Gonfaus, F. X. Roca, and J. González, "Age and gender recognition in the wild with deep attention," *Pattern Recognition*, vol. 72, pp. 563–571, 2017.

[27] M. Duan, K. Li, C. Yang, and K. Li, "A hybrid deep learning cnn–elm for age and gender classification," *Neurocomputing*, vol. 275, pp. 448–461, 2018.

[28] R. Layne, T. M. Hospedales, S. Gong, and Q. Mary, "Person re-identification by attributes.," in *British Machine Vision Conference (BMVC)*, vol. 2, p. 8, 2012.

[29] R. Layne, T. M. Hospedales, and S. Gong, "Attributes-based re-identification," in *Person re-identification*, pp. 93–117, Springer, 2014.

[30] P. Sudowe, H. Spitzer, and B. Leibe, "Person attribute recognition with a jointly-trained holistic cnn model," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV)*, pp. 87–95, 2015.

[31] J. Zhu, S. Liao, Z. Lei, and S. Z. Li, "Multi-label convolutional neural network based pedestrian attribute classification," *Image and Vision Computing*, vol. 58, pp. 224–229, 2017.

[32] D. Li, X. Chen, and K. Huang, "Multi-attribute learning for pedestrian attribute recognition in surveillance scenarios," in *3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pp. 111–115, IEEE, 2015.

[33] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev, "Panda: Pose aligned networks for deep attribute modeling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1637–1644, 2014.

[34] J. Zhu, S. Liao, D. Yi, Z. Lei, and S. Z. Li, "Multi-label cnn based pedestrian attribute learning for soft biometrics," in *2015 International Conference on Biometrics (ICB)*, pp. 535–540, IEEE, 2015.

[35] K. Yu, B. Leng, Z. Zhang, D. Li, and K. Huang, "Weakly-supervised learning of mid-level features for pedestrian attribute recognition and localization," *arXiv preprint arXiv:1611.05603*, 2016.

[36] Z. Z. Dangwei Li, Xiaotang Chen and K. Huang, "Pose guided deep model for pedestrian attribute recognition in surveillance scenarios," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2018.

[37] P. Liu, X. Liu, J. Yan, and J. Shao, "Localization guided learning for pedestrian attribute recognition," in *British Machine Vision Conference (BMVC)*, 2018.

[38] C. Tang, L. Sheng, Z. Zhang, and X. Hu, "Improving pedestrian attribute recognition with weakly-supervised multi-scale attribute-specific localization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 4997–5006, 2019.

[39] M. S. Sarfraz, A. Schumann, Y. Wang, and R. Stiefelhagen, "Deep view-sensitive pedestrian attribute inference in an end-to-end model," *arXiv preprint arXiv:1707.06089*, 2017.

[40] X. Liu, H. Zhao, M. Tian, L. Sheng, J. Shao, S. Yi, J. Yan, and X. Wang, "Hydraplus-net: Attentive deep features for pedestrian analysis," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 350–359, 2017.

[41] F. Zhu, H. Li, W. Ouyang, N. Yu, and X. Wang, "Learning spatial regularization with image-level supervisions for multi-label image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5513–5522, 2017.

[42] N. Sarafianos, X. Xu, and I. A. Kakadiaris, "Deep imbalanced attribute classification using visual attention aggregation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 680–697, 2018.

[43] J. Wang, X. Zhu, S. Gong, and W. Li, "Attribute recognition by joint recurrent learning of context and correlation," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[44] X. Zhao, L. Sang, G. Ding, Y. Guo, and X. Jin, "Grouping attribute recognition for pedestrian with joint recurrent learning.," in *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3177–3183, 2018.

[45] N. Sarafianos, T. Giannakopoulos, C. Nikou, and I. A. Kakadiaris, "Curriculum learning for multi-task classification of visual attributes," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 2608–2615, 2017.

[46] E. Bekele and W. Lawson, "The deeper, the better: Analysis of person attributes recognition," in *14th IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, pp. 1–8, IEEE, 2019.

[47] Y. Deng, P. Luo, C. C. Loy, and X. Tang, "Pedestrian attribute recognition at far distance," in *Proceedings of the 22nd ACM International Conference on Multimedia*, pp. 789–792, 2014.

[48] D. Li, Z. Zhang, X. Chen, and K. Huang, "A richly annotated pedestrian dataset for person retrieval in real surveillance scenarios," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1575–1590, 2018.

Appendix A

Age classification details

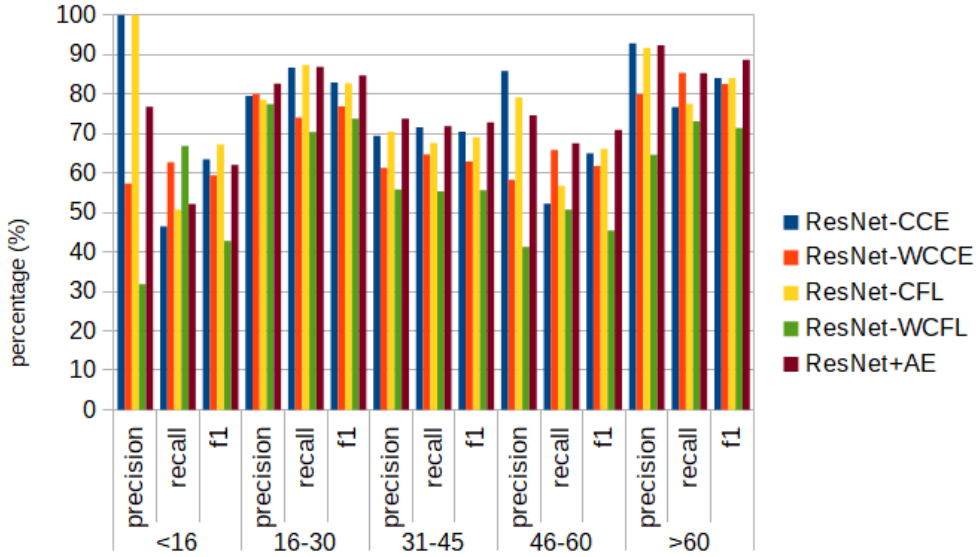


Figure A.1: Detailed classification performance per age-group in the PETA dataset (in %).

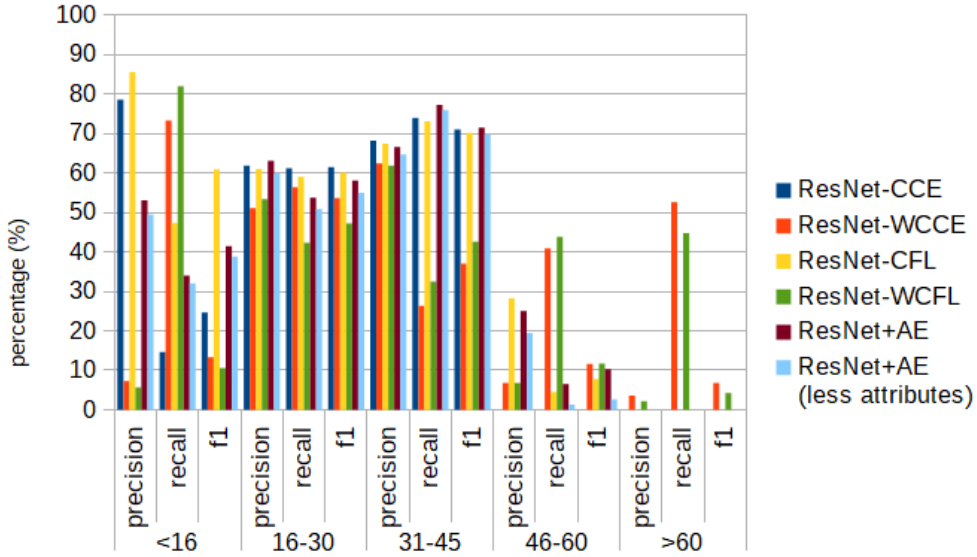


Figure A.2: Detailed classification performance per age-group in the RAP v2 dataset (in %).

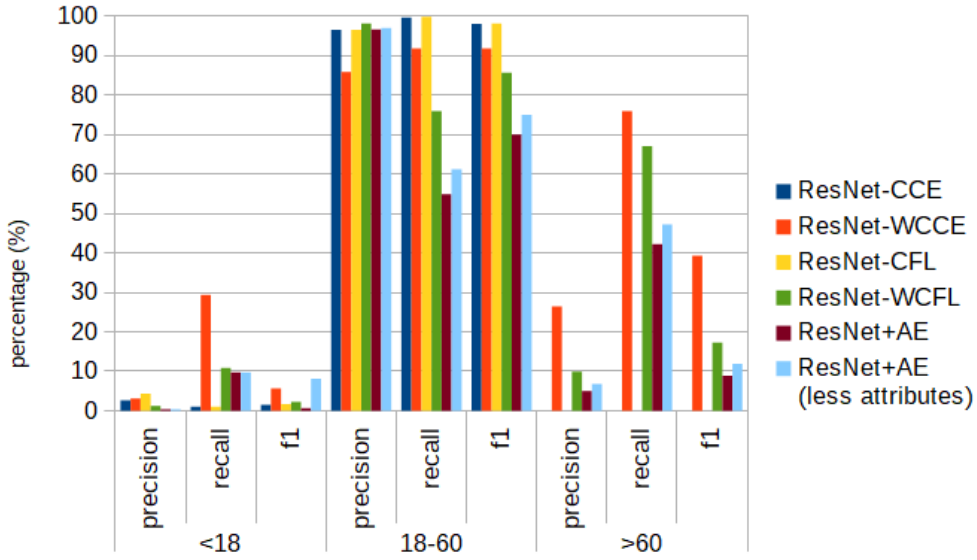


Figure A.3: Detailed classification performance per age-group in the PA100k dataset (in %).

Short Biography

Georgia Chatzitzisi was born in Kozani, Greece in 1993. She holds a MEng degree from the Department of Computer Science & Engineering, School of Engineering, University of Ioannina, from where she graduated in 2018. During that year, she became an MSc student at the same institution under the supervision of prof. Christophoros Nikou. Her research interests are directed towards bringing machine learning techniques into computer vision problems.