Università degli Studi di Salerno

Dottorato di Ricerca in Informatica e Ingegneria dell'Informazione
Ciclo XXXIII – a.a. 2019/2020

TESI DI DOTTORATO / PH.D. THESIS

Intelligent embedded systems for facial soft biometrics in social robotics

VINCENZO VIGILANTE

SUPERVISOR: **PROF. PASQUALE FOGGIA**
DR. NICOLA STRISCIUGLIO

PHD PROGRAM DIRECTOR: **PROF. PASQUALE CHIACCHIO**

Dipartimento di Ingegneria dell'Informazione ed Elettrica
e Matematica Applicata
Dipartimento di Informatica

Contents

Program testing can never
prove the absence of bugs

- Edsger Wybe Dijkstra -

Chapter 1

Introduction

1.1 Application context

Social robotic systems are electro-mechanical systems whose main task is the social interaction with their human users. Popular applications include elderly care [1, 2] and autism treatment [3], but also provide guidance to visitors in public places [4, 5, 6]. The proposed robots often include sophisticated perception systems to achieve a better, more natural interaction.

The authors of [1] developed a robot with the intent of enabling independent aging in place. In their project the benefits provided by the robot are three-fold: decreasing loneliness, support in household tasks, medical and social assistance through remote communication. To do that, they build a wheeled robot and provide it with navigation and mapping capability, human pose detection based on RGB-D imagery, gesture recognition, object recognition and a touch screen and voice interface. The effort in this work is more oriented to practical considerations such as help with the mundane tasks and fall detection and prevention rather than social interaction; even though the participants in the experimental study appreciate the robot, they confess that they would prefer being taken care of by a human.

Similar shortcomings are found in different works. The Hector robot [7] (Figure 1.2) aims to support mildly cognitively impaired

Figure 1.1: A social robot greets participants to a conference

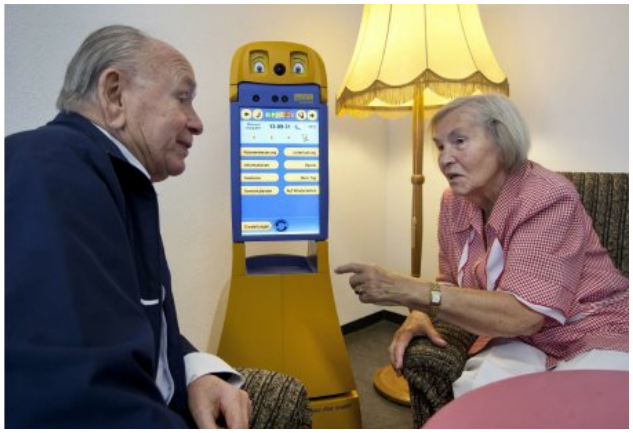

Figure 1.2: The Hector robot supporting an elder couple

Figure 1.3: The Care-O-Bot robot displaying its manipulation capabilities

people and help them, rather than with physical manipulation, providing social and cognitive support. The robot talks to the user, shows initiative, and has a certain personality. They conducted a short user trial (2 days per user) and found out that the ability of the robot to show initiative is a key factor in the perceived acceptance and improve the enjoyability, while technical insufficiencies (i.e. imprecise perception) hinder the naturalness of the experience, to the point that interaction through a graphical interface is preferred to speech.

Care-O-Bot [8] (Figure 1.3) can navigate indoor, execute manipulation tasks and act as a walking support. The earlier versions of the platform include a touch screen for interaction as well as regular voice commands. In their 4th iteration [9] the authors recognized that more attention should be devoted to the robot capability to generate empathy through the pursuit of human likeliness; they aim to achieve this through improved multimodal feedback as well as better understanding of the context.

RHINO [10] (Figure 1.4) guides visitors through a museum. It is an early implementation of such application and is not very focused on natural interfacing. MiviaBot [6] (Figure 1.1) implements the same task 20 years later with increased attention to the

Figure 1.4: The RHINO robot in action

interactive aspect; it integrates multiple visual cues such as gender and age for determining the best way to address people and provide a personalized interaction.

iSocioBot [5] (Figure 1.5 was designed for public events and interacts through a series of predefined questions and replies. It is designed for shorter interactions than a companion robot is, and integrates limited context awareness, i.e. face recognition and tracking, using eye contact to establish a minimal degree of empathy. The authors conclude that the next iteration of the robot should make the dialog more personal by taking into consideration the identity of its interlocutor.

SPENCER [4] aims to guide passengers through airports. The focus of the work is on social aware navigation rather than verbal interaction. Nevertheless, the system employs multiple visual cues such as body posture, gender, age, head pose, spokesperson detection and object detection. Some of those clues are used to apply culturally dependent social rules and so they are critical for a correct social interaction, such as appropriate approaching speed and direction (according to proxemics) and appropriate address-

Figure 1.5: The iSocioBot robot in two different design iteration

ing of people in a group. In a crowded settings the authors find that audio is unreliable so this intensifies the importance of visual clues to obviate.

Popular commercial robotic platforms, such as the Pepper Robot are being employed in shops, expositions, hotels. This and other similar platforms, though, are not able to be completely autonomous in their interaction since they fail to completely gain the trust of their interlocutor. This limits them to be only used as an initial introductory step and the intervention of a human clerk is needed to help the customer with their requests.

The bottleneck of the communication capabilities of those social robots used to be comprehension of the natural language or the ability to understand spoken words. In the latest years commercial systems have been able to achieve stunning results in challenging environments, and this is especially true for commercial applications, where companies have access to a huge amount of private data [11, 12].

While there is still work to be done in this area, the bottleneck is shifting: even with near-perfect transcription of speech and rea-

sonably reliable interpretation of intents, the interaction fails to feel natural [13]. Research in the field of psychological and anthropological aspects of human-robot interaction shows that the acceptance of a robotic interlocutor by a human subject requires the robot to provoke the same emphatic responses that are observed in human-to-human interaction [14]. To elicit those responses it is observed that the robot needs to exhibit human-like behaviors. Coarse physical resemblance is preferred over more accurate reproduction of human physiognomy, and the more similar the looks are, the more accurate the behavior needs to be [15]. As testified indirectly by the user trials that accompany the implementations described above, and direcly by numerous other studies [16], achieving human acceptance in social robotics is much more a matter of similarity in behavior than of physical similarity. It emerges that correctly answering to questions is not enough anymore: effective social robotics require a coordinated application of different disciplines, including electro-mechanics, computer science, psychology and neuroscience, supported by a robust perception of both intents and other contextual cues.

Human behavior indeed relies on the exploitation of social cues such as posture, gait, facial expression, personal characteristics such as gender and age of the interlocutor, and in general the perception of the context. The integration of all those cues into the reasoning and dialog capabilities of the robot would enable the unit to be perceived as a peer by a human interlocutor [17], thus to be trusted as unique interface, thus eliminating the need of additional human operators.

To use different words: the road to truly autonomous social robots passes through perception of contextual clues and correct integration of those into the behavioral routines. The reliability of those clues is then one of the keys towards a successful, natural, social interaction.

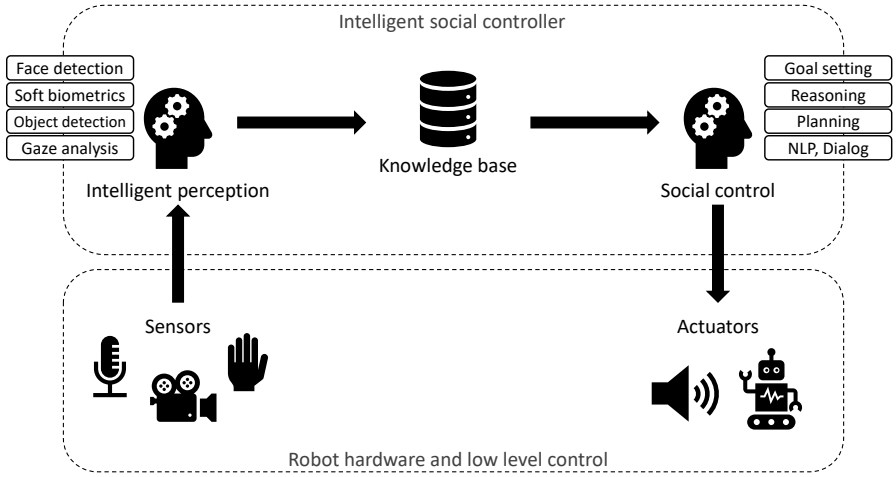

Figure 1.6: General architecture of a social robot. A non-exhaustive list of tasks is reported for each subsystem as an example.

1.1.1 Architecture of a social robot

In the control of a social robot, the role of intelligent perception is to feed a knowledge base from which information is then sourced to guide the interaction. Such an "intelligent social controller" operates at a higher level with respect to the traditional robot control as shown in Figure 1.6.

The decoupling between the two modules of perception and control is obtained through the use of the knowledge base. This decoupling is important since not all the available data is utilized at all times. Rather, the module that implements socially-aware robot control determines the relevance of each piece of information as it queries the base.

Several implementations have been proposed concerning the robot control algorithm. One central concept is the mental state of the robot, which includes its motives, beliefs, desires and intentions [18]. Sociality is focused around interaction: this means that

the robot must be able to understand the mental state of others as well as its own. The control algorithm keeps track of the state of mind of the robot and of the one of other individuals in its environment, and sets its goals consequently. Goals guide planning thus actuation of the output behaviour.

In OpenPsi[19] a similar framework is described and implemented, where the robot state of mind is focused around the concept of *demands*, which are the ultimate desires of the robot. Those demands include for example integrity, i.e. avoidance of pain or even affiliation, the acceptance of the robot by its social group. An *urge* arises when any of this demand is not fulfilled, and this sets the goal for the robot planning and action. Their framework is based on Dietrich Dorner's Psi-Theory about human cognitive processes, emotion and motivation that guide intentions and behaviours.

OpenPsi is developed as an evolution of the OpenCogPrime (OCP) framework [20], which implements a cognitive architecture for a social robot. The framework uses a graph database as its knowledge base and integrates a reasoning engine in its social control module. The OCP framework also integrates subsystems for natural language processing (NLP), one for understanding (input NLP) and for communicating (output NLP)

One example application of this framework is learning new behaviours by imitation and reinforcement, which emerge naturally thanks to the acquisition of positive and negative feedback from other agents combined with the nature of the demands of the robot. This application has been experimented in the project known as OpenPetBrain [21], which features a pet dog as a virtual simulated agent which interacts with simulated people.

This kind of control technology can be ported from the virtual world to physical robotic platforms only if there are adequate perception systems capable of populating the knowledge base reliably with information from the real world. For example, a robot which is supposed to learn behaviours by imitation and reinforcement must be capable of reliably determining the emotional reaction of people around, in order to derive its feedback.

The aim of the thesis is indeed to develop perceptive algorithms able to support a cognitive architecture for a social robot deployed in a realistic situation.

The presence of NLP in this kind of framework is extremely beneficial when our agent is supposed to support an android robot rather than a simulated pet. A social agent is considered so if capable of interactive, communicative behaviour [18], and the verbal channel definitely conveys a significant part of the information in an human-to-human interaction. Therefore dialogue represents an important means through which the robot communicates. In the context of Figure 1.6, dialog takes place in the social control module, and may be guided by the same motivations described above.

The dialog system may be quite complex itself. Part of the dialog can be generated automatically following the social rules of conversation. According to literature [22], when the robot assumes the role of listener, several types of responses can be automatically generated, drawing from predefined categories (backchannels, repeats, elaborating questions and so on) and implementing the so called "attentive listening" where the main objective is show interest and empathy and encouraging the user to keep talking. If the robot is required to talk about specific topics, for example, to give information, those interactions will be scripted separately with an approach derived from the chatbot literature [23].

Whether the dialog is scripted or automatically generated, the robot should be able to personalize its answers based on what is stored in the knowledge base: for example having emotional information about the interlocutor available is very beneficial in order to correctly show empathy; [24] design a dialog model for a social robot and highlight the importance of perceiving social cues such as gaze and gesture to be integrated into their dialog: understanding deictic gestures such as pointing heavily relies on the availability of contextual information, for example if the interlocutor is pointing at something or referring to "this or that", the dialog module should be able to resolve the reference by looking up where the user is pointing, and what object is there, or what is the hands of the user when he talks about "this object".

1.1.2 Facial soft biometrics for social robotics

While classical biometry aims to establish the identity of an individual in a natural and reliable way [25], the term *soft biometrics* was coined to describe those characteristics that only provide some information about the subject, but are not able to individually authenticate the person, due to lack of distinctiveness or permanence [26]. This kind of information includes gender, age, facial expression, presence of a beard, weight, height, color of the clothes, and much more. Their utility, beside being a quick and dirty way of identifying people in the short term [27], or a way to improve the reliability of identity recognition [28], lies in the fact that soft biometric traits are established and time-proven by humans: they are created in a natural way by humans with the aim of distinguishing their peers. For this reason they represent a fundamental asset to artificial intelligence systems whose target is to blend with people imitating their behaviour.

A significant fraction of the soft biometric traits that we listed so far as examples have something in common: they can be extracted by the analysis of the face alone. Arguably, the face contains the most information about an individual, that is why we have our face in our passport photos and driving licenses, that is why we refer to "that person" by mentioning their gender, their age and color of their hair.

A verbal interaction system integrated in a social robot as described in the previous paragraph will use soft biometric information to contextualize its speech and its understanding.

For instance, age is crucial to understand how a person should be addressed: we may be used at being addressed by automated systems in a random way, sometimes too formal, sometimes not enough. A robot that tries to blend in human society would create a cognitive dissonance if it were to address a younger person in a formal way, or an older person in an informal way, so this information shall be taken in consideration, when available.

In addition to formality, children shall be addressed in a completely different way: children have scarce knowledge of the world

and its facts and conventions; interaction with a little child will not reference any complex information that the child may not know about, or include adult content that the child will not appreciate. Talking to elders may also require a different approach than talking to younger people; for example, few elders are interested in the latest trends of technology or make use extensive use of social networks. Would you reference memes from the internet to a stranger older fellow? Would you talk technobabble to them, unless they do it first? A robot should use the same heuristics, and age recognition allows them to.

Those are only a couple examples of how the presence of age in the knowledge base of the robot can benefit the naturalness of the social verbal interaction. Additional decision making processes will benefit by it: as seen in previous work for instance, the use of age allows to understand which members of the group should be addressed. If a family approaches the robot in a public environment, appropriate interaction requires the understanding of the family roles, distinguishing for example the parents from the children. A waiter-robot should not hand the bill to the children. A robot programmed to offer guidance, should only give complex directions to the adults of the family, since they are the ones supposed to lead the group.

Additionally, soft biometrics are always needed for understanding deictic speech: for example, when its interlocutor references "her", the robot should be able to link the pronoun to the identity, and doing that requires that it knows the gender of the people in its environment.

The gender of the interlocutor is also needed for correct conjugation of words in languages such as Italian, French or German, that have specific rules concerning this topic.

Finally, understanding facial expressions unlocks a whole other non-verbal communication channel to the use of the robot. The main ingenuity that is attributed to robots is their incapacity to understand emotions. A robot that is capable of reading facial expression can understand the subtext of any spoken utterance. A simple utterance such as "I'm fine" can have multiple meanings

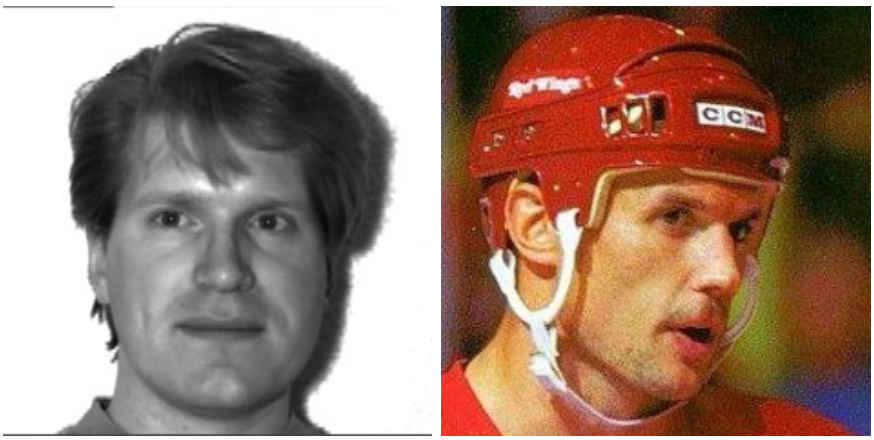

Figure 1.7: A face from the yalefaces dataset [30] (left) and one from the more recent VGGFace2 dataset [32] (right)

depending on the underlying emotion. Talking about macabre topics may be part of a joke or may be the tale of a tragic event: a social robot should definitely be able to distinguish the two cases, in order to react appropriately.

1.1.3 Application constraints

Automatic systems for extracting soft biometrics have come a long way. From the early proposed methods [29, 30] to the more sophisticated recent approaches [31] machine learning algorithms have been applied to problem of extracting information from a face image, with a varying degree of success on the different tasks that belong to the category facial soft biometric applications.

One aspect that guided the evolution of those method is the possibilty of working in unconstraned settings. Early methods, in fact, could only operate in controlled conditions, where the background is neutral, the face is still and in sharp focus, the lighting is appropriate and there is no occlusion. In those conditions they were able to perform very well on the standard benchmarks from their time. While those conditions were reasonable to expect in a collaborative setting, such as a face-based authentication system,

the methods from the early times were not expected to work in uncollaborative scenarios, where there is severe variability in pose and lighting, where occlusions may happen and the image could be affected by different kinds of corruptions. Such methods would not be applicable, for instance, to a setting such as social robotics, where naturalness is key, where the information must be extracted from the face in a transparent way, and the user is expected to be in an unknown environment, in an unknown pose, and the image is acquired with whatever camera is available. Fortunately methods got better while and benchmarks got more complicated to measure the accuracy of those methods in more realistic conditions; in Figure 1.7 two images from two different benchmark are compared that are about 20 years apart from each other: the image from the older dataset features a white background, soft lighting and frontal pose, while the image chosen from the more recent dataset shows significant occlusion, noise and partially lateral pose. Those are fairly extreme examples in both ways, but they convey how the practices in collection datasets have changed to accomodate the robustness requirement of new applications. It turns out that deep convolutional neural networks (DCNNs) are fairly good at dealing with all this variability, definitely better that older methods, but there is still work to do in this area, as it will be shown in the following of the thesis.

Aside for accuracy-related concerns, one aspect of interest is the amount of resource needed for performing each task. Social robots are often mobile platform, implying that they are battery powered and equipped with embedded computing devices. Those devices must perform all the computation needed for the robot to work, including the extraction of contextual information such as soft biometrics through the methods that will be described in this work. This means that the processing power available is limited and so is memory. Furthermore, in an interactive setting such as social robotics, latency time is crucial. In a dyadic conversation the typical silence time between utterances is 100 to 300 milliseconds [33]; having longer processing times would make the conversation awkward before saying the first word, frustrating the

effort and voiding the motivation for doing the prediction in the first place. For this reason, prediction must happen in short times, despite the limited availability of power. One alternative approach would be to offload computation to an external device, reached via network, for example in a "cloud" configuration. Such a solution has been used in the past and is viable in certain conditions but it causes a robustness tradeoff in many others: wireless connections work reliably in nominal conditions, but in a crowded place their speed plummets or they cease working altogether due to electromagnetic interference. This is not a rare occurrence, it happens consistently in application environments of our interests, such as museums and expositions. For this reason, in this work we neglect this setup and study the behaviour of our proposed systems in the embedded scenario.

One disadvantage of DCNNs is that they tend to be resource-hungry. For example, VGG-16 [34], the most commonly used architecture for face-related tasks, requires 527 MB for the storage and more than 13 billion operations to process one input image. Embedded systems are slowly catching up, but still today with cutting edge technology, such a load is non-trivial for an embedded device that needs to run with strict constraints in terms of power consumption, heat dissipation, space occupation and manufacturing cost. The allowed timespan of 100 300 is not at all minuscule by modern computing standard, but it should be considered that in that timespan more than one single task must take place. For this reason and the ones listed before, it is crucial to research ways to produce DCNNs that are fast and slim, but that retain the reliability and accuracy that made large networks so useful in the first place.

1.2 Motivation and thesis overview

This thesis aims to design and evaluate vision based methods for efficient and reliable soft facial biometrics in the context of social robotics: we addresses the concerns listed, researching the features

and allowing for the development of a facial soft biometrics submodule for a social robot, aware of the constraints of the specific application, namely the need for running in real time on an embedded system, with limited resources and limited latency time and the need of the system to be resilient to all kinds of natural disturbances that will occur in an unconstrained environment while interacting with uncontrained interlocutor.

The work is organized as follows: in Section 1.3 we discuss the state of the art for facial soft biometrics, at first in general and then in particular, addressing each task of interest for this work, with its peculiarities and solutions, surveying the methods and the datasets. Then we survey the efforts made towards the solution of the two peculiar issues that we identified, namely resource efficiency for embedded applications and robustness to corruptions and perturbations from the real world.

In Chapter 2 we discuss design principles that allow for faster network architecture and propose a novel architecture to efficiently predict gender from faces. In Chapter 3 we discuss the challenges related to the task of ethnicity recognition and evaluate the accuracy of different network architectures on a novel dataset that suits the constraints of our problem In Chapter 4 we discuss in detail the image corruptions that occur in realistic scenarios as well as the perturbations that may affect the appearance of a face in a sequence of frames and analyze their effect on the network accuracy, while proposing possible solutions. In Chapter 5 we discuss the challenges that need to be faced when developing an age predictor and we propose a training methodology that is able to achieve state-of-the-art results on a realistic images while retaining reasonable computational complexity.

In Chapter 6 we draw our conclusions and sketch some proposals from future research directions.

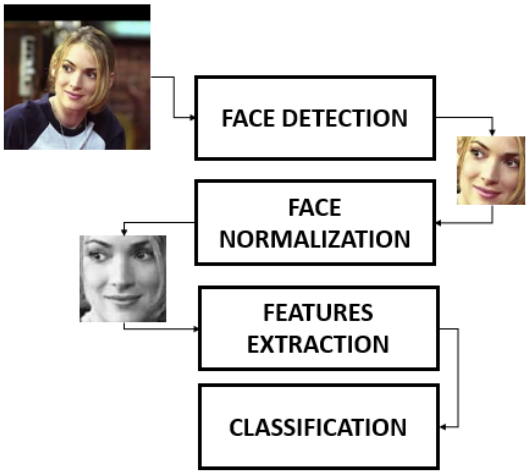

Figure 1.8: Functional processing pipeline of a typical system that performs face analyisis.

1.3 State of the art

1.3.1 Facial soft biometrics

The typical pipeline for a face analysis system is shown in Figure 1.8 and consists of the following logical steps: (1) face detection; (2) face normalization/alignment; (3) feature extraction and classification. The framework is generic and shared among all the tasks concerning face analysis, with minimal differences.

In the first step, the position of the face in the image is identified with model based approaches, such as [35], [36] and [37]; more recent approaches exploit one-shot cnn-based detector [38, 39]. The chosen algorithm must be accurate and efficient, since searching for a small face in a large image can be computationally taxing. Missing or significant imprecision in the detection and localization of the face will obviously mislead every subsequent processing step, hence the method needs to be accurate. While early methods such as [35] were only able to detect frontal faces, modern detectors will return faces in a variety of poses. The use of a dataset with sig-

nificant pose variability will require to use an unconstrained face detector; processing steps downstream must be able to deal with faces in such unconstrained poses.

In the second step, the face image is processed in order to achieve a canonical representation of the face in terms of position and appearance; having a canonical representation of the face reduces the amount of variability which is irrelevant to classification. This step is crucial when the downstream step (feature extraction) is not particularly effective in dealing with changes in illumination, pose, occlusions and so on; having such variability reduced, significantly improves the discrimination capability of the feature, thus improving the final classification performance. On the contrary, if the feature extractor is particularly robust, the normalization step can be omitted or reduced to a few simple operations; reducing the complexity of the normalization step is helpful since complex normalization algorithm can be slow and/or error-prone: if feature extraction relies heavily of the correct operation of normalization, errors in this phase will almost certainly lead to randomness in the classification result.

In order to compensate for pose variability, the face pose must be first established: to that aim, a suitable detection algorithm is used to locate the facial landmarks inside the face region. The facial landmarks are known points in the face that are easy to identify for a human: the tip of the nose and the centers of the eyes have a major role in determining the pose, but are not the only ones. Once the facial landmarks are identified, the image can be scaled and rotated to match a predefined template: an affine transformation is computed to bring the eyes and the nose in fixed locations. This kind of normalization can fully compensate for in-plane rotation of the face while other kinds of rotations of the head are preserved in the image (Figure 1.9 top). More sophisticated methods may exploit more landmarks to perform more radical transformations; full frontalization approaches [40] will distort the image in an effort to bring the face in a frontal pose. The difference between the two solutions can be appreciated in Figure 1.9. However, frontalization methods have two main drawbacks: they

Figure 1.9: Affine transformation (top) compared to full face frontalization (bottom) [1].

can be extremely slow and they introduce a consistent deformation of the face, possibly corrupting the characteristics of interest in the face. For this reason, most proposed methodologies for soft biometric extraction implement either no pose normalization or a simple solution based on affine-transformation; they typically neglect face frontalization, leaving the burden of dealing with off-plane rotations to the feature extractor.

Common approaches for compensating variability in illumination include histogram equalization [41], contrast stretching[42], plane fitting [43].

In the third and fourth step, the actual classification takes place; the feature extraction step aims to produce a representation of the face with lower dimensionality. All the irrelevant variability in the input image should be discarded in this phase, while the most representative features to the task at hand must be encoded in the resulting feature vector.

Three main strategies may be identified for feature extraction: (1) handcrafted features, (2) trainable features or (3) a combination of them.

Handcrafted features are carefully designed by humans explic-

[1] Image courtesy of github.com/dougsouza/face-frontalization

itly for the specific problem, while trainable features are general purpose meta-descriptors with a large number of parameters that can be tuned automatically from examples. Trainable features include all the techniques related to deep convolutional neural networks, that employ stacks of convolution operations using filters which are learned from the data.

Finally, the classification step employs machine learning techniques to extract the target biometrics from the feature vectors. Support Vector Machines (SVM) and its variations have been the most commonly used classification methods for a long time. More recently Neural Network based classifiers are most commonly used because they can be trained jointly with CNN feature extractor [31].

In the subsequent paragraphs we will survey the literature for each specific biometric task covered in this thesis.

1.3.1.1 Gender recognition

Gender recognition from faces is one of the basic capabilities of the human beings. Extending this capability to machines is of great interest in many application areas, beside social robotics and conversational agents. *Digital signage* is becoming more and more established as an application; in this scenario a digital billboard is used in place of a static one, to show dynamic advertisements, customized depending on the characteristics of the person looking at the monitor itself. Gender recognition can be profitably used in this area, since it allows to boost the effectiveness of the advertisement campaigns. Being digital signage an older application than social robotics, we appreciate that the push for developing efficient and effective gender recognition algorithms has been around for a long time [44].

Traditionally, handcrafted features have been used for distinguishing men and women. Researchers were able to identify obvious clues that can effectively distinguish gender in many situations, such as beard for man and long hair for women, even though they do not realize a perfect separation. Following this observa-

tion methods have been proposed that exploit features such as color [45] and texture [46] or a combination of the two [47]. This methods are typically fast but they are not robust in that they are only effective in simpler cases and they are confused by more challenging examples.

There is neurophysiological evidence that the shape of the jaws and cheekbones is the main feature that human use to recognize the gender of their peers [48]. Indeed, it is known that estrogen allow fat to be developed in the region around the cheeks, making the facial traits of a woman rounder and softer while those of a men are typically harder. Based on this consideration, methods were proposed to exploit shape information, for example through the HOG descriptor [49], or fusing shape information with different kinds of features [47, 50].

[51] is one of the last methods to use handcrafted features (Local Binary Patterns, LBP) and it achieved 96.86% accuracy on the challenging Labeled Faces in the Wild (LFW) dataset, 2% more than previous attempts thanks to the "unreasonable effectiveness of data"[52]; they used in fact a huge automatically annotated dataset of 4 million instances to train a linear Support Vector Machine (SVM).

The real turning point for accurate gender recognition though was the introduction of automatic feature learning. This kind of systems in fact are much more flexible with much more parameters, thus they are able to better exploid said unreasonable effectivenes of data. Trainable COSFIRE filters [53] employ a bank of Gabor filters with learnable parameters, allowing the msot discriminant filters to be learnt from data. The filters are tuned on a small set and then used to train a classifier (typically SVM). Gabor filters, that are not cheap to compute. The general trend in fact is that trainable features tend to be more accurate than handcrafted ones, but less efficient.

CNNs were the final evolution of the trainable methods taking full advantage from large datasets: they have a massive amount of parameters to be configured (millions) and require huge datasets to reach a good level of generalization. It is known in fact that

the more parameters a learnable system has, the more data is needed to avoid overfitting (curse of dimensionality). [54] use a CNN with three convolutional layers and two fully connected layers. [55] use a deeper CNN with five convolutional layers and three fully connected layer, achieving an accuracy of 97.1% on the FERET dataset. Unfortunately none of them reports results on the LFW dataset, but the FERET dataset is considered to be equally challenging as the LFW dataset.

After those early results, CNNs developed to an extraordinary extent. The main application for CNN has been object recognition, in fact the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [56] has pushed the development of more and more powerful architectures. Those architectures have been ported to different problems such as, indeed, gender recognition.

In a recent work [57], the authors were able train the very deep an powerful ResNet-50 CNN and obtain an accuracy of 99.3% on the LFW benchmark proving the extraordinary effectiveness of CNNs for the problem at hand;

However, this and other very accurate methods need gigabytes of RAM and storage, and billions of floating point operations for a single prediction. Typical processing units available for a reasonable cost on robots and smart cameras are become quite powerful, but not nearly as powerful as those methods would need them to be; From these considerations, it emerges a clear need for a gender recognition method which is both *accurate in the wild* and able to run in *real time on embedded devices*. If those two constraints are met, such a method would be applicable in the most common real-world applications.

The authors of [58] propose an ensemble of CNN models: with reference to the VGG architectural principles, they specifically address the problem of reducing the computational load; they find an optimal architecture in terms of depth, number of feature maps and input size, then they train the best architecture three times and combine them in an ensamble to reach 97.31% performance on the LFW dataset whle significantly reducing processing time. VGG architecture has been also used in [59], where the authors

compare MobileNet and VGG in the field of social robotic, also considering the computational burden. Those and other works testify the attention that is being devoted to porting powerful methods in constrained applications such as social robotics and akin interactive, human-centered fields, such as autonomous driving [60], that require careful design of a real-time capable network architecture [61, 62].

Within this context, in Section 1.3.2.1 we survey previous research in the field of efficient neural network design, while in Chapter 2 we specifically proposed an optimal DCNN architecture specifically tuned for gender recognition.

1.3.1.2 Age estimation

Age recognition is definitely trickier with respect to gender. For starters, age recognition can be hardly considered a classification problem: age is a real number, and so its estimation it may be better suited as a regression task. Also it is not trivial to uniquely identify age from a face image, even for an human observer: one would need to know the date of birth of the person to know the exact age; alternatively the apparent age can be estimated by the looks, yielding an approach known as apparent age; while apparent gender rarely differs from actual gender, the apparent age will contain intrinsic error.

Secondly, correctness of the estimated age is hard to evaluate: is saying that a 80-year old is 79 an error? Is it an error saying that he is 75? Is it an error saying that a 11 year old is 16? It appears that there is no universally agreeable answer to that. For this reason, different evaluation protocols exist for different benchmarks in literature. Some will just evaluate the Mean Absolute Error (MAE) as the mean of absolute differences between estimated and actual age; this will mean that mistaking a 80 year old for a 75 years old will have the same weight as mistaking an 11 years old for a 16 years old, that is probably an undesired effect, since that is not an equally severe error according to human perception [31]. Some datasets may measure the accuracy, divid-

ing faces in classes by age, where each class includes ages from a certain range (e.g. 3-6, 7-12, 13-16, 17-20, and so on); this approach allows to create groups according to what human perception may consider to be equivalent ages. Arguably the most agreeable evaluation system is proposed by the ChaLearn Looking at People (LAP) benchmark[63]: the LAP dataset has every image annotated by multiple people with the apparent age and, for every image, the mean and variance of the annotation is computed. Errors are weighted by the variance in the annotation so that errors on samples that have higher variability in the annotation itself will be weighted less and vice-versa;

More precise explaination of all the protocols will be given in the following, since our experimentation we will evaluate our results according to all those different protocols.

We will now describe the methods and datasets from literature: like for gender, the first efforts to tackle the task of age estimation relied on hand crafted features [64]; however, these techniques were able to achieve reasonable accuracy only in controlled conditions (e.g. frontal pose, high quality, high resolution), while their accuracy dropped when exposed to the variations in lighting and pose happening in real environments [31]. The advent of deep learning greatly allowed for age estimation methods that are significantly more reliable in simple scenarios, and vastly superior in challenging conditions, making possible the design of algorithms sufficiently accurate for real applications [31] [65] [66].

Although very effective, the methodologies based on convolutional neural networks, again, are often slow and resource demanding. Efficient network architectures targeted at biometric analysis exist [58], but they often require a sacrifice in accuracy, while the most accurate methodologies can be extremely bulky and slow [57], namely unusable for practical applications despite their reliability.

A second problem that hinders researchers in the field of age estimation is the absence of a large, reliably annotated dataset. This problem is due to the cost and difficulty of annotating a wide dataset. The one proposed during the LAP challenge 2016 [63], also known as APPA-REAL, is very reliable, being each face

annotated by multiple people as described before; the process is accurate but costly and, in fact, the dataset includes only 7,591 images; for this reason, it is insufficient to train a deep network on its own, and it is only used for fine tuning after the mandatory step of pre-training on a large-scale dataset [31]. The largest dataset for age estimation available in the literature is IMDB-Wiki [67], whose authors adopted a different approach for age labelling. They tapped from the profiles of famous people available into the public image databases of Wikipedia and the Internet Movie Database and automatically annotated more than $500,000$ images obtaining the age from the birth date of the person and the date of the picture; of course, this procedure does not ensure the reliability of the annotations, so much that the authors themselves recommend using the dataset with caution as there are several errors. Similar considerations apply for Cross-Age Celebrity Dataset (CACD) [68], which include around $163,000$ images annotated with the same protocol adopted for IMDB-Wiki.

In absence of better alternatives, IMDB-Wiki is the current standard for pre-training convolutional neural networks for age estimation. However, to obtain state of the art performance, it is necessary to carefully "clean" the dataset in order to consider only the correctly labelled images requiring some labor and getting varying results depending on the exact method chosen for cleaning. The authors of [57], winner of the LAP 2016 competition, applied a combination of automatic and manual filtering strategies to discard almost half of the images and to obtain a cleaner version they call *IMDB-Wiki-cleaned*, unfortunately not publicly available. Therefore, a significant effort is required to design and implement an effective training procedure of convolutional neural networks for age estimation.

In this work, the use of knowledge distillation is proposed to overcome these limitations. Knowledge distillation [69] is a technique used to train small, efficient convolutional neural networks with reduced need of resources and improved accuracy. More information on this technique will be described in Chapter 5, alongside the details of the proposed method and its results.

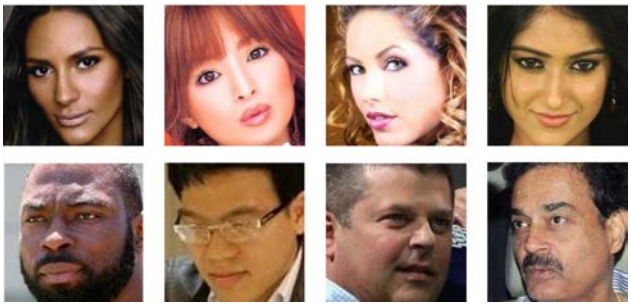

Figure 1.10: Faces of different ethnicities (from left to right: African American, East Asian, Caucasian Latin, Asian Indian)

1.3.1.3 Ethnicity recognition

Ethnicity recognition is a growing topic in the community, as testified by the new datasets and methods [70, 71, 72, 73] that are regularly published in recent times.

Methods are proposed as support for other soft biometrics in applications where variations in ethnicity affects performance (gender recognition, age estimation, identitication) or in forensics applications (ethnicity based subject identification for public safety).

Nevertheless, in a recent comprehensive survey [74] the authors notice that the lack of ethnicity data is one of the main obstacles to the further development of the topic: in the era of deep learning a large amount of data is necessary to train an effective method. The datasets that are currently available have insufficient size, when compared to the ones available for the other facial soft biometrics [75]. As a consequence of this, it has been recently shown [73] that the CNNs trained for ethnicity recognition on the currently available datasets have a limited generalization capability on different test sets.

The lack of ethnicity data can be attributed to the fact that the concept of ethnicity is controversial. It can be defined qualitatively and not quantitatively. Identifying universal distinguishing features is harder than it is for other kinds of biometrics, such as gender: "ethnicity", as intended by humans, has no biological va-

Table 1.1: Public datasets of faces annotated with ethnicity groups.

Dataset	Images (Subjects)	Ethnicity groups
FERET [77]	14,126 (1,199)	Caucasian, Asian, Oriental African
JAFFE [78]	2,130 (10)	Japanese
IFDB [79]	3,600 (616)	Iranian
CASPEAL [80]	30,900 (1,040)	Chinese
MORPH-II [81]	55,134 (13,618)	African, European, Asian, Hispanic, Others
FEI [82]	2,800 (200)	Brazilian
PubFig [83]	58,797 (200)	Asian, Caucasian, African American, Indian
CUN [84]	112,000 (1,120)	Chinese
HUDA [85]	N/A	Saudi Arabia
EGA [86]	72,266 (469)	African American, Asian, Caucasian, Indian, Latin
CAFE [87]	1,192 (154)	Caucasian, East Asians, Pacific Region
LFWA+ [88]	13,233 (5,749)	White, Black, Asian
UTKFace [72]	20,000 (N/A)	White, Black, Asian, Indian, Others
FairFace [73]	108,192 (N/A)	White, Black, East Asian, Southeast Asian, Indian, Middle Eastern, Latin

lidity, since there are no genetic characteristics that allow individuals to be grouped according to the well distinguished "ethnicities" [76], such as the ones shown in Figure 1.10. For example, ethnicity cannot be automatically inferred knowing the place of birth: being the categorization a human construct, it takes humans to manually annotate the categories. Additionally perception of the ethnicity is not universal and different people may not agree on the ethnicity of a person from their looks; they may not even agree on how many ethnicity are there to be distinguished.

We summarize public datasets available for this task in Table 1.1. Those datasets have three main drawbacks.

To begin with, different dataset define different categories. A

standard categorization does not exist, due to the subjectiveness of ethnicity perception; furthermore some datasets are devoted to specific application contexts, for example containing faces from a single macro-ethnicity (e.g. Chinese, Brazilian, Japanese, Iranian, Saudi Arabia); these data cannot be used for the task on their own but they get often integrated into larger datasets. Other datasets have different ethnicities in them, CAFE, FERET, PubFig, EGA, LFWA+, UTKFace and FairFace but they still are inconsistent with each other. This landscape also complicates comparison between methods, rendering some results not reproducible.

A second issue is that the datasets are not very large, having thousands of images at best and typically not well balanced among the different ethnicities (i.e. some ethnicities contain thousands of samples while other only a few hundreds). As said before, deep networks benefit from huge quantities of data in their training process. FERET [77] dataset has been used very often for ethnicity recognition; unfortunalety it does not have an official division between training and test set, rendering results not reproducible nor comparable. MORPH-II contains many samples of European and Asian people, while the other classes only amount to the 4% of the whole dataset. EGA [86] merges 6 pre-existing datasets (including FERET); its main drawback is that it contains few subjects per ethnicity, which is not ideal for training, however it includes more than $70,000$ samples and it is used very often as a benchmark. In deep learning applications, LFWA+ and UTKFace have known the best success; they use fairly common labels (White, Black, Asian) and only contain a few thousands images. Cross-dataset experiments demonstrated that the deep networks trained on those datasets are not able to generalize on different test sets [73]. FairFace is definitely the largest, most complete dataset currently available for ethnicity recognition, since it is composed by $108,192$ images, with a clear, official division in training and test set. Furthermore it is balanced, having the samples almost equally distributed among 7 different classes. The authors of the dataset show the superior generalization capability achieved by training on FairFace rather than on LFWA+ and UTKFace [73].

Finally, to the best of our knowledge, none of the existing datasets is annotated by people of different ethnicities. Involving people of different ethnicities in the annotation procedure allows to minimize the effect of the "race bias", where people's perception of the ethnicity depends on their own belonging [74].

The brief survey of the dataset literature exposes the necessity of collecting a large large and heterogeneous dataset, that is annotated for ethnicity recognition in a reliable manner, taking into account the race bias defining and annotating the ethnicity groups.

Due to the lack of data, Ethnicity recognition literature is more skewed towards handcrafted features than it is for the other soft biometrics, since such systems do not rely as heavily on data. Common traits that are considered in designing such features include the color of the skin, the shape of the eyes, the position of the facial landmarks. Most recent methods however are based on automatic representation learning, mainly through CNNs and they manage to obtain the best results.

Among the handcrafted features, the skin color is the most often used; [89] trains an SVM classifier using color values and color histograms and obtains a result of 78.5% on the FERET dataset, using the classes Black, White and Asian. The method from [83] performs prediction of multiple attributes including ethnicity on their own PubFig dataset using color as a feature. Other methods use feature selection with algorithms like KCFA [90] or Adaboost [91]. However, color features are not invariant to illumination, rendering them not robust in real environments. Other approaches use texture and shape descriptors, or a combination of them, relying on features that do not depend on skin color. The authors of [92] use Haar features and Adaboost and measure their performance on a private dataset with 3 classes. Gabor features are used in two other works [93, 94] using Adaboost for feature selection and SVM for classification. The methods in [95] and [96] feed LBP histograms to a KNN classifier; the first uses PCA to select the most discriminant LBP and Haar features, while the second uses the Weber Local Descriptor (WLD) [97]. Many of these

works collect their own datasets to overcome to the drawbacks of the existing ones.

In [98] the authors successfully train a CNN for different face-related tasks, including ethnicity recognition. To overcome the drawbacks of the dataset, they make abundant use of data augmentation techniques. Their accuracy on on the FERET dataset with 3 classes (White, Asian, Other) is 93.9% . A more recent work [70] achieves a result of 98.9% on the same dataset using a fine-tuned VGG-Face architecture as a feature extractor which is appliend on a aligned version of the face; SVM in used for classification.

On the MORPH-II dataset Yi et al. [99] achieve 99.11% accuracy. They apply 23 different shallow multi-task CNNs to classify patches taken from the aligned face image at 4 different scales. They fuse the decision in the output layer, which provides both the ethnicity group and the age estimation. The authors only use Black and White as classes and ignore Asian, Hispanic and Other since the first two classes alone represent the 96% of the dataset. Hu et al. [71] obtain 98.6% accuracy using a multi-task version of the AlexNet architecture. The LFW+ dataset is annotated with different facial attributes, including ethnicity and used for training.

In Guo et al. [100] also realize a multi-task classifier: the faces are detected and aligned, cropped and resized to 60x60 then used in grayscale to extract using "BIF" features, that are biologically inspired. Exploiting a feature selection approach, they can distinguish Black from White people with 99% accuracy. Karkkainen et al. [73] use different datasets to train a ResNet-34 model. The evaluate the accuracy on different test sets to assess the generalization capabilities deriving from the use of each training set. The results show FairFace allows significantly higher generalization than UTKFace and LFWA+, so demonstrating the importance of correctly designing the training set.

From the analysis of the state of the art we draw three main conclusions:

- in recent literature, ethnicity recognition typically repre-

sents only one part of a multi-task system where multiple attributes are estimated, or is used as an ancillary task to a different soft biometric tasks such as gender recognition and age estimation;

- modern methods, including CNN architectures, saturated the capabilities of the datasets that are acquired in controlled laboratory conditions such as FERET and MORPH-II;

- most currently available datasets, cannot provide the network models with generalization capabilities; FairFace is by far the best dataset.

For all these considerations it emerges a need for a public large and challenging "in the wild" dataset, which is realiably annotated with the most common ethnicities, that allows to train and benchmark the new approaches. In Chapter 3 we design such a dataset and then we use it for training and comparing different architectures; we show that the accuracy is improved over the state of the art, when compared on our benchmark and third party benchmarks.

1.3.1.4 Emotion recognition

Emotion recognition arguably is the most representative application in human-centric computing. It plays a crucial role in social robotics, allowing for an better understanding of the social subtext during conversations, thus being the central information in a dialog system that aims for improving empathy.

Different methods and datasets have been proposed for emotional applications throughout the years; recent research even focuses on the fusion of different data modalities, such as video, audio and text, even though video data is found to contain the largest amount of emotional information [101]. At the same time, the fundamental analysis of face images has been considering more and more complex scenarios [102, 103]. The publication of 'in the wild' data sets acquired in challenging conditions and methods

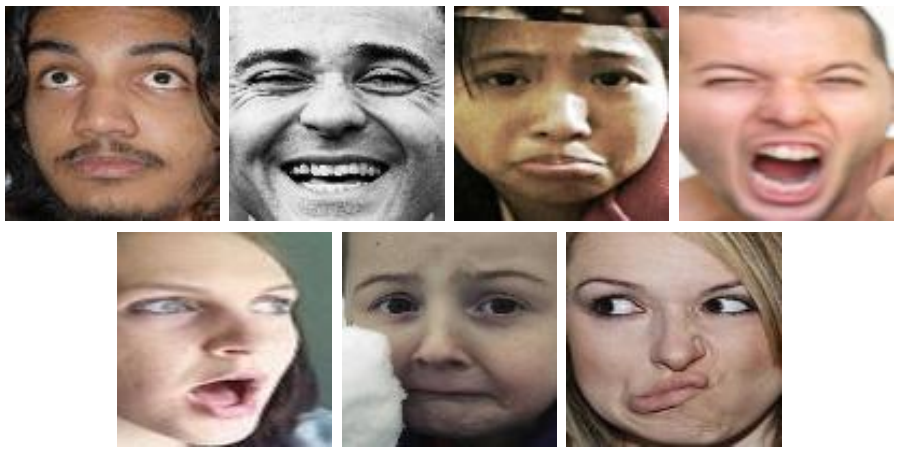

Figure 1.11: Emotional Images from RAF-DB (neutral, happy, sad, angry, surprise, fear, disgust)

that tackle the challenges of those new data sets witness an interest in improving existing approaches [104]. To fulfil the needs of Deep Learning based methods, datasets need to be both representative of real conditions and increasingly large. With such constraints, the design and efficient gathering of such datasets becomes a challenge itself [105].

[106] created the FER-2013 data set, which contains more than $28k$ grayscale images (of size 48×48 pixels) from Google Search. Later, [107] made available AffectNet, which was collected with a similar approach, with almost 1 million images. AffectNet is automatically annotated in part, with about 60% annotator agreement only, making it unreliable for thorough evaluation of classification methods. In general, face emotion recognition data sets have noisy labels, due to the subjective nature of emotion perception. Subsequent works aimed at reliably labeling data sets with redundant information coming from multiple annotators. [108] developed the FERPlus data set including the same images from the FER-2013 data set with annotation improved by crowd-sourcing. Each image is tagged by ten different annotators. [109] obtained the RAF-DB data set (Figure 1.11 by adopting the same approach to annotate

$30k$ facial images from the internet, with 40 annotations per image on average; their effort produced a very reliable dataset despite the subjective nature of the task.

Many techniques were proposed throughout the years to improve facial expression recognition methods. If we consider the RAF-DB benchmark, we observe a trend of improvement in the state-of-the-art approaches. The authors of the RAF-DB data set themselves proposed a method based on Deep Locality-Preserving (DLP) learning; in this method the loss function explicitly addresses the intra-class variance [109] encouraging the activation of the last hidden layer for samples of the same class to have a common centroid in the feature space. They reached an accuracy of 74.2%. [110] trained a 'contrastive' encoder in a double encoder-decoder setting. The learned features are used for generating two images, the original one and a version with neutral-expression. The encoder trained in this setting is used as input for a fully connected classifier. [111] proposed a Multi-region Ensemble CNN (MRE-CNN). Three significant sub-regions are cropped from the face (the left eye, the nose and the mouth) and each is given as input to a double-input network, alongside with the full face image. Their final result is an accuracy of 76.7% using a VGG-16 backbone. [112] designed a neural network trained using Global-Local Attention (gACNN) and a VGG-16 backbone. Facial landmarks were used to compute local attention from patches, then the information was integrated with global attention. The use of attention makes this method particularly robust to occlusions, obtaining an overall 85.1% accuracy. [113] reported an accuracy equal to 87% using covariance pooling, based on the intuition that second order statistics better model the face changes that represent an emotion rather than max or average pooling. SPDNet layers were used to reduce the dimensionality of covariance matrices while preserving the spatial structure. The center loss function, a simplified version of DLP, was used for training. [114] used 3D reconstruction as a method for face frontalization to reduce the variability of the pose of the faces in the images fed to the classifier. An Inception-ResnetV1 architecture pretrained on the VGG-Face2 data set was

then fine-tuned on the RAF-DB data set, achieving 85.1% accuracy.

Many efforts were made to collect in-the-wild data sets, with a high degree of variability and a reliable annotation. Progress were also made to craft methods that achieve higher and higher recognition accuracy. Despite the efforts, the data sets always consist of photographs from the web: they surely portrait a "wild" condition, with a wide variety of poses, conditions, occlusions and backgrounds, but they fail to accurately reproduce typical corruptions that happen on a typical real world setup, such as camera noise, motion blur and so on. No method is currently supported by an analysis of the robustness to such corruptions; this represents an element of uncertainty on the performance to be expected when deploying such systems.

In Chapter 4 we choose 4 CNN architectures with different characteristics and verify their robustness to different kinds of corruptions of the input images; furthermore we propose two approaches to improve the robustness and evaluate their effectiveness.

Additionally, we observe that a sequence of frames acquired from real scenarios will have slight perturbation from one frame to the next, such as adjusting the focus, movements or rotation of the face or just random noise: it is desirable for the prediction to be stable through those perturbations, i.e. the perturbations should not cause the CNN output to change. In Chapter 4 we evaluate the stability of all the proposed methods.

1.3.2 Network optimization for social robotics

1.3.2.1 Efficient CNNs

Convolutional Neural Network have revolutionized the field of computer vision with their superior accuracy, ease of use and extreme flexibility. Since AlexNet improved state of art accuracy in image recognition by 10% in 2012 [115], one research trend has been to design more and more accurate architectures. The improvement in accuracy often comes with significantly more com-

plex structures, typically deeper [34] (i.e. with more cascaded layers) and with more parameters, to the point that a separation emerged in the state of the art between methods that can be used in practice and methods that cannot [116].

Starting on this observation a new research trend developed where proposed methods are evaluated based on different parameters aside from their absolute accuracy on the ImageNet benchmark: main parameters have been identified to be memory footprint, parameters, operations count, inference time and power consumption [116]. Different design considerations affect many of those at once, but each one has its unique effect on the capability of a network to be run in an application setting. For instance, we already highlighted how inference time is a crucial parameter to keep into consideration in the interactive environment of social robotics, because reaction to stimuli must happen in a timely manner; it has been observed and experimentally confirmed that operations count (i.e. the number of multiply-adds needed for every forward pass) is a proxy for inference time. This is not the whole story, though, since modern hardware heavily relies on parallelization capabilities, so the inference time will be determined by the relationship between the way those operation are organized and the organization of the specific hardware [117, 118, 119].

Approaches such as Network Pruning and Deep Compression [120] have been proposed to reduce the storage size required to hold the network parameters with negligible accuracy loss. The authors observe that such a feature allows for significant savings in bandwidth for remotely updating a deployed device. SqueezeNet [121] is a re-designed architecture that stems from AlexNet. It preserves most of its original accuracy while further reducing the storage space by a 510x factor. Incidentally the more efficient design also improves other aspects such as inference time..

MobileNets [122] introduce the depthwise-separable convolution: rather than having all the input channels involved in the convolution operation with all the learned filters, the authors employ a sequence of depthwise convolution (a novel operator) and pointwise convolution (a standard 1x1 convolution). The depth-

wise convolution applies one filter to each input channel, while the pointwise convolution merges the intermediate results pixel by pixel across all channels. With this approach the number of operations is reduced by about 9 times, while the representative power of the network is largely preserved, allowing to achieve high accuracies with fast inference times even on mobile CPUs. Using the depthwise-separable convolution the authors design a family of network architectures called MobileNets that achieve different tradeoffs in terms of accuracy and inference times.

SqueezeNext [123] improves on the MobileNet concept by proposing an hardware-aware design: they observe that depthwise-separable convolution are inefficient on neural network accelerators with a large number of processing elements (PEs) and design an architecture that does not rely on them, taking better advantage on hardware parallelism, raising hardware utilization by 20%.

ShuffleNet [118] introduces channel shuffle for group convolution: building on the concept of pointwise convolution, that are the most computationally expensive operation in Mobilenets, the authors observe that they can be made more efficient if we separately combine a subset of the channels. This idea, introduced by AlexNet [115], appears to defeat the purpose of pointwise convolution, that is to contaminate information between all channels; the authors fix this by introducing a channel shuffle layer that gives the network its name: by shuffling channels they ensure that, in a stack of layers, the information flows efficiently between all channels. Using an ARM cpu, the authors prove ShuffleNet-0.5 to be 13x faster than AlexNet with comparable accuracy, better than previous attempts

On top of their efficient architecture, the authors of MobileNets and ShuffleNet suggest the possibility of applying quantization to obtain further advantages on common processing platforms. Integer operation are much faster and require much less transistors to be executed than floating point operations, resulting in reduced manufacturing cost and power consumption. On top of that it has been proved that, while gradient propagation benefits from larger representations (to alleviate the problem of vanishing gradients),

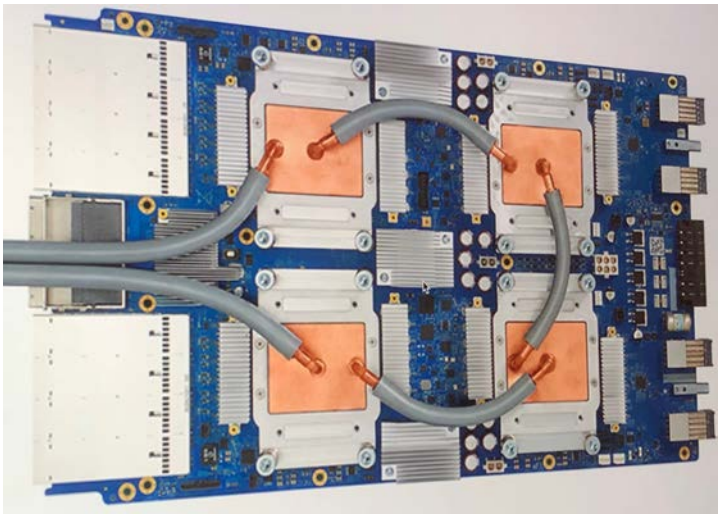

Figure 1.12: Google Tensor Processing Unit 3.0 (250 W power consumption, 32 GB memory, 90 TOPS). Image courtesy of Zinskauf, CC BY-SA 4.0, via Wikimedia Commons

precision as low as 16 or 8 bits is enough for accurate inference. For this reason many approaches have been proposed to quantize the weights and the activations of neural networks [124, 125]. XNOR-net [126] goes as far as only using just 1 bit for weights and activations. To minimize accuracy loss, an established method consists in training the network with full accuracy (32 or 64 bit float), then fine tuning it with weights and activation quantized and artificially constrained to lower accuracy (e.g. 8 bits integer values); the procedure is called quantization-aware training [127].

Manifacturers are taking advantage of these considerations, designing integer-only neural network accelerators. This includes Tensor Processing Units (TPUs) by Google (see Figure 1.12), Myriad X from Intel and many others.

Successive versions of ShuffleNet [117] and MobileNets [128, 129] improve the state of the art efficiency by updating the previous version of each architecture with modern features from the state of the art. MobileNetV2 introduces the inverted residual bottleneck layer: a linear bottleneck embeds the information in a

representation with smaller dimensionality (fewer channels); being it linear, the information loss is limited. Bottlenecks are connected though shortcuts, to improve gradient propagation as shown in ResNet [130]. In a residual architecture, the memory is dominated by the input and output size of the residual blocks, i.e. the size of the layers that have skip connections, which need to be kept in memory for long term. The architecture is particularly memory efficient, since only the small bottleneck layers participate in skip connections, and their size is limited. Squeezenet V2 improves in its predecessors by acknowledging that the number of operations is not sufficient to accurately predict the inference time and introducing hardware-related considerations in the design process, such as minimizing the number of memory access operations; the authors use those considerations to design a network architecture that is significantly faster than Mobilenetv2 on their benchmark GPU and ARM cpu. They do so by giving up group convolutions that are not memory-access efficient and introducing the channel split layer instead, to only process part of the channels in each block; the addition operation at the end of the block is removed and replaced with concatenation of the processed channels with the not processed ones; the number of input and output and output channels for each convolution are kept equal.

MnasNet introduces a search algorithm for identifying the best tradeoff parameters of a given network architecture on a given hardware platform.

MobileNetV3[129] uses the hardware-aware Network Architecture Search (NAS) algorithm described above [119], combining it with design principles taken from novel architecture advances.

A relatively recent survey [131] measures and compares the efficiency of different architectures that are commonly used in terms of inference time, number of operations, number of parameters and accuracy. Their work is a useful reference for choosing a neural network for practical application from the existing architectures.

[58] specifically design a shallower version of the VGG16 architecture [132] for the task of gender recognition, using an ensemble to improve recognition accuracy. Their work shows that the net-

works designed for Image Recognition are a good starting point for transfer-learning to other application domains, but specific efforts for designing a task-specific architecture may be done to push the state of the art further again.

1.3.2.2 Robustness to corruptions and perturbations

Convolutional Neural Networks (CNNs) are very effective in solving computer vision tasks, with deeper and more complex architectures scoring very high accuracy on publicly available benchmarks. However, huge network architectures are sensitive to slight variations of the input data. [133] showed one aspect of this problem in the form of adversarial attacks: images can be slightly modified with tailor-made noise, such that they remain visually identical for a human eye but strongly affect the response of neurons in a neural network.

The robustness problems of neural networks are also revealed when recognition methods are deployed in real scenarios [134]. Variations such as small changes in the framing of the shot, motion blur, focus or the amount of Gaussian noise do not typically affect a human observer but may jeopardize the performance of CNNs [135]. [136] demonstrated that even a moderate blur can severely affect the reliability of object recognition systems, when the architecture is trained on a data set of generally sharp images. [134] explored the instability of the learned representations with respect to distortions such as JPEG compression, image scaling and cropping. [137] showed how image degradations reduce the performance of the trained models. Recently, the AutoAugment strategy for data augmentation was proposed, which consists of a set of policies optimized on the data set at hand [138]. It was demonstrated to improve the robustness of image classification models [139]. On a different line, architectural modifications to existing models were proposed to improve their robustness to corruptions and perturbations. An anti-aliasing filter was deployed before sub-sampling operations by [140], and a new *push-pull* layer was proposed by [141] to learn feature extractors in CNNs that are

intrinsically more robust to noise and corruptions.

Applications like cognitive robotics and intelligent surveillance require to process face images acquired in unconstrained conditions, where many of the mentioned corruptions may occur and affect the performance of the recognition methods. For example, the CMOS sensors deployed in embedded cameras produce noise that can be modeled as the combination of Gaussian noise and shot noise, the former being mainly due to the sensor temperature and the latter being more prominent with high exposure. The limited dynamic range of these sensors produces images with low useful contrast when the scene includes bright and dark parts. In such conditions, the acquired faces look very dark, e.g. when the shot is taken in back-light. Furthermore, memory and bandwidth constraints may require the use of image compression algorithms (e.g. JPEG) that may in turn include artifacts in the images. The limitations of the acquisition devices lead to blurred images due to poor focus and to motion blur artefacts caused by the movement of the subject or the camera itself. The image may present occlusions caused by dirt or water on the camera lens, such as in video surveillance images.

Further challenges come from the variations that occur in image sequences. In real-world applications, recognition methods analyze continuous streams of video data. Each frame usually exhibits coherent content but has slight differences with respect to the previous frame. This is peculiar for face analysis: the appearance of a face can have slight variations (e.g. small pose or expression changes) that, independently of other types of corruptions, represent a challenge and require the learning of robust features to perform consistent analyses. While humans are generally non-sensitive to these variations, neural networks may exhibit instability: from one frame to the next, the subject or the framing may move, the face may slightly change pose, rotating in plane or appearing skewed to the camera. When dealing with videos, one would expect the CNN output to be stable from one frame to the next, while it has been observed that variations such as image shifts severely affect classification performance of current deep

network models [140].

The analysis of slight variations in facial expressions is a challenging problem under controlled conditions, eventually complicated by network robustness issues in real scenarios. The performance of existing emotion recognition methods reported in the literature is often computed on benchmark data sets, with limited or absent consideration of corruptions and perturbations that can occur in the real-world. This type of analysis, although important for the design of improved methods and the progress of the field, does not allow to assess the performance of the developed methods when deployed in practice. We show that the classification error of SOTA methods for emotion recognition easily increases of more than 70% when input data is subjected to corruptions and perturbations. Robustness-by-design is thus required for engineering AI systems to reliably work in their deployment environments.

Chapter 2

Efficient CNN architectures for gender recognition

Based on:
A Convolutional Neural Network for Gender Recognition Optimizing the Accuracy/Speed Tradeoff
A Greco, A Saggese, M Vento, V Vigilante - IEEE Access, 2020

2.1 Background

In this chapter, we aim to design a very efficient CNN architecture for the task of gender recognition.

We first select a known architecture that leverages the latest devices from the state of the art of deep learning; we then show different variants of the chosen architecture to study the effect of the variation on both classification accuracy and prediction latency. To this aim, we choose MobileNets v2 as reference architecture, since it demonstrated remarkable accuracy in image classification, of which gender recognition is clearly a subdomain. The specific application to gender classification, though, gives us the possibility to explicitly rearrange the building blocks in a way that yields the best tradeoff for the problem at hand. In particular, starting from the consideration that the extraction of soft biometrics from faces does not rely on image resolution like the general problem of image classification does, we hypothesize that a reduction of the input size of the network does not significantly affect the accuracy. In addition, since the classification is limited to a single domain, namely the faces, we can reduce the number of feature maps and the number of layers to realize networks that are not so deep, but still achieving excellent performance, comparable to the state of the art, and a better tradeoff with respect to the naive application of the original versions of MobileNets. We find that, as opposed to the general trend in deep learning, a smaller network is able to achieve a notable gender recognition performance without loosing in terms of accuracy.

In addition, since our application requires a neural network that is robust in real world conditions, we train it on a very large dataset that presents significant face variability and we measure

our performance on the well known LFW+ benchmark, which is acquired "in the wild". We compare our network with other methods in the state of the art, to show that the proposed system has comparable or better accuracy but much lower computational demand.

We benchmark our proposed architecture on a low-end embedded device which is widely used for robotic applications. We compare the measured latency and accuracy with the ones of existing optimized architectures, namely Xception [142], Squeezenet [121] and Shufflenet [117]; the experimental evaluation demonstrates the superiority of our solution, which is able to run in real time and to achieve high accuracy in real conditions, with a better trade-off with respect to all the other architectures.

2.2 Methodology

Our proposed feature extracture is based on the design of the multi-purpose neural network family named MobileNets [122][128]. The main reason behind this choice is that the architecture is very suited for applications which require a trade off between accuracy and processing speed on mobile or embedded platforms. Indeed, the authors discovered that a convolutional layer can be split in a "depthwise" operation followed by a "pointwise" operation while still retaining much of the representative power of the network. The combination of the two operation (called depthwise-separable convolution) is functionally equivalent to a 3x3 convolution while requiring 8 to 9 times less operations, with a consequent reduction in the number of parameters and inference time [122]. In [128], the *linear bottleneck* layers are built out of the *separable* ones: when such layers are stacked, a separable convolution is forerun by an additional pointwise layer with linear activation, to form a "bottleneck", where the number of feature maps is increased (expansion) and then decreased (projection): the data are scattered in a higher-dimensional space so that the non-linear power of ReLU activation can be exploited without information loss. In addition,

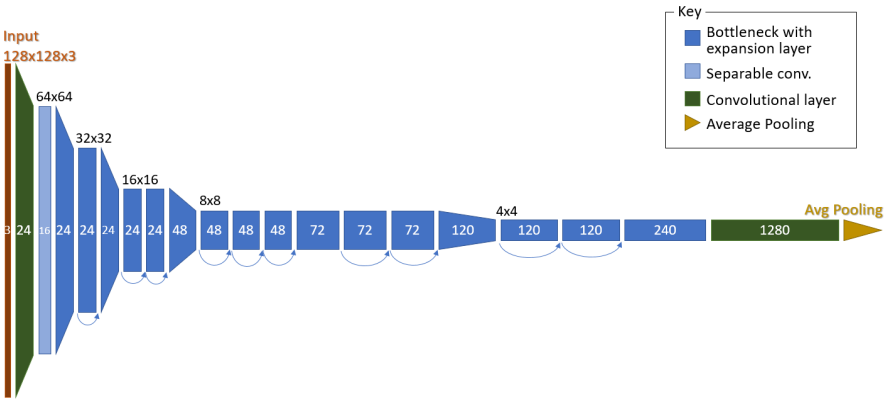

Figure 2.1: The original MobileNets v2 architecture (width multiplier = 0.5, input size = 128).

Table 2.1: Different variables of the architecture experimented in this work.

Change	Experimented values
Input resolution	224x, 160x, 128x, 96x, 64x, 48x, 32x
Width multiplier	1.0, 0.75, 0.5, 0.35
Number of layers	17 (full), 8, 6, 4 blocks

the residual connection from [130] are added to ease backpropagation, but they are also useful to improve the automatic optimization of the computation graph when executed: the presence of skip connections forces a particular order of execution where the memory requirement is dominated by the size of the input and of the output tensors of each residual block (much smaller than the expanded tensors that are treated between the bottlenecks).

2.2.1 Minimization

In this chapter, we experiment different variants of that architecture (depicted in Figure 2.1) to find out how the performance is affected. The variables that we consider, as reported in Table 2.1,

are the input resolution, the width multiplier, namely the ratio of the number of feature maps will be in each convolutional layer with respect to the original network, and the number of layers that compose the architecture.

Starting from the assumption that the gender recognition from faces does not require a huge resolution in most of the cases, the first variant we consider is the input size. Since smaller tensors will save precious memory and improve caching, also requiring less computation, we reduce the input size until we find that further reduction harms the recognition accuracy. We will test various input resolutions (from 224×224 to 32×32) for each width multiplier to find the optimal pair of values. The authors of MobileNet do not use sizes smaller than 96×96 since a smaller size is less convenient when the application concerns object recognition or detection, because the recognition becomes difficult even for human eyes. Since our architecture is tailored on gender recognition, this limit does not apply for us: we can empirically evaluate that 32×32 pixels are enough for a human to distinguish males from females. We show in our experiments that this statement is more or less valid also for neural networks; indeed, a good performance is also achieved with faces of 64×64 pixels.

As for the width multiplier, we will experiment the same values as the original authors of MobileNets, namely 1.0, 0.75, 0.5 and 0.35. Reducing the number of feature maps will strongly reduce the computational load, since the aggregation of the different channels is the most costly operation in an architecture based on separable convolutions [122]. Furthermore, the reduction of the number of feature maps will significantly reduce the memory footprint of the network and the number of parameters.

As a third way to optimize the network we will exploit that, for gender recognition, it has been shown that a very deep network may be overkill; the authors in [58] used a VGG-inspired architecture and showed that very few layers could achieve a very good result. As shown in that work, the gender recognition CNNs do not take advantage using a very deep hierarchy of features, maybe due to the simplicity of the problem with respect to tasks

such as face recognition, age estimation, object detection, where deeper networks generally achieve better performance [31]. Following this intuition, we will experiment how the reduction of the number of layers affects the performance. The rationale is that, starting with a network with minimal input size, width multiplier and number of layers, we will obtain an optimized architecture removing groups of adjacent layers that all have the same number of feature maps (same number of output channels). In Section 2.3.4 we will remove one, two or three groups of layers, showing that the impact on the performance is limited. The resulting architectures are described in Table 2.2.

2.2.2 Training

All the network architectures are trained from scratch. The Xavier Uniform method [143] is used for parameter initialization, which has been proven to allow neural networks to achieve quick convergence and high accuracy in several computer vision tasks; we did not use experiment different initialization methods, since this experimental analysis is focused on the efficiency of the method. For the same reason, we set the batch size to 64 and perform 100 epochs of 400,000 samples each.

We use data augmentation to improve the training effectiveness: each loaded image is randomly modified in one or more of the following ways:

1. Random crop, to model the effects of imprecise unaligned face detection
2. Horizontal flip
3. Image resampling, to simulate low resolution
4. Brightness change
5. Addition of gaussian noise, to simulate noisy images

The learning rate is initially set to 0.005 and it is halved every 20 epochs. The Adam optimizer is used with parameters b1=0.9,

Table 2.2: Reduction of the depth in successive steps. The leftmost column shows the number of feature maps ("width") for each residual block in the original network; m represents the width multiplier. Successive reductions collapse adjacent blocks with the same "width", starting from 17 of the original neural network architecture.

Original (17)	Half net (8)	Smaller (6)	Smallest (4)
16*m	16*m	16*m	16*m
24*m	24*m	24*m	24*m
24*m	32*m	32*m	32*m
32*m	32*m	32*m	64*m
32*m	64*m	64*m	1280
32*m	64*m	64*m	avg
64*m	64*m	1280	2
64*m	64*m	avg	
64*m	1280	2	
64*m	avg		
96*m	2		
96*m			
96*m			
160*m			
160*m			
160*m			
320*m			
1280			
avg			
2			

b2=0.999, decay=5e-5. Inspiring to related literature, we inserted a dropout layer between the last convolutional layer and the last fully connected layer. The dropout rate is set to 0.2.

2.2.3 Preprocessing

As described in Section 1.3.1, a method is composed of different steps. In this chapter we focus on improving the efficiency of the combined implementation of steps 3 and 4, feature extraction and classification; however in this paragraph we describe the rest of the method used in our experimental setup since, as previously described, preprocessing steps affect the overall result in a significant manner.

As for the detection step, we adopt the well-known Viola Jones face detector [35], which is quite reliable when applied to frontal faces but it is still very fast when compared to modern alternatives. We do not use any face alignment in our pipeline; the main reason is two-fold: the face detector is only trained on frontal faces, so the variability seen by the feature extractor is already limited; furthermore we aim to design a very efficient system and so we assume not to have enough computational power to run an accurate feature extractor. Since the downsides outweight the benefits we decided to drop the normalization step altogether and to only rely on the discriminant power of the neural network to deal with all the variations.

The detected face is cropped and then resampled with bilinear interpolation to match the input size of the network. This resampling method has been preferred to Nearest-Neighbour because NN produces significant artifacts on the images with lower resolution and negatively affects accuracy; more complex methods such as Bicubic resampling produce similar results in the spatial and frequency domain but with increased complexity.

2.3 Results

We perform a comprehensive experimental analysis on several public datasets; we describe them in Section 2.3.1, while in Section 2.3.2 we give details about our experimental procedure, to make it reproducible. Then we report the results of all our experiments in the following Subsections. In Subsection 2.3.3 we describe, at various input resolutions, the effect of decreasing the number of feature maps; in Subsection 2.3.4 we evaluate how the reduction of the number of layers affects the performance and we show how the accuracy is traded with speed in the proposed variants of the basic architecture. In Subsection 2.3.5 we compare our proposed solution with other architectures on the considered datasets. Finally, in Subsection 2.3.6 we analyze the results in real environments and show how our approach is able to succeed in the target applications while different solutions fail.

2.3.1 Datasets

In this section we are going to introduce the datasets used in our experiments.

2.3.1.1 VGGFace

The VGGFace dataset [144] was built to train Deep Neural Networks on the problem of face recognition, where no existing public dataset were large enough to effectively train DNNs. The dataset is gathered in an inexpensive way, using services such as Google Search to obtain a huge quantity of weakly annotated images. Such images were then filtered and the annotations fixed and verified manually through a fast inexact process to achieve a certain dataset purity, less than 100% but vastly sufficient to be used for training purposes.

The second version of the VGGFace [32], namely VGGFace2, was gathered in a similar way but contains a larger quantity of subjects (9,131), images (3.31 millions) and variations in pose, age, illumination, ethnicity and context. This dataset was originally

gathered for face recognition, but it is also annotated with gender, so it is suitable for our aim. The dataset is already partitioned in training and test set. From the training set we extracted 2 millions of images for training and we kept 200.000 more images for validation. The partition was performed on a subject-independent basis, i.e. no subject identities in the training set are in the validation set. The validation set is perfectly balanced (100.000 males and 100.000 females) while the training set is slightly unbalanced (57% males, 43% females). The test set was used as it is for testing, as intended.

2.3.1.2 LFW dataset

The LFW dataset [145] is the most popular benchmark for gender recognition, even though it was originally created for unconstrained face recognition. It contains 13,233 images of 5749 unique subjects, with a significant imbalance between males (77%) and females (23%). Since LFW is a standard for gender recognition, we have used it as reference for our experimental analysis; for a fair performance comparison, we used the same test set proposed in [51], [58] and [57].

2.3.1.3 MIVIA-Gender dataset

The MIVIA-Gender dataset [146] has been acquired in real scenarios and it is particularly suited for evaluating the performance in unconstrained environments. In fact, it contains face images captured in extreme lighting conditions, with motion blur, different poses and expressions, low resolution and low quality. The dataset is composed by almost 6,000 face images and it is partitioned in three subsets, namely UNISA-1, that is acquired in more controlled situations, UNISA-2 and SM, that are very challenging and have been acquired in different scenarios. We used this dataset for testing the capabilities of the CNNs to generalize in real environments.

2.3.1.4 IMDB-WIKI dataset

The IMDB-WIKI dataset [147] consists of images of celebrities collected from the famous IMDB website and from Wikipedia. The total number of images of the two partitions, namely IMDB and WIKI, is $523,051$. The faces are automatically annotated with gender and age labels, but the authors themselves declare that they can not vouch for the accuracy of the annotations. In fact, they assume that all the images with a single face belong to the celebrity and automatically annotate them with the gender declared in the profile; this assumption results in several errors in the IMDB partition. Consequently, it is recommended to use the WIKI partition, that is more accurate, for testing purposes; in spite of this, we used both the partitions for our experimental analysis, in order to increase the size of the test set.

2.3.1.5 Adience dataset

The Adience dataset [148] consists of 26,580 images of 2,284 different subjects. It is commonly used for gender recognition and age group classification. It has an extreme variety in terms of age, including a large quantity of children and includes a lot of images with very low quality and resolution. Therefore, it is a good dataset for testing the gender recognition capabilities in very challenging conditions.

2.3.2 Experimental protocol

All the architectures were trained with Tensorflow and Keras on a Titan Xp GPU. The latency is measured on a CPU-only setup, without any GPU acceleration and on batches of size 1. The reported latency is computed as an average of 100 executions, where the neural network is loaded once and 100 different batches of 1 image each are fed into it consecutively. The measured time does not include the time for loading/acquiring the image nor the time for finding the face into the image (i.e. detection).

Specifically, we used an embedded platform for testing, namely an ARM Cortex A53 (ARMv8) clocked at 1.2GHz, on board of a Raspberry Pi 3 Model B, with 1GB ram. The setup is meant to simulate real use conditions in absence of dedicated hardware, that is still a common case nowadays. Many mid-high end embedded devices such as smart cameras use ARMv7 or ARMv8 chips, where Cortex-A7 and Cortex-A53 are common choices and achieve similar performance.

In the first evaluation on the LFW dataset we include two comparable results from the state of the art: the first (hereinafter *SoA Fast*) is the network ensemble presented in [58], specifically designed to be lightweight and fast; the second is at the best of our knowledge the most accurate architecture on the target dataset available in the literature [57] (hereinafter *SoA Best*). The experiments in these two papers are performed on the same set of data, the LFW test set, with the same experimental protocol: all the evaluation is performed in a cross-dataset fashion, without fine tuning on the target dataset. Such experimental protocol allows to obtain a more reliable, pessimistic, estimate of the network generalization capabilities when the system is deployed in real scenarios, that is one of our purposes. Furthermore, we also considered for comparison purposes other networks widely used in other image classification tasks: Xception, Shufflenet and Squeezenet.

According to the same rationale, we perform a more extensive evaluation on all the considered datasets by using the same cross-dataset evaluation on all the considered datasets, namely the VGGFace2 test set, LFW, MIVIA-Gender, IMDB-WIKI and Adience.

2.3.3 Input size and number of feature maps

In the first experiment we evaluate the performance of the proposed method on the LFW dataset by varying both the input size and the width multiplier, namely the fraction of the original feature maps. The results are shown in Figure 2.2. For this evaluation, we will adopt the notation x_y, where x is the input size

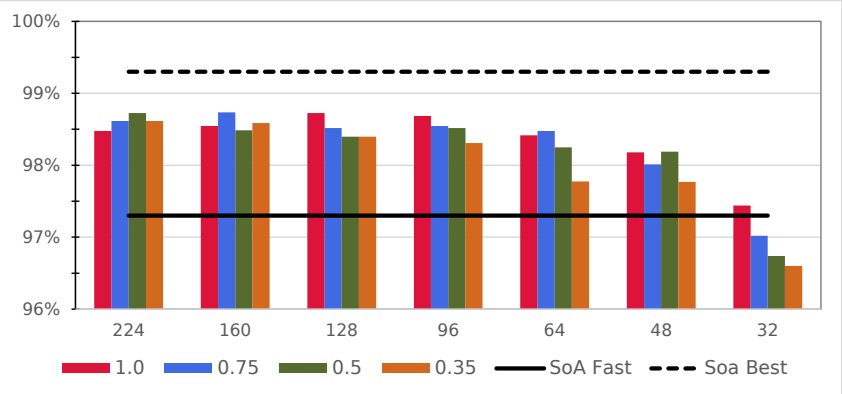

Figure 2.2: Classification accuracy vs.input size (224, 160, ...) and width multiplier (1.0, 0.75, ...) on the LFW dataset. On the chart we also display two main results of the state of the art for comparison, namely SoA Fast [58] and SoA Best [57]. More details are reported in Section 2.3.3.

and y is the width multiplier. The original MobileNet v2 network architecture is marked with the label 224_1.0; this is the largest, most complex model that we experiment and compare with the optimized versions. The most noteworthy consideration is the fact that the original version does not obtain the best performance. Indeed, the best accuracy of 98.73% is achieved with the network 160_0.75. This difference may be interpreted as an effect of overfitting or by considering that the average size of the face images available in the VGGFace2 is significantly smaller than 224×224. In any case, the performance is quite stable with respect to the input size and a bit more sensitive according to the width multiplier, with a reduction of the performance when this parameter is set to 32. However, even in this case the performance are never before 96.5%, while being more stable in the other cases in the range $97.7\% - 98.6\%$.

We also notice that somehow a larger input size can compensate for a lower width multiplier and viceversa: the architectures 128_1.0, 160_0.75 and 224_0.5 achieve almost the same accuracy. It means that the variability of the results among different versions

is mainly due to the quantity of parameters and so to the general representative power of the network rather than to one specific variation of the architecture.

The performance is significantly reduced when the input size drops below 64×64. This may be due to the fact that, even if 32×32 is typically enough for a human to distinguish gender, the proposed network architecture applies a double strided convolution in the first hidden layers, and much information are discarded from the 32×32 image starting from the second layer.

2.3.4 Network depth

In this second experiment we verify how and whether the reduction of the number of layers affects the performance. We choose two configurations for the input size and the width multiplier and use those parameter to train optimized architectures. We use 96_0.75 and 64_0.5 that are two mid-low sized configurations that still yield a good accuracy, and 160_0.75 that is a bigger configuration that achieves our best result on this dataset, as shown in the previous Subsection.

In Figure 2.3 we compare the full-size network (17 residual blocks) with some reduced versions (8, 6 and 4 blocks). Many aspects emerge from these results. We can see that even if the depth of the network is severely reduced along with the latency, the classification accuracy is pretty consistent. In particular, we clearly see that it is much more convenient to reduce the depth of 96_0.75 to 8 or even to 6 instead of moving to the 64_0.5 configuration. With respect to the 160_0.75 architecture, it is clear that a great performance drop occurs reducing the depth. A cause is probably the overfitting: too many parameters have to be learned, but the structure of the network is too shallow to construct an adequate feature hierarchy, so the performance is noticeably affected with respect to equivalent architectures with less parameters (i.e. 96_0.75 and 64_0.5). The adoption of dropout, as described in Section 2.2.2, is not sufficient to avoid that. Another cause may be the fact that, having a larger input resolution, the last convo-

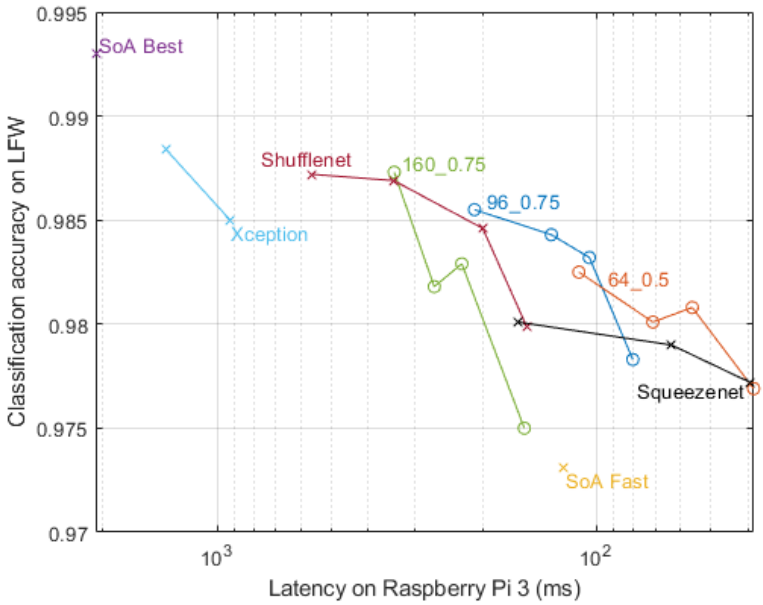

Figure 2.3: Scatter plot of latency versus accuracy on the LFW dataset. For our proposed architectures (circles), each line represents a different combination between input size and width multiplier and every point indicates a different number of blocks. The other points (crosses) represent variants of different architectures we compare with.

lutional layer produces larger feature maps, that are less suited for gender classification with respect to the smaller ones, where the information is condensed. Finally, we observe that difference between shallow and deep network is less pronouced with smaller resolutions (i.e. 64_0.5). With such a small resolution, the full size network would have very small feature maps as output of the last convolutional layer (up to 1×1 if the input is 32×32), while shallower networks alleviate this problem, providing the fully connected layer with enough spatial granularity.

Table 2.3: Evaluation of different architectures on different datasets. The table reports the processing time on the target embedded platform as well as the accuracy on each dataset.

	Latency	Accuracy (%)							
Model	**(ms)**	**LFW**	**VGG val.**	**VGG test**	**UNISA-1**	**UNISA-2+SM**	**IMDB**	**WIKI**	**Adience**
xception-71	623	98.50	**97.80**	96.17	**97.92**	93.25	80.17	94.97	83.66
xception-150	1363	**98.84**	97.70	**97.02**	**97.92**	**94.72**	**80.76**	95.90	**84.49**
shufflenet-0.5-64	153	97.99	96.52	96.27	93.75	91.11	80.21	94.57	83.14
shufflenet-0.5-112	199	98.46	97.00	96.69	97.66	93.25	80.61	95.44	83.95
shufflenet-0.5-224	342	98.69	97.32	96.84	96.88	94.46	80.64	95.84	84.22
shufflenet-1-224	561	98.72	97.33	96.94	96.35	94.36	80.74	**95.97**	84.27
squeezenet-224	161	98.05	96.52	96.48	95.57	90.48	80.41	94.91	82.83
squeezenet-112	63	97.89	96.30	95.91	95.83	90.16	80.20	94.62	81.80
squeezenet-64	39	97.72	95.84	95.67	94.27	88.76	79.98	94.03	81.59
proposed 64_0.5_4	**38**	97.69	95.88	95.48	91.93	86.38	79.97	93.93	81.61
proposed 64_0.5_6	56	98.08	96.46	96.10	92.97	91.41	80.29	94.79	82.68
proposed 64_0.5_8	71	98.01	96.69	96.34	94.53	92.54	80.35	95.01	82.91
proposed 64_0.5_17	113	98.25	96.70	96.30	96.61	92.06	80.41	94.94	83.23
proposed 96_0.75_4	80	97.83	96.28	95.77	93.49	86.74	80.24	94.44	82.41
proposed 96_0.75_6	104	98.32	96.69	96.45	95.31	91.18	80.51	95.12	83.31
proposed 96_0.75_8	131	98.43	96.94	96.59	97.40	92.24	80.58	95.46	83.56
proposed 96_0.75_17	209	98.55	97.15	96.73	97.40	93.04	80.66	95.67	84.48
proposed 160_0.75_4	115	97.50	95.72	95.30	87.76	82.96	80.11	94.07	81.41
proposed 160_0.75_6	226	98.29	96.81	96.35	95.83	91.00	80.51	95.34	83.12
proposed 160_0.75_8	267	98.18	96.93	96.53	95.05	92.73	80.63	95.40	83.41
proposed 160_0.75_17	341	98.73	97.18	96.86	96.35	93.58	80.74	95.78	84.45

2.3.5 Comparison with other architectures

In this section we compare our proposed solutions with other architectures on all the considered datasets. Hereinafter, we will use the notation x_y_z, where x and y are still the input size and the width multiplier, while z is the number of blocks.

In addition to *SoA Fast* [58] and *SoA Best* [57], whose results are available only for the LFW dataset, we include three more architectures that have been proven effective and efficient for the generic task of object recognition training them for gender recognition. In particular we experiment the architecture named Xception [142] that improves over the popular Inception architecture using depthwise convolution, like in our proposed architecture. Then, we experiment the Squeezenet architecture [121], that is thought for embedded systems, even though it does not directly optimize the processing speed with respect to the classification accuracy. Finally we experiment ShufflenetV2 [117], that is a very efficient architecture optimized with special reference to the hardware that we target to obtain the best results with the minimum possible processing time. For each of the considered networks we considered different input sizes that make sense to the specific architecture and are comparable to our proposed network. Since Shufflenet comes in two different versions, with full feature maps (ShufflenetV2-1) and half feature maps (ShufflenetV2-.5), we experiment both the variants.

Looking at Figure 2.3 we can note that the accuracy achieved by the smallest proposed network, namely 64_0.5_4, is still higher than the one reached by *SoA Fast* (97.69% vs 97.31%), even achieving lower latency (38 ms vs 122 ms). Compared to *SoA Best* [57], the proposed architecture yields an arguably similar accuracy (only 0.57% lower) but it is significantly faster, since all our proposed architectures require between 40 ms and 340 ms while *SoA Best* is more than 6 times slower). It is also worth pointing out the differences in the training procedure with respect to the one applied in *SoA Best* [57], in order to explain the performance gap on the LFW dataset. In our case no pretraining is

performed, while the authors of [57] prove that a face recognition pretraining significantly improves classification of accuracy of the final model. Then, we use VGGFace2 as training dataset, while [57] used the IMDB-WIKI cleaned. Our training dataset is bigger (2 million images versus 250.000) and and this is an advantage, but the IMDB-Wiki dataset contains 50% of the identities contained in the LFW test set. Finally, we use a different type of data augmentation and a different optimisation algorithm, that we think is more suitable for our architecture as explained in Section 2.2.2. The difference is confirmed by the fact that when we train the architectures from [57] on the VGGFace2 dataset, we obtain 98.75% performance, even with face recognition pretraining, that is lower than the one that the original authors obtain (99.30%). We think that the 0.5% difference is due to the identity overlap: in the hardest cases, for people whose face does not express their gender in a clear way, estimating gender is easier when the classifier has already seen samples for the same person.

As for the other architectures, from the results reported in Table 2.3, we can note that Xception obtains the best performance, but it is significantly slower than the others; it requires too much processing time (1363 ms), so it is not suited for our purposes. The second best is Shufflenet, but the accuracy significantly decreases when we reduce its input size. With the same input size, our proposed version 64_0.5_8, for example, is 50% faster with comparable or better accuracy (between 0.05% and 1.50% of improvement on the considered datasets). Larger versions of the architecture take much more time to process with respect to our proposed equivalent. The performance of Squeezenet is lower than the other networks when the full input size is used, but reducing this parameter the architecture retains most of its accuracy greatly improving the processing speed. However, fixed the processing time, our network achieves a comparable (64_0.5_4 vs squeezenet-64) or higher (64_0.5_6 or 64_0.5_8 vs squeezenet-112) accuracy than Squeezenet.

The experimental results demonstrate that crafting a specially tailored network is worthwhile to obtain the best efficiency in a

specific problem such as gender recognition. In fact, our proposed architecture was explicitly tailored for gender recognition in terms of input size, number of feature maps and number of layers, while the other architectures are designed with reference to object classification. Such task based optimization allows to find the best trade-off between accuracy and processing time and to achieve our goals.

Another trend that we can note analyzing the results reported in Table 2.3, is that the relative accuracy is consistent among different datasets, i.e. the architectures that perform better on the reference LFW benchmark, still perform better than others on all the considered datasets. As expected, we can observe a fluctuation of the performance on the different datasets, according to their intrinsic challenges: the results on LFW, VGG-Face DS 2 and WIKI are typically higher, while UNISA-2+SM is lower and Adience is the lowest together with IMDB. In fact, UNISA-2 and SM are very challenging partitions of the MIVIA-Gender dataset, acquired from surveillance cameras with extreme lighting conditions, face poses and low quality and resolution. Adience is mainly used for age estimation and contains a huge number of newborns, infants and toddler, where even human performance is near-random trying to guess gender from the face. IMDB dataset notoriously includes very noisy annotation of identity, due to the presence of images with multiple people in them, so it is not commonly used as a benchmark for evaluation, but more often for training. In all the cases, our proposed architecture is always able to achieve very high accuracy, even requiring significantly less processing time.

2.3.6 Practical considerations

To confirm that our proposed models can be effectively used in real environments we can do some additional measures to estimate the time constraints more precisely. Cascade detection algorithms such as the one from Viola and Jones that we adopt, have different running times depending on how much face-like configurations are seen in the frame. We measure that on the target platform, the

detection algorithm will take less than 100 ms to run in typical worst case conditions (where many faces are present). We consider a reasonable worst case of 3 faces per frame, and we consider acceptable the whole system to run at 3 fps. This processing speed is to be considered perfectly acceptable for applications such as digital signage, automatized social interaction and statistics.

With those constraints a time of 70ms or lower is acceptable. We can use our optimized models for the target application, for example 64_0.5_8, since an accuracy of about 98% can be considered enough in the wild for the target applications. The accuracy can also be slightly improved through ensembling classification on successive frames. Squeezenet also makes a suitable architecture for such an application, but only if we use a reduced input size. *SoA Best* would not be able to run in real time on the considered platform, having a time of 2 seconds per face that would be unacceptable for those applications requiring a strict real time; the same considerations can be done for Xception and Shufflenet. Furthermore *SoA Best* (which uses ResNet-50) and Xception, have to rely on 1GB additional swap space on flash memory, since they do not fit in the available RAM.

To finally assess that the 98% accuracy is reasonable for our model, in Figure 2.4 we show some of the samples for which our system gets an error. They are mainly due to non-evident gender features on the face, or to the variability in gender and ethnicity: since the training dataset is not balanced with respect to them, we expect that the accuracy drop classifying children, elders and Asians, since most people in the training set are caucasian adults. This shows that the network, even in its simplified more efficient form, successfully learned how to classify gender from faces.

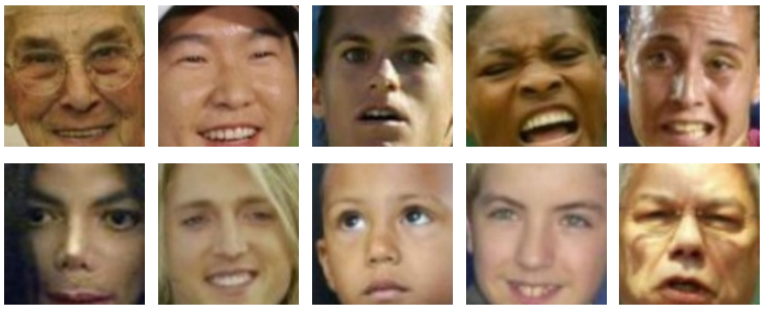

Figure 2.4: Samples of misclassifications on the LFW test set. Faces in the first row were miclassified as males, while the ones in the second row were mistaken for females. Most of the few errors concern children, elders, Asians and objectively difficult samples.

Chapter 3

Deep networks for ethnicity recognition

Based on:
Benchmarking deep network architectures for ethnicity recognition using a new large face dataset
A Greco, G Percannella, M Vento, V Vigilante - Machine Vision and Applications, 2020

3.1 Background

In this chapter we aim to develop a reliable neural network for ethnicity recognition. As discussed in Section 1.3.1.3, we believe that the key to developing such a reliable network does not lie in specific architectural features, bur rather in the use of a suitable dataset, which appears to not exist in literature yet.

For that reason, we used an efficient procedure to annotate more than 3 millions images, leveraging the 9,129 identities available in the publicly available VGGFace2 dataset [32]. The annotation distinguishes 4 ethnicity groups, namely African American (AA), East Asian (EA), Caucasian Latin (CL) and Asian Indian (AI). To avoid the bias possibly introduced by the other race effect, the annotation is done by three people belonging to different ethnicities, one African American, one Caucasian Latin and one Asian Indian, choosing the final ethnicity group through a majority voting. The opinion of a fourth annotator has been required in case of a tie.

In the following of this chapter, we use this dataset for training modern deep network architectures, such as ResNet-50, VGG-16, VGG-Face and the efficient architecture MobileNet v2 which we used in the previous chapter, obtaining more than 94% of accuracy on the test set. In addition, following on the experiments carried out in [73], we perform a cross dataset evaluation demonstrating that neural networks (ResNet-34 and VGG-Face) trained with VMER are able to better generalize on a different test set (UTK-Face) with respect to the same networks trained with FairFace, so confirming the effectiveness of the new set of labels. Finally, we visualize the features learned by a CNN trained with VMER to

demonstrate how effectively they encode distinctive facial traits, recognizable by humans.

We find that using this dataset for training convolutional neural networks allows to obtain better accuracy that existing results in the state of the art; we consider our results as a baseline of the performance achievable with the modern deep network architectures and assume that this contribution can pave the way for future experiments and applications in this research field.

For this reason, we make the whole benchmark publicly available with the name VGGFace2 Mivia Ethnicity Recognition (VMER) dataset. The utility for the scientific research is at least threefold: i) the high number of samples available in the dataset allows the training of CNN networks, including larger architectures, that are more prone to overfitting; ii) the annotation protocol, designed for reducing the own race bias, ensures the accuracy of the ethnicity labels; iii) the availability of other annotations for the same images, namely identity and gender, makes the dataset particularly suited for future advanced analyses, such as multi-task learning or forensics applications.

The chapter is organized as follows: in Section 3.2 we describe the dataset, giving details about the available face images and the characteristics of the considered ethnicity categories; in Section 3.3 we describe the experimental setup, including all the details to make our experimentation reproducible; in section 3.4 we report and comment the results of our experimental analysis;

3.2 Dataset

3.2.1 Description

The proposed VMER dataset is composed by images collected from the original VGGFace2 [32], which is so far the largest face dataset in the world including more than 3.3 millions face images, with an average of about 362 samples per subject (minimum 87 images per subject). It also includes gender labels and consists of 62% males and 38% females.

Figure 3.1: Samples of African American, East Asian, Caucasian Latin and Asian Indian people available in the VMER dataset.

Table 3.1: Number of images and subjects for each ethnicity available in the VMER dataset, divided in training and test set.

Ethnicity	Number of samples (subjects)		Percentage of samples	
	Training	**Test**	**Training**	**Test**
African American	242,783 (712)	10,373 (34)	7.7	6.1
East Asian	187,893 (533)	18,750 (62)	6.0	11.1
Caucasian Latin	2,507,837 (6854)	130,900 (380)	79.9	77.4
Asian Indian	202,205 (530)	9,001 (24)	6.4	5.3
Total	3,140,718 (8,629)	169,024 (500)	3,309,742 (9,129)	

The images in the dataset have been acquired in different lighting and occlusion conditions and the faces of the subjects are characterized by different pose, age, ethnicity and size. In particular, more than 75% of the faces have a resolution between 50×50 and 200×200 pixels, which is less than the input size of most of the popular CNNs. It is important to take into account this aspect when dealing with these face images, as we will show in our experimental analysis.

3.2.2 Ethnicity annotation

The categorization of the ethnicity is a task anything but simple even for a human, as witnessed by the scientific literature in this field [74]; imagine how complex this classification can be for a computer vision algorithm, which can only make use of color, texture, morphological features and craniofacial measurements that can be automatically extracted from a face image. As extensively discussed in Section 1.3.1.3, the ethnicity annotation requires a manual procedure that takes into account the somatic facial features which a human uses to distinguish the ethnicity categories.

According to the most recent trends, we choose to divide our dataset into the following four categories:

- African American (AA): the individuals of this ethnicity group typically have African, North American or South American origins and are characterized by dark skin color, full lips and wide nose.

- East Asian (EA): people belonging to this group have Chinese or other East and South East Asian origins. Their color skin is light, with shades from white to yellowish, and small nose, but the most distinctive feature is the almond shape of the eyes and the inclination between the medial and the lateral canthus, which make the eye look narrower.

- Caucasian Latin (CL): humans of such ethnicity have European, South American, Western Asian and North African

origins and are characterized by a white or tanned skin, medium nose and lips and horizontally aligned eyes.

- Asian Indian (AI): folk belonging to this ethnicity group have Indian, South Asian or Pacific Island origins. They have characteristics in common with EAs and CLs, but we can distinguish them by noting very slight differences. They have a slightly darker skin color and eyes with more defined features with respect to EAs and CLs.

Examples of face images belonging to the four ethnicity categories are depicted in Figure 1.

In order to avoid the other race effect, we asked three people belonging to different ethnicities, namely one African American, one Caucasian Latin and one Asian Indian, to annotate each identity with an ethnicity label among the considered four.

The results of the annotations fully confirm the importance of consulting multiple annotators. In fact, the three people fully agreed on only 85% of the dataset (7,779 identities); in 14% of the cases (1,278 identities), only two of the annotators gave unanimous labels, while in the remaining 1% (74 identities) they all provided conflicting opinions. The inter-rater agreement, computed with the Cohen's Kappa [149], is equal to 0.74 and confirms our hypotheses. This value confirms a good agreement between the annotators, but it also shows the necessity of averaging the annotations in order to avoid the other race effect.

To obtain the final annotations we applied a majority voting rule, which allowed to determine the ethnicity label for 99% of the face images in the dataset; as for the remaining 1%, we employed a tie-break rule, by asking a fourth annotator the opinion about the ethnicity. Such annotator, unlike the others, was allowed to gather information about the identities (known the name and surname of the celebrity, it was possible to determine the birth place and so on) and the opinions of the other annotators, in order to take more into consideration the opinion of the person of the same ethnicity group, according to the ORE concept; despite this apparent advantage, the role of the latter was anything but simple,

because the remaining 74 identities had characteristics common to different ethnicity groups, so confirming the difficulty of this task even for a human.

3.2.3 Dataset statistics

The face images have been then divided in training and test set, by preserving the identity partition provided by the original VGGFace2 authors. The training and the test set are already splitted and the ethnicity labels are available upon request at https://mivia.unisa.it. The downloadable package also contains the different annotation files produced by the three annotators.

The final VMER dataset, whose detailed statistics are reported in Table 3.1, consists of 3,309,742 face images of 9,129 identities. There is no subject overlap between the training and the test sets, namely the samples of the subjects used for training the networks are not included in the test set. In face analysis, this separation is very important for evaluating the generalization capabilities of the neural networks.

The training and the test set are unbalanced, since around 80% of the images belong to the Caucasian Latin category; this is not representative of the real distribution of ethnicities in the world. Nevertheless we argue that the available samples are sufficient for obtaining a wide training set in which all the ethnicity categories are equally represented. The less represented class in the training set, namely the East Asian, includes $187,893$ samples; if we randomly select $187,893$ samples from each of the 4 classes, it is possible to obtain a balanced training set with more than 750.000 face images, that is by far the largest existing balanced training set for ethnicity recognition. In 3.4 we demostrate how this procedure allows the convolutional neural networks to learn a set of features not specialized on the most represented ethnicities.

In the following of the chapter, we perform different experiments with the original training and the balanced one in order to evaluate the impact of this aspect on the overall performance.

3.3 Experimental setup

In this section we describe the considered CNNs and the protocol adopted for our experimental analysis. We analyze the results from different points of view, evaluating the effect of the data augmentation, the possibility to balance the per-class error, the impact of the input size, the generalization capability and the learned features.

3.3.1 Deep network architectures

For our experimental analysis we have chosen the CNNs that we consider the most promising and interesting among the modern deep network architectures, namely VGG-16, VGG-Face, ResNet-50 and MobileNet v2.

VGG-16 [132] is one of the most experimented CNN architectures for facial soft biometrics analysis. It achieved a significant success thanks to its shallow architecture (around 130K parameters, 13 convolutional layers and 3 fully connected layers), that allows to better generalize even in presence of small training sets. It achieves state of the art age estimation accuracy [150] and it is not a case, since there are not very large datasets for training deep networks for age estimation. Being one of the most popular CNNs, we include it in our performance analysis. As evident from the name, it consists of 16 layers. The typical input size used is 224×224 pixels.

VGG-Face [144] is VGG-16 trained from scratch for face recognition on almost 1,000,000 images. This CNN is probably the most adopted architecture for facial soft biometrics analysis. Indeed, the availability of weights pre-trained on a very large number of face images and not for general image classification task (ImageNet) makes it very suited for transfer learning. VGG-Face achieved an impressive accuracy in face recognition [144] and age estimation [150] and it has been successfully experimented also for gender recognition [59]; for this reason, we believe it can be effective also for ethnicity recognition purposes.

ResNet-50 is one of the residual networks [130] proposed by the Microsoft research group, which won the ILSVRC and COCO 2015 competitions. The most important feature of such architecture is the introduction of the residual blocks, which allow ResNet to require less processing time for training and less extra parameters for increasing the depth of the network. Consequently, various versions of ResNet have been proposed with increasing number of layers (18, 34, 50, 101, 152). However, it has been demonstrated that ResNet-50 is very effective for other facial soft biometrics analysis, namely age group classification [150] and emotion recognition [151]. For this reason, in this paper we use its version with 50 layers, which takes as input an image of 224×224 pixels.

MobileNet v2 is one of the MobileNets architectures [122], very suited for mobile and embedded vision applications; think, as an example, to a cognitive robot or a smart camera that is able to perform face analysis in real-time with a good accuracy even using normal CPUs [59]. The software optimization which allows this network to significantly reduce the processing time is the transformation of the convolutional layers in depthwise and pointwise operations, without paying a lot in terms of accuracy. In our opinion, this network architecture can be useful for real-time ethnicity recognition applications running on low cost devices. In this paper, we use the most popular v2 version, which consists of 17 layers and that, in its original version trained with ImageNet, requires an input of 224×224 pixels.

3.3.2 Experimental protocol

For each considered CNN we apply the same experimental protocol. First of all, we start from the models and the weights already available for all the networks; in particular, we use the implementations of the CNNs already available in Keras with Tensorflow backend. Then, we train them by starting from the pre-trained weights, performing a fine tuning of all the layers.

We use the Adam optimizer and start from a learning rate equal to 0.0005, applying a learning rate decay of 0.5 every 6

epochs; moreover, we setup a weight decay equal to 5e-5. We impose a batch size equal to 64 and build the batch in order to preserve the a priori distribution of the training set. In more details, since the training set (see Table 3.1) includes 7.7% of African American, 6.0% of East Asian, 79.9% of Caucasian Latin and 6.4% of Asian Indian face images, we build each batch with 5 African American, 4 East Asian, 51 Caucasian Latin and 4 Asian Indian faces.

We fix the maximum number of training epochs to 20 and implement an early stopping mechanism: if the accuracy on the validation set does not improve for 3 consecutive epochs, the training is stopped.

3.4 Results

The results achieved by the considered convolutional neural networks, trained with the above mentioned protocol, are reported in Figure 3.2. As noticed in other face analysis tasks [150], VGG-Face is the most effective CNN also on the VMER dataset, obtaining 94.1% of accuracy. However, the gap with the other networks is not so wide, being all able to achieve an accuracy greater than 93% (MobileNet v2 94.0%, VGG-16 93.7%, ResNet-50 93.1%).

Such results suggest that the proposed dataset allows to effectively train CNN architectures for ethnicity recognition. However, it is worth to deepen the analysis by applying data augmentation or specific design choices for balancing the errors and optimizing the processing time, in order to evaluate the impact of these factors.

3.4.1 Effect of data augmentation

Data augmentation on the training set is a strategy which demonstrated to be very effective for improving the generalization capabilities of the neural networks; this is definitely true for face analysis, since the possible face variations in terms of pose, orientation, resolution, image quality and occlusions, require the adoption of

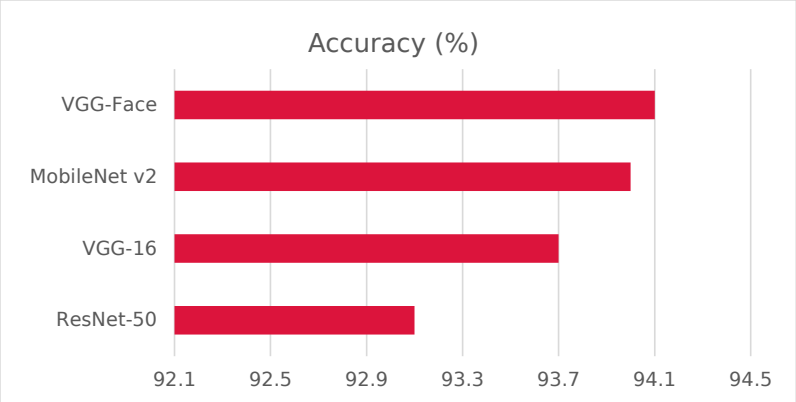

Figure 3.2: Ethnicity recognition accuracy (%) of the considered CNNs on the VMER dataset. In this experiment, the CNNs are trained without data augmentation.

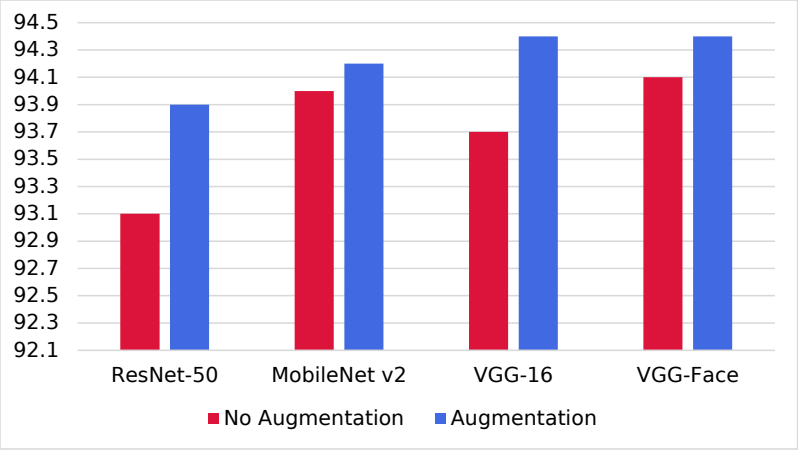

Figure 3.3: Ethnicity recognition accuracy (%) of the considered CNNs without and with data augmentation on the VMER dataset. The positive effect of the data augmentation is evident for all the CNNs.

techniques for making the training set more representative of the real facial variability.

Therefore, we performed a new training of the considered CNNs by applying data augmentation. To make a fair comparison, we have not increased the number of training samples, but we defined a pseudo-random procedure to synthetically add variations to the available face images. In particular, we randomly applied the following data augmentation techniques: gaussian noise addition, brightness change, image rescaling and random flip. It is worth mentioning that these operations are not mutually exclusive, since we randomly combined also 2 or more augmentation strategies.

To reproduce the effect of motion blur or low image quality, we add gaussian noise, produced with a zero-mean gaussian distribution, by fixing $sigma = 12$. To complete the transformation, we normalize the image to have values between 0 and 255.

To simulate overexposure and underexposure, which can be present depending on how the camera is installed with respect to the light source, we randomly add or subtract brightness to the original image. In particular, we subtract 30% of the pixel intensity values to reduce the brightness of the original image, while doing the opposite to reproduce the overexposure. Also in this case, we finally normalize the image to have values in the range [0, 255].

In real environments the distance between the face and the camera is always variable; if the person is far from the camera, the resulting face image can have a very low resolution. To reproduce this effect, we randomly subsample the original image by resizing it with a random scaling factor of 2 or 4. Of course, this transformation is applied before rescaling the face image to 224×224 pixels, namely the size required by the target CNNs.

Finally, we further augment the dataset by randomly flipping the original image in the horizontal direction.

The results of this experiment, shown in Figure 3.3, demonstrate the effectiveness of the data augmentation, since all the CNNs benefit from the application of this strategy. VGG-Face and

Table 3.2: Per-class and overall ethnicity recognition accuracy achieved by the considered network architectures, when trained with balanced and unbalanced data. The last columns reports the arithmetic mean of the accuracy, and its standard deviation.

CNN	**Balanced training**	**Accuracy (%)**						
		AA	**EA**	**CL**	**AI**	**Overall**	**Mean**	**Std**
VGG-Face	No	79.2	90.3	97.8	71.9	94.4	84.8	10.0
VGG-Face	Yes	82.7	93.0	96.7	77.4	94.4	87.5	7.7
VGG-16	No	80.0	91.0	97.8	69.2	94.4	84.5	10.9
VGG-16	Yes	84.8	92.9	96.0	78.9	94.1	88.2	6.7
MobileNetV2	No	80.9	87.6	97.8	71.2	94.2	84.4	9.7
MobileNetV2	Yes	89.3	93.7	94.1	83.2	93.2	90.1	4.4
ResNet-50	No	80.5	88.0	97.7	66.4	93.9	83.2	11.4
ResNet-50	Yes	87.9	93.1	93.7	82.2	92.7	89.2	4.6

VGG-16 achieve an accuracy of 94.4%, while MobileNet v2 and ResNet-50 94.2% and 93.9%, respectively. Among them, ResNet-50 and VGG-16 obtain a more relevant performance improvement (0.8% and 0.7%), while VGG-Face and MobileNet v2 achieve a smaller increase (0.4% and 0.2%), probably because they start from a higher accuracy.

It is important to clarify that we applied the data augmentation only on the training set and not on the test set, to make a fair comparison; thus, the performance improvement is even more significant and the augmentation techniques demonstrated their effectiveness also on the original (non-synthetic) samples.

Considering the improvement achieved with data augmentation, the other two experiments reported in the following are carried out by applying this technique.

3.4.2 Effect of data balancing

As evident from the results reported in Table 3.2, the CNNs are more specialized in the recognition of Caucasian Latin individuals. This accuracy imbalance is probably due to the different number of samples available for the various ethnicity groups, which implies

an unbalanced prior distribution of the training set and a specialization of the neural networks in the classification of the most represented classes. Performance imbalance is not necessarily a negative factor, since some real problems have intrinsic imbalance; in fact, the ability to recognize more effectively the most representative categories could be a desired behavior. Think, as an example, to a self-service petrol station, which must automatically recognize each type of banknote; the ability to recognize more reliably small denominations, which are presented with higher probability by the customers, is certainly a desired feature.

Nevertheless, especially when the dataset is not balanced due to lack of samples and not for a choice, it might be interesting to balance the accuracy on each class and reduce the imbalance introduced by the a priori distribution of the dataset. To this aim, we investigate this aspect performing a new experiment with a balanced version of the dataset, which consists of around $750,000$ samples. In particular, we train all the CNNs by using a batch composed by the same number of samples for every ethnicity group. In this way, for each epoch the neural networks do not perceive the imbalance and do not rely on the a priori distribution. The results of this experiment are reported in Table 3.2.

When training with this data balancing technique, we expect a more constant accuracy across the different classes, with a possibly smaller overall accuracy, and this is the result indeed, as it can be observed in Figure 3.4. All the architectures achieve higher mean accuracy. From Table 3.2 we also notice that the standard deviation is significantly lower with respect to their non-balanced couterpart: as expected all classes are recognized with similar accuracy.

When using data balancing, MobileNet v2 and ResNet-50 experience a significant drop in their overall accuracy (93.2% vs 94.2% and 92.7% vs 93.9%), while VGG-16 and VGG-Face have a smaller decrease in accuracy; in particular, VGG-Face retains its original 94.4% overall accuracy. However the increment in per-class accuracy is more modest.

This experiments shows the robustness of the VGG-Face ar-

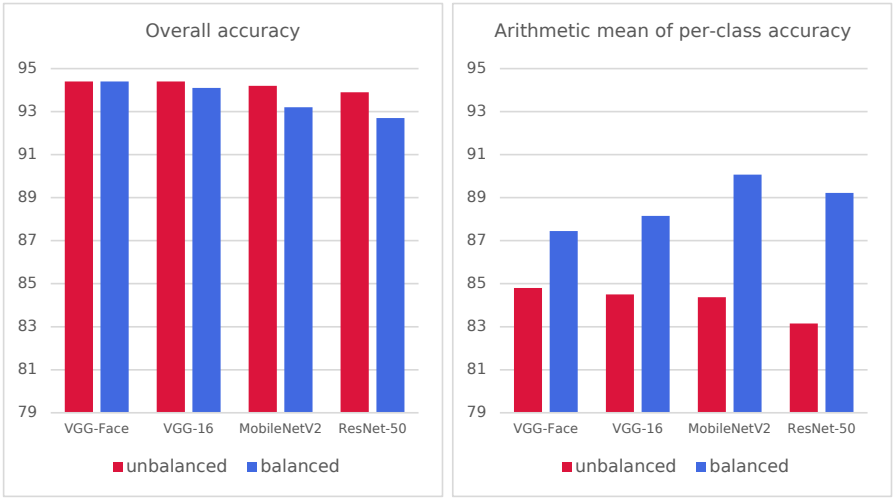

Figure 3.4: Accuracy of our ethnicity recognition models when trained with or without data balancing. On the left we graph the overall accuracy (more represented classes are weighted more); on the right we graph the arithmetic mean of the per-class accuracies.

chitecture as well as the efficacy of balanced training.

3.4.3 Impact of the input size

All the considered networks require an input size of 224×224, while the size of more than 85% of the face images available in the dataset is less than 200×200 pixels. Considering this aspect, we hypothesize that a reduction of the input size should not significantly affect the ethnicity recognition accuracy; on the other hand, it surely implies a gain in terms of training and inference time due to the reduction of parameters and operations.

To this aim, we modify all the considered CNNs to accept an input size equal to 96×96, that is the average size of all the face images in the dataset. Then, we re-train them by applying the same experimental protocol described in Section 3.3.2 with data augmentation.

The results of this experiment are reported in Figure 3.5 and

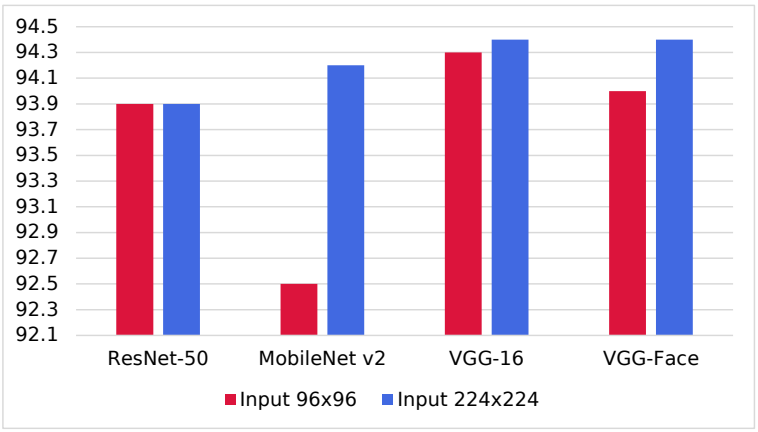

Figure 3.5: Ethnicity recognition accuracy (%) of the considered CNNs by varying the input size (96×96 and 224×224) on the VMER dataset. A significant performance decrease affects only MobileNet v2, while the others are more or less independent on the input size.

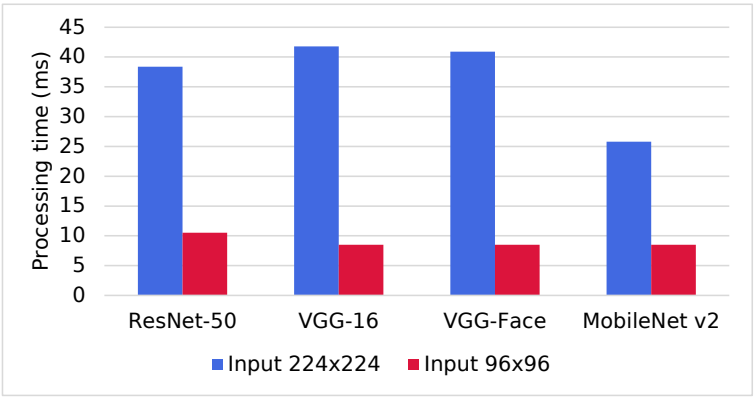

Figure 3.6: Processing time (ms) for a batch of 64 images of the considered CNNs by varying the input size (96×96 and 224×224) on a NVIDIA Titan Xp GPU. The processing time is reduced for all the CNNs of a factor between 3 and 5.

Table 3.3: Per-class and overall ethnicity recognition accuracy achieved by ResNet-34 and VGG-Face on the test sets of VMER, FairFace and UTKFace, by varying the training set. The networks trained with the proposed dataset achieves the best performance over the UTKFace test set, demonstrating that VMER allows to improve the generalization capability.

Test set	**CNN**	**Training set**	**Accuracy (%)**				
			AA	**EA**	**CL**	**AI**	**Overall**
VMER	ResNet-34	VMER	76.6	87.9	97.8	59.8	93.4
	ResNet-34	FairFace	69.3	55.1	96.7	11.5	85.9
	VGG-Face	VMER	79.2	90.3	97.8	71.9	**94.4**
	VGG-Face[1]	FairFace	67.9	83.0	84.1	50.7	81.3
FairFace	ResNet-34	VMER	87.5	81.3	85.0	55.3	80.2
	ResNet-34	FairFace	81.9	88.9	89.9	59.7	**84.3**
	VGG-Face	VMER	86.1	81.5	83.1	56.5	79.4
	VGG-Face[2]	FairFace	81.1	85.0	83.3	43.3	77.6
UTKFace	ResNet-34	VMER	82.7	90.3	96.7	64.3	**89.5**
	ResNet-34	FairFace	69.9	90.2	96.9	31.4	83.5
	VGG-Face	VMER	81.7	90.2	96.4	64.8	89.3
	VGG-Face[3]	FairFace	50.0	73.5	88.7	29.1	75.0

confirm our hypotheses. In fact, only MobileNet v2 has a significant drop of the performance with respect to the original CNN (92.5% vs 94.2%), while ResNet-50, VGG-16 and VGG-Face have a drop of less than 0.5%. On the other hand, such design choice allows to reduce the processing time by a factor between 3 and 5, as shown in Figure 3.6.

Hence, the idea of reducing the input size of the CNNs, adapting the whole architecture to this choice, can be useful whether there are strict constraints in terms of processing time and memory.

3.4.4 Generalization capability

In this Section we perform a cross-dataset experiment to verify whether the proposed VMER allows to improve the generalization capability of the convolutional neural networks trained with its images and labels.

To this aim, we follow the experimental protocol described in [73]. We train the same network architecture with two training sets, namely VMER and FairFace, and evaluate its performance on a third test set, e.g. UTKFace. As done in [73], we use an ImageNet pretrained version of ResNet-34 and run the training procedure with an Adam optimizer and a learning rate of 0.0001 for 100 epochs, until the validation accuracy stops improving. In addition, we perform a similar experiment with VGG-Face, by comparing the performance of the network trained on VMER with the same architecture trained by using FairFace[3].

Since FairFace includes seven ethnicity categories, we follow the instructions given in [73] for reducing the classes to the same four available in VMER. In particular, they propose the following mapping: Indian and Black are trivially mapped on *Asian Indian* and *African American*, East Asian and Southeast Asian are grouped in the *East Asian* class and the remaining categories

[3]We used the VGG-Face model fine tuned on FairFace available in the DeepFace library: https://github.com/serengil/deepface.

(Middle Eastern, White and Latino) are considered *Caucasian Latin*.

The results of this experiment are summarized in Table 3.3. The ResNet-34 and the VGG-Face networks trained on VMER achieve on UTKFace an overall accuracy of 89.5% and 89.3%, respectively; the corresponding CNNs trained on FairFace obtain a substantially lower performance, namely 83.5% and 75.0%. This result shows that the networks trained on VMER have a greater generalization capability, while those trained with FairFace are more specialized on their training set.

In fact, ResNet-34 trained on FairFace achieves on the test set of the same dataset the best accuracy (84.3%), but the performance is significantly lower than the one obtained by the corresponding CNN trained with VMER on the other test sets. Looking at Table 3.3, we notice that VMER allows to better generalize on the Asian Indian samples, while the ResNet-34 trained with FairFace have a dramatic decrease of the accuracy on this category (31.4% for UTKFace, 11.5% for VMER). This difference is probably due to the greater number of samples available in VMER for each category and to the high accuracy of the ethnicity annotations.

3.4.5 Feature visualization

The last experiment we present has the aim of visualizing the discriminative features learned by a CNN trained with VMER. To achieve this goal, we firstly compute the class activation maps [152] to determine the regions in the image which are relevant for recognizing a specific ethnicity category. A class activation map is a heat map computed for each pixel of the input image; its pixels with red color gradations correspond to the regions of the image most used by the neural network to recognize the specific class to which the sample belongs. Since this technique is designed for network architectures having an average pooling and a linear dense layer after the final convolutional layer, we applied it on VGG-Face by using the tool available in keras-vis[4].

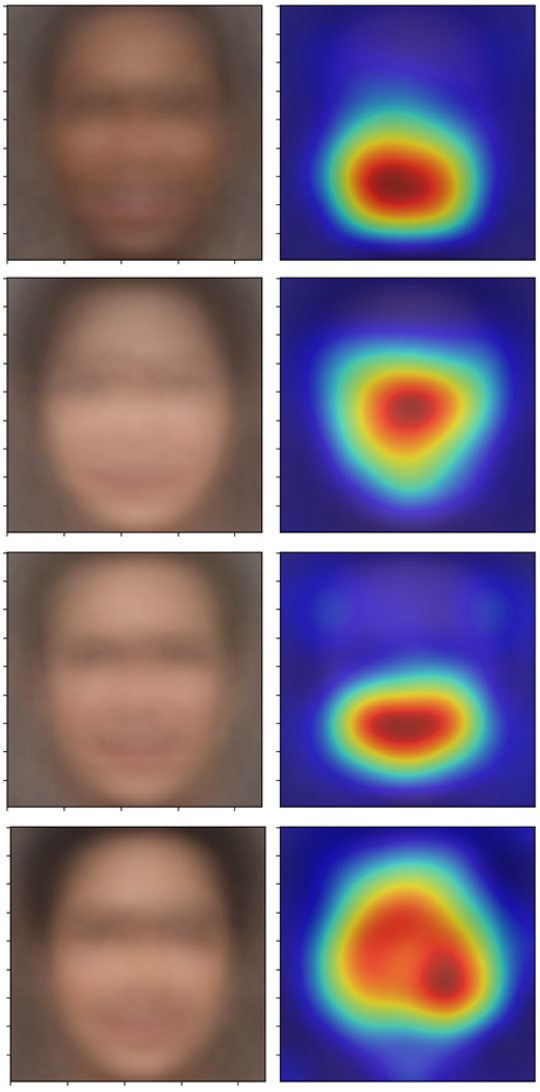

Figure 3.7: Average face images and class activation maps obtained by applying our VGG-Face trained on VMER over all the African American (first row), East Asian (second row), Caucasian Latin (third row) and Asian Indian (fourth row) samples. The parts in red correspond to the face regions more relevant for determining the ethnicity.

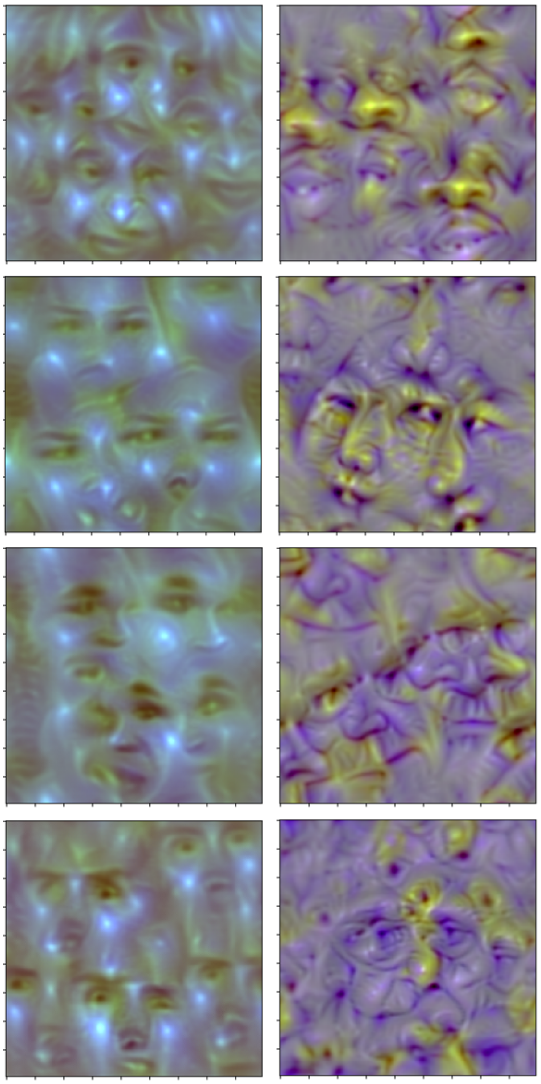

Figure 3.8: Result of the Activation Maximization applied on four output neurons of the original VGG-Face trained for face recognition (first column) and of the one fine-tuned for ethnicity recognition (second column). The neurons of the original CNN are sensitive to the whole face image, while the ones belonging to our version are activated by specific parts of the face.

Figure 3.7 shows the average class activation maps obtained when VGG-Face is applied on African American, East Asian, Caucasian Latin and Asian Indian samples. It is evident that the considered CNN recognizes the African American samples by analyzing the region of the lips and the nose, which are discriminative features for this ethnicity. The neuron that recognizes East Asian faces focuses its attention on the region including the eyes and the nose, whose particular shape and size are distinctive facial traits. As evident in Figure 3.7, the average images of Caucasian Latins and Asian Indians are quite similar and the distinction between the two ethnicity categories is harder. The average class activation maps show that the CNN focus its attention on the lower part of the face for Caucasian Latins, while for the Asian Indians is arguably more sparse, including the forehead and the cheekbones.

To find deeper insights about the features learned with our procedure, we apply the Activation Maximization (AM) method [153] over the four neurons in the output layer of our VGG-Face network fine tuned on the VMER dataset, and over four neurons of the last layer of the original one pre-trained for face recognition. This method allows to iteratively generate the image patterns that maximize the activation of the considered neuron; therefore, we can infer the distinctive facial traits for each ethnicity and the differences with respect to the original features.

Figure 3.8 shows the results of the Activation Maximization. We can note that the image patterns which maximize the activation of the output neurons of the original VGG-Face, optimized for face recognition, include more or less the whole face. On the other hand, the output neurons of the VGG-Face fine-tuned for ethnicity recognition are sensitive to more specific facial traits, consistently with respect to the class activation maps.

In particular, the output neuron responsible for the classification of African Americans is activated by full lips and wide noses, while the one that recognizes East Asians is sensitive to almond eyes and small noses. The output neuron specialized in the Caucasian Latin category uses thin lips and particular shapes of the nose and of the eyes to recognize face images belonging to this

category. Finally, we are not able to find distinctive facial traits which activate the output neuron responsible for Asian Indians; the lack of focus on specific image patterns partially explains the difficulties of the CNN in recognizing samples of this class. We believe that the recognition of this particular ethnicity deserves further future investigations.

[4]https://github.com/raghakot/keras-vis

Chapter 4

Towards robust emotion recognition in extreme conditions

4.1 Background

In this chapter we focus on the task of emotion recognition. As anticipated in Section 1.3.2.2, we recognize that real-world imagery from application scenarios presents numerous challenges that do not appear in standard benchmarks, namely corruption of single images and perturbation across a sequence of images. For this reason, we aim to evaluate CNN-based methods for emotion recognition from faces with respect to common image corruptions and perturbations.

Following on the work of [135], we define a set of image corruptions typical for the application at hand, and a set of perturbations on subsequent frames of a video sequence. We evaluate the effect of architectural and data-related changes on the robustness of the methods, namely the use of an anti-aliasing filter before down-sampling operations [134] and the AutoAugment policies [138] for augmenting training data, respectively. We constructed a new benchmark data set by modifying the RAF-DB data set with custom corruptions and perturbations of different intensity. We generated a new validation set for each corruption (18 corruption times 5 intensity levels) and for each perturbation type (10 sets).

We benchmark the performance of networks that ranked among the best performing models on the RAF-DB data set, namely SENet, Xception and DenseNet, when the input images are subjected to corruptions and perturbations. We use the VGG architecture as the baseline network for our evaluation, since it has been widely used for face analysis [132]. We also evaluate an handcrafted feature-based methods, namely LBP histograms with a Support Vector Machine classifier [154], which achieved the state-of-the-art performance before deep networks became the *de-facto* standard for many computer vision applications, including facial expression recognition [155].

Furthermore we study the impact of two different improvements: a technique concerning training data augmentation (i.e. AutoAugment), and one architectural modification, the use of an-

tialiasing in down-sampling operations.

The code, the data set, the trained network models and the AutoAugment policies for the considered application are made publicly available.

The chapter is structured as follows: in Section 4.2, we describe the experimental framework, data and evaluation metrics, the considered methods and the training hyper-parameters, and the modifications we deployed to improve the robustness of existing models. In Section 4.3 we report and discuss the results that we achieved.

4.2 Experimental framework

We defined a benchmark framework for evaluation of classifier robustness, based on the approach proposed in [135]. We trained several methods on the images of the original training set of the RAF-DB data set and tested on several corrupted and perturbed versions of the test set. The benchmark does not involve training on a corrupted or perturbed version of the training set.

We designed the corruptions and perturbations of the test images to simulate out-of-distribution samples that occur in real applications of emotion recognition. We call RAF-DB-C and RAF-DB-P the corrupted and perturbed test sets, respectively.

We trained enhanced methods for increased robustness. On one hand we studied the effects of architectural changes in the network design and on the other hand, we evaluated the modification of the training data by means of specific data augmentation policies.

In the rest of the section, we provide details about the experimental framework, namely the data, the methods and the evaluation protocol that we adopted.

4.2.1 Data set and evaluation metrics

The RAF-DB data set is one of the most popular benchmarks for emotion recognition [109]. It consists of 29,672 face images, of

which 15,339 are annotated with the six basic emotions theorized by [156], plus a neutral class that represents the absence of emotion. The images are divided in a training (12,271 instances) and a test set (3,068 instances); we used the detected, cropped and aligned faces, according to the indications of the authors.

The data set is widely used because of its reliable ground truth: each image is annotated by 40 different individuals and the multi-label annotation is available as a seven dimensional vector, in which the number of votes for each class are provided. This allows the training procedure to take advantage of the intrinsic ambiguity of emotion evaluation.

In the training phase we discarded the samples with ambiguous annotation, according to the criteria described by [108], and computed the class probability distribution removing the outlier votes. In detail:

- the votes of the classes with less than 10% of the total votes is set to zero;
- the samples as *no face* or *unknown* are discarded;
- the samples with more than two classes with equal votes are discarded;
- the samples for which the winner class has less than 50% of the votes are discarded;
- the label vector is normalized to length 1.

4.2.1.1 RAF-DB-C

We created the RAF-DB-C data set by applying to the images contained in the RAF-DB test set 18 corruptions from a set C that we will describe in this paragraph. We extended the corruptions proposed by [135] with five others that are common in face analysis problems. Each type of corruption $c \in C$ is applied to the original images with five different levels of severity $s \in S$, where $S = \{1, 2, 3, 4, 5\}$.

We grouped the corruptions in four categories, namely *blur*, *noise*, *digital* and *mixed*; details about their implementation and the values of the used parameters[1] In Fig. 4.1, we show some examples of the considered corruptions (rows) with different severity (columns).

The people framed in real environments are typically not aware of the presence of the camera. Therefore, the movement of their faces, often very sudden, can cause blur on the acquired image. In addition, manual blur can be added in the face pre-processing to improve the image given as input to a neural network. To take into account these possible corruptions, we consider the *blur* category, including *Gaussian blur*, *defocus blur*, *zoom blur* and *motion blur*. The Gaussian blur is deliberately applied as a pre-processing step on the acquired image to mitigate the effect of acquisition noise and to enhance the image patterns at different scales. The defocus blur occurs when using cameras with limited depth of field (DoF) deployed in scenarios with large DoF. Zoom blur appears when a person moves towards the camera rapidly, so increasing the size of the face captured by the sensor. Motion blur occurs when a subject suddenly moves, quickly changing his face pose.

Image noise is a corruption due to the electronic noise produced by the image sensor at high temperatures or with long exposure; it appears as random speckles that can substantially degrade the image quality. In particular, the *noise* corruptions include *Gaussian noise* and *shot noise*. The first corruption increases proportionally with the temperature of the CMOS sensor; considering that in real environments the camera is always active and the sensor works perpetually at high temperatures, this source of noise is very common. The shot noise is a corruption occurring in case of high exposure, that is typical for installations in shop windows or for cameras pointing to the store entrance.

Other corruptions very relevant in real environments are due to manual camera settings, image compression and image transformation. To improve the image rendering, the automatic gain

[1]The code is available at https://github.com/MiviaLab/emotion-robustness are reported in Table 4.1.

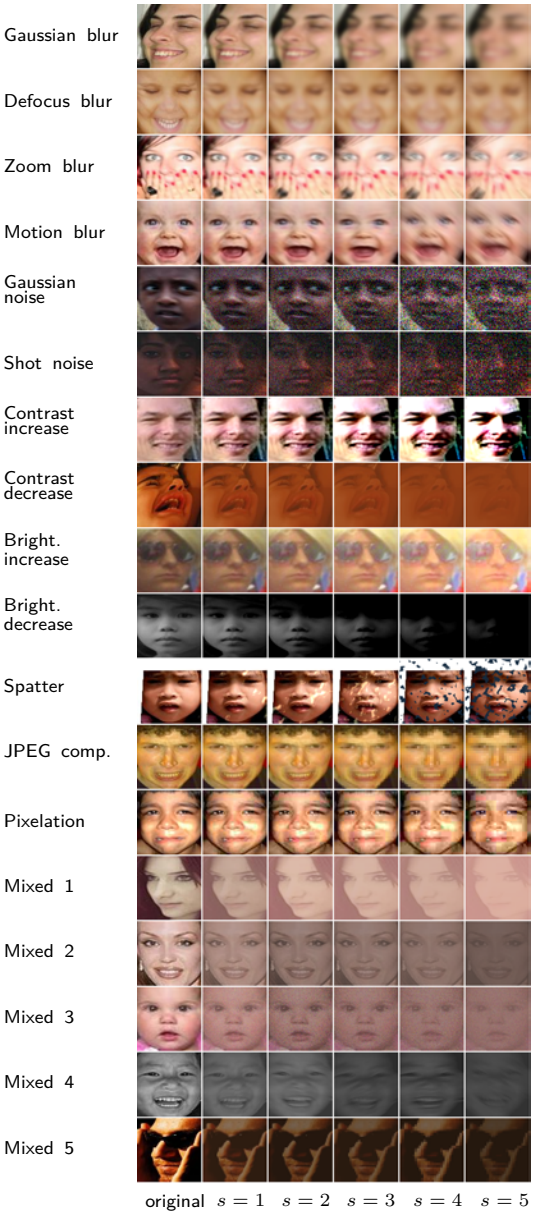

Figure 4.1: Examples of corruptions. The first column contains images from the original RAF-DB test set, while the others depict the versions obtained by applying the considered corruptions with increasing value of severity (from 1 to 5).

Table 4.1: Details and parameters for the implementation of the corruptions at different severity. $I(x, y, d)$ refers at the original image and $I_c(x, y, d)$ at the corrupted image, while $d \in \{R, G, B\}$ indicates the channel in the RGB space. Mixed corruptions are obtained as a combination of basic corruptions.

Corruption	**Param.**	**Value (per severity)**					**Description**
		$s=1$	$s=2$	$s=3$	$s=4$	$s=5$	
Gaussian blur	σ	1	1.8	2.6	3.4	4	$I_c(x,y,d) = I(x,y,d) * G_\sigma(x,y), \forall d$ where $G_\sigma(x,y) = e^{(x^2+y^2)/2\sigma^2}$
Defocus blur	r	1.5	2	2	2.5	3	$I_c(x,y,d) = I(x,y,d) * (D_r(x,y) * G_\sigma(x,y))$, being D_r a disc shaped kernel[2] with radius r
	σ	0.1	0.2	0.3	0.4	0.4	
Zoom blur	z	1.11	1.18	1.26	1.32	1.4	$I_c(x,y,d) = \frac{1}{\lvert T \rvert} \sum_t zoom_t(I(x,y,d))$, $t \in T = \{t = 1 + n\epsilon, \quad t \leq z, \quad \forall n \in \mathbb{N}\}$ $zoom_t(I)$ enlarges the image I by a factor t using linear interpolation[3].
	ϵ	0.01	0.01	0.02	0.02	0.03	
Motion blur	r	3.3	5	5	5	6.7	Implementation from the ImageMagick library[4], with random angle[2].
	σ	1	1.7	2.7	4	5	
Gaussian noise	σ	0.08	0.12	0.18	0.24	0.3	$I_c(x,y,d) = I(x,y,d) + \mathcal{N}(0, \sigma^2)$
Shot noise	q	60	29	15	8	5	$I_c(x,y,d) = Poisson(I(x,y,d) * q)/q$
Contrast incr.	q	1.5	1.9	2.6	3.3	5	$I_c(x,y,d) = (I(x,y,d) - \mu_d) * q + \mu_d$, where μ_d is the average value of the original image I over the channel d.
Contrast decr.	q	0.4	0.33	0.24	0.16	0.1	
Brightness incr.	q	0.1	0.2	0.3	0.4	0.5	$I_c(x,y,v) = I(x,y,v) + q, \quad \forall x, y$, where v is the v channel in hsv image representation
Brightness decr.	q	-0.1	-0.2	-0.3	-0.4	-0.5	
Spatter	c_0	0.65	0.65	0.65	0.65	0.67	Implementation by [135].
	c_1	0.3	0.3	0.3	0.3	0.4	
	c_2	4	3	2	1	1	
	c_3	0.69	0.68	0.68	0.65	0.65	
	c_4	0	0	0	1	1	
JPEG compr.	quality	25	18	15	10	7	Implementation from the Pillow library[5].
Pixelation	q	0.6	0.5	0.41	0.3	0.25	$I_c(x,y,d) = I(\lfloor xf \rfloor / q, \lfloor yf \rfloor / q, d)$,
Mixed 1 brightness incr. + contrast decr.	s_b	1	2	2	2	3	$I_1 = BrightnessIncrease(I, s_b)$ $I_c = ContrastDecrease(I_1, s_c)$
	s_c	1	1	2	3	4	
Mixed 2 brightness decr. + contrast decr.	s_b	1	2	2	2	3	$I_1 = BrightnessDecrease(I, s_b)$ $I_c = ContrastDecrease(I_1, s_c)$
	s_c	1	1	2	3	4	
Mixed 3 gaussian noise + brightness decr. + contrast decr.	s_g	1	2	2	3	3	$I_1 = GaussianNoise(I, s_m)$ $I_2 = BrightnessDecrease(I_1, s_b)$ $I_c = ContrastDecrease(I_2, s_c)$
	s_b	1	2	2	2	2	
	s_c	1	1	2	3	4	
Mixed 4 motion blur + contrast decr. + brightness decr.	s_m	2	3	4	5	5	$I_1 = MotionBlur(I, s_m)$ $I_2 = BrightnessDecrease(I_1, s_b)$ $I_c = ContrastDecrease(I_2, s_c)$
	s_b	1	1	2	2	2	
	s_c	1	1	2	1	3	
Mixed 5 pixelation + contrast decr. + brightness decr.	s_p	1	2	3	4	4	$I_1 = Pixelation(I, s_p)$ $I_2 = BrightnessDecrease(I_1, s_b)$ $I_c = ContrastDecrease(I_2, s_c)$
	s_b	1	2	2	2	3	
	s_c	1	1	2	1	3	

control (AGC) and the digital wide dynamic range (DWDR, also called Dynamic Contrast) dynamically modify the brightness and the contrast of the acquired image according to the environmental conditions. Often, the image processing is performed on an external server, acquiring the frames in motion JPEG (MJPEG) to reduce the required bandwidth; of course, the compressed image may lose quality and details. Furthermore, the faces are subjected to rescaling for adapting them to the input size of the neural network adopted for emotion recognition. We group all these corruptions in the *digital* category. Therefore, the *digital corruptions* include *contrast increase*, *contrast decrease*, *brightness increase*, *brightness decrease*, *spatter*, *JPEG compression* and *pixelation*.

Contrast and brightness are variations of the image due to the lighting conditions or to specific camera settings. Brightness variations occur outdoor with daylight, while in indoor environments it depends on the artificial illumination. Contrast corruptions occur when the difference between the brightest and the darkest pixels in the image (dynamic range) is high. The *spatter* consists of random patterns of obstructions on the camera lens. We simulate this effect by adding bright occlusions for low corruption severity and dark patterns for higher corruption severity. *JPEG compression* is often used to reduce the amount of data transferred on networks to an external server for the real-time processing of the images and can introduce compression artifacts. We reproduce its destructive effects by gradually decreasing the quality of the compression. Image *pixelation* happens when an image is scaled from a lower resolution to a higher one. It is typical for face analysis in real environments, since the faces have a smaller resolution than

[2] The r and σ parameter value are expressed in pixels and referred to a 48×48 image. The value of the parameter is scaled proportionally with larger images.

[3] Implementation from the SciPy library: https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.zoom.html

[4] https://imagemagick.org/api/effect.php#MotionBlurImage

[5] https://pillow.readthedocs.io/en/3.1.x/handbook/image-file-formats.html#jpeg

the input size of the neural networks used for facial soft biometrics recognition.

The *mixed* corruptions, that we added to those proposed by [135], consist of combinations of single corruption types that simulate other challenging real-world scenarios. We create five mixed corruptions. We combine the *contrast decrease* with *brightness increase* and *brightness decrease*, which occur when the automatic exposure control of the cameras targets different elements in the scene and the resulting dynamic range of the images is compressed. In environments with low illumination, contrast and brightness decrease together: we combined it with *added Gaussian noise*, to simulate high gain on the camera sensor, and with *motion blur*, usually caused by long shutter times. We also combined contrast and brightness decrease with *pixelation*, to simulate faces at low resolution in dark environments.

For the evaluation, we adopted the experimental protocol proposed by [135]. Let E_d indicate the classification error on a data set d, computed as the ratio between the number of wrongly classified images and the total number of images. Therefore, we use E_o to indicate the classification error on the original RAF-DB test set, and E_c for the classification error obtained on a set of images with corruption type $c \in C$. Since the images with corruption c are provided with different levels of corruption severity, we compute the classification error E_c as the average of the errors obtained for each severity:

$$E_c = \frac{1}{|S|} \sum_{s \in S} E_{c,s} \tag{4.1}$$

where $E_{c,s}$ is the classification error on samples with corruption type c and severity level s.

Then, we compute the corruption error $\widetilde{E}_c$ as the classification error E_c normalized by the error E_c^{b} obtained by another classifier taken as the baseline:

$$\widetilde{E}_c = \frac{E_c}{E_c^{\mathrm{b}}} \tag{4.2}$$

Finally, we compute the mean corruption error $\overline{E}$ as the average of the $\widetilde{E}_c$ over all the corruptions $c \in C$:

$$\overline{E} = \frac{1}{|C|} \sum_{c \in C} \widetilde{E}_c \tag{4.3}$$

where $|C|$ is the cardinality of the set C. The smaller the value of $\overline{E}$, the better the robustness of the method with respect to the corruptions.

In addition to the absolute error, we compute the relative corruption error $\widetilde{RE}_c$ as the gap between the classification error on the original test set E_o and that on the corrupted sets E_c, normalized with respect to the error gap achieved by the baseline. For a specific corruption type c, it is defined as:

$$\widetilde{RE}_c = \frac{E_c - E_o}{E_c^{\mathrm{b}} - E_o^{\mathrm{b}}} \tag{4.4}$$

Finally, we compute the *mean relative corruption error* $\overline{RE}$, i.e. the average of the $\widetilde{RE}$ achieved for all the considered corruptions.

$$\overline{RE} = \frac{1}{|C|} \sum_{c \in C} \widetilde{RE}_c \tag{4.5}$$

4.2.1.2 RAF-DB-P

We created the RAF-DB-P test set by applying 10 types of perturbation to the images of the RAF-DB test set. In contrast to the corruptions, a perturbation concerns a sequence of frames: it consists of a small corruption incrementally applied to subsequent frames. Its temporal character determines substantial appearance changes between the first and last frame of a sequence. The perturbations challenge the performance of the recognition methods when they are applied in real scenarios and have to perform analyses over time.

We selected a set P of perturbations that typically occur when dealing with faces in real scenarios. For each perturbation $p \in P$, an image in the RAF-DB test set is replicated into a sequence of 30

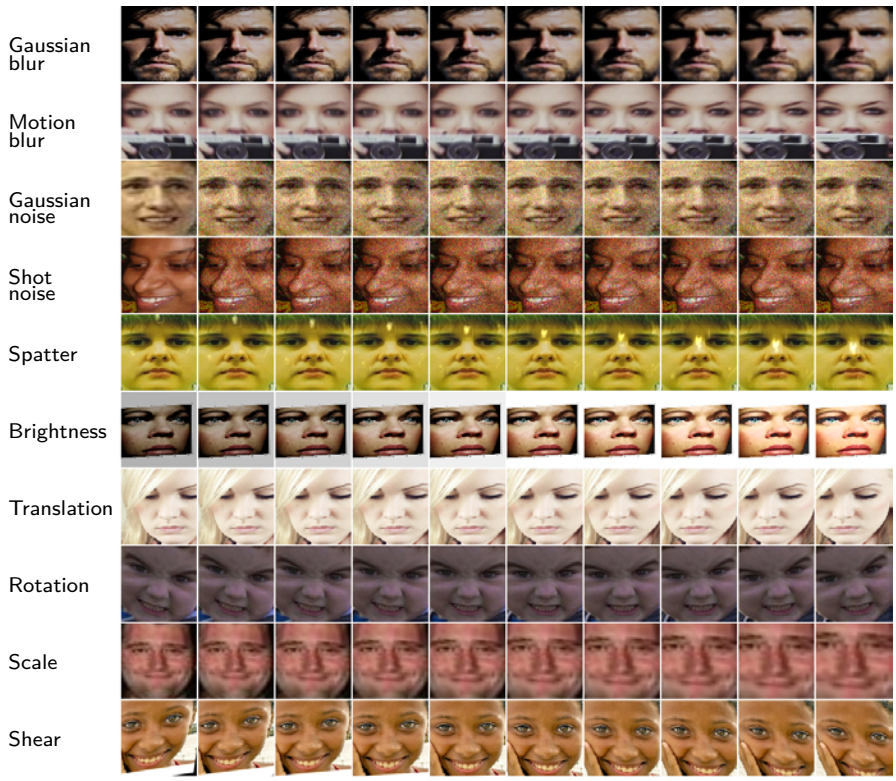

Figure 4.2: Examples of perturbations. Each row contains a perturbation type, whose temporal sequence is represented from left to right (one every three frames is shown).

frames, each with slight changes with respect to the previous one. In Fig. 4.2, we show few examples of perturbed image sequences.

We group the perturbations in four classes, namely *blur*, *noise*, *digital* and *transformation*. To implement the first category, a different noise pattern is applied to the original images, independent among different frames of the sequence. For all the other perturbations, the modifications are applied to each frame of the sequence in an incremental way.

The *blur* category includes *Gaussian blur* and *motion blur*. Perturbed sequences with Gaussian blur are generated by apply-

ing a Gaussian blur corruption incrementally frame after frame: the standard deviation of the Gaussian blur for the j-th frame is $\sigma_j = 0.25 + 0.035i$, where $j \in [0, 29]$. Similarly, sequences with motion blur are generated applying a motion blur corruption with $r = 10$ and $\sigma = 3$ and motion angle θ_j that increases for consecutive frames as $\theta_j = (4 \cdot j)^\circ$, where $j \in [0, 29]$.

Noise perturbations include *Gaussian noise* and *shot noise*. The perturbed sequences are generated applying the corresponding Gaussian and shot noise corruption with the severity $s = 2$ repeatedly to the original image, as the noise has no inter-frame dependency.

The *digital* perturbations are *spatter* and *brightness increase*. For the spatter perturbation, a pattern of translucent water droplets is created and superimposed to the first frame causing occlusions. For subsequent frames, the droplet pattern is incrementally blurred and shifted downwards on the image. The implementation and the parameters are those used by [135]. In the case of brightness perturbations, subsequent frames of a sequence are modified by applying an incremental brightness increase corruption to the previous frame: at the j-th frame, the control parameter q_j has value $q_j = \frac{j-15}{50}$, with $j \in [0, 29]$.

Finally, the *transformation* category includes *translation*, *rotation*, *scale* and *shear* perturbations. Translation consists in shifting the image for one pixel to the right with respect to the previous frame. Rotation is implemented by rotating the face image one degree counterclockwise at every consecutive frame in the range $[-15; 15]$ degrees. The scale transformation is obtained as follows. We define a region of interest around the face in an image of the RAF-DB test set of size $w \times w$ and centered at location (x, y). We change the size of the region in the j-th frame of the perturbed sequence to $w_j \times w_j$, where $w_j = w \cdot (0.79 + \frac{29-j}{29} 0.51)$ with $j \in [0, 29]$, and keep its center location fixed in (x, y). This results in a loose face crop in frame 0 and increasingly tighter crops in subsequent frames.

These three transformations are experienced in real-world applications due to the imprecision of face detection and face align-

ment algorithms. Typically, smoothing algorithms such as the Kalman filter are used to track a face producing a delay, which causes slight but continuous variations of position, rotation and scale. The shear transformation simulates a change of perspective of the face by bending the image according to the affine transformation $\binom{x'}{y'} = \left(\begin{smallmatrix} 1 & \alpha \\ \alpha & 1 \end{smallmatrix}\right)\binom{x}{y}$, where the parameter $\alpha \in [-0.15; 0.15]$ is increased in steps of 0.01.

To evaluate the stability of classification on perturbed image sequences, we compute the flip probability (F). It measures the likelihood that the predicted class changes across consecutive frames of a sequence of N frames $\boldsymbol{x} = \{x_1, x_2, \ldots, x_N \ | x_i \in \mathbb{X}\}$. Given a classification method $f : \mathbb{X} \to \{1, 2, \ldots, M\}$, which assigns one of M classes to images $x_i \in \mathbb{X}$, the flip probability for the sequence $\boldsymbol{x}$ is computed as the average of the number of changes of the classification output across consecutive frame. It is defined as:

$$F(\boldsymbol{x}) = \frac{1}{N-1} \sum_{j=2}^{N} [1 - \delta(f(x_j), f(x_{j-1}))] \tag{4.6}$$

where $\delta(\cdot, \cdot)$ is the Kronecker delta function; note that the function $1 - \delta(f(x_j), f(x_{j-1}))$ assumes value equal to 0 if the method f predicts the same class for the frames x_{j-1} and x_j, 1 otherwise.

Let us consider a perturbation type p from the set P. We compute the flip probability for the set of sequences with perturbation p as the average of the flip probability computed for each sequence $\boldsymbol{x} \in p$:

$$F_p = \frac{1}{|p|} \sum_{\boldsymbol{x} \in p} F(\boldsymbol{x}) \tag{4.7}$$

As the F_p may assume values in different ranges for different perturbations $p \in P$, we normalize its value by the corresponding flip probability F_p^b achieved by another classifier taken as the baseline. The normalized flip probability is computed as:

$$\widetilde{F_p} = \frac{F_p}{F_p^b} \tag{4.8}$$

The overall measure of the flip rate for all perturbation types in the set P is the average of the flip rate of the single perturbations $\widetilde{F_p}$ and is called *mean Flip Rate* and is defined as:

$$\overline{F} = \frac{1}{|P|} \sum_{p \in P} \widetilde{F_p} \tag{4.9}$$

The smaller this value, the better the stability of the method against perturbations.

4.2.2 Methods

We benchmarked the performance of different convolutional network models, namely VGG, SENet, Xception and DenseNet, as well as that of a method based on Local Binary Patterns (LBP) and an SVM classifier, that achieved the state-of-the-art performance prior to the development of deep-learning models for automatic feature learning. We chose different methods with peculiar architectures to evaluate the impact of the specific design choices on the overall network robustness.

VGG. We selected the VGG-Face network as the baseline for our analysis [144]. It is one of the most widely used networks for facial soft biometrics analysis, based on the VGG-16 architecture [132], trained for face recognition on about 1M images. VGG-Face achieved high performance in the face recognition [144] and age estimation [31] problems. The VGG architecture was shown to have good generalization capabilities also for small data sets [157]. Its lightweight model has only 130k parameters.

SENet. Designed by [158], it achieved the state-of-the-art performance on emotion recognition [159]. The version that we apply, namely SENet-50 (i.e. a SENet with 50 layers), is based

on ResNet-50 [130]. SENet uses Squeeze-and-Excitation modules, adaptively weighting the input channels when computing the output feature maps. The intuition to explicitly model the inter-dependencies between the channels was demonstrated to be a winning strategy, reducing the error by 25% on the ImageNet benchmark over its plain ResNet counterpart [158]. The trained model has about 25M parameters.

DenseNet. The Densely Connected Convolutional Network is designed to increase the representation capabilities of the underlying network model [160]. It adopts a dense connectivity scheme to improve the information flow between the layers, by forwarding and concatenating feature maps to subsequent layers and using a growth rate to establish how much each layer contributes to the global state. The architecture of DenseNet leverages transition layers, which do convolution and pooling between connected dense blocks, to normalize the size of the feature maps computed by different layers. We used the DenseNet-121-32 network, which has 121 layers and growth rate $k = 32$. It obtained a good trade-off of performance and size on the ImageNet classification challenge. It has $7M$ parameters, substantially less than the other architectures with comparable performance. To the best of our knowledge, the DenseNet architecture has not yet been benchmarked on the problem of facial expression recognition.

Xception. Proposed by [142], this network architecture inherits the large use of identity connections of the ResNet architecture and combines it with inception modules and depth-wise separable convolutions. Its architecture holds the advantage of factoring convolutions into different branches and separates the convolutions in depth-wise and point-wise components, so retaining the performance of the network while reducing the number of parameters to 20M.

LBP-SVM. LBP is the acronym for Local Binary Patterns [161], a method for dynamic texture description that

achieved the state-of-the-art performance in facial expression recognition before CNNs became popular for image classification [154]. The LBP descriptor is invariant to translations and rotations, and robust to illumination changes. Each pixel is compared to its 8 neighbours: the 8-digit binary code will have a 1 or a 0 in each position according to the pixel being brighter than its neighbour or not. The extended operator, devised by [162], considers neighbourhoods of different size. Following the method proposed by [154], we compute the $LBP^{u2}_{8,2}$ by [162], that means that we consider 8 neighbours, but with a distance of 2 from the central pixel. Furthermore, we do not consider all the possible 256 patterns as bins for the histogram, but only the 59 uniform ones, as they are shown to preserve large part of the information while reducing the dimensionality. We resample the image to 110×150 pixels, then we divide it in 6×7 blocks and compute the histogram on each of them. The feature vector is composed of the concatenation of the block-wise vectors. As suggested in the reference method, we use an SVM with a RBF kernel as classifier and we perform a grid search to select the optimal values for the parameters C and γ for the RBF-SVM; selected values are $C = 4$ and $\gamma = 3 \cdot 10^{-6}$.

4.2.3 Training procedure

The pre-training of a CNN on face recognition demonstrated its effectiveness for improving the classification performance on tasks like facial gender recognition and age estimation [57, 31]. Hence, we start from methods pre-trained on the VGG-Face2 data set. When available, we used the model weights released by the authors (e.g. for VGG and SENet). In the case of Xception and DenseNet we pre-trained the networks ourselves by following the protocol of [32].

Subsequently, we trained the networks on the RAF-DB data set for 220 epochs. We used the Stochastic Gradient Descent optimizer with a momentum equal to 0.9, a batch size of 64, and an initial learning rate equal to 0.002 that we halved every 40 epochs.

We used a cross-entropy loss with a weight decay of 0.005. We resized the input images to native input size of each network, i.e. 299×299 for Xception, 224×224 for all the others, and zero-centered every channel by subtracting the mean computed on the VGGFace2 data set. Training on zero-centered data is very common and improves the convergence of the loss function.

We applied a standard augmentation commonly used for face analysis and emotion recognition [163]. The augmentation strategy, that we call hereinafter *basic*, includes random rotation, shear, cropping, horizontal flipping and change of brightness and contrast. The horizontal flipping is applied with a probability of 0.5, the random rotation is chosen in a range of ± 10 degrees, while the shear matrix $\left(\begin{smallmatrix} 1 & \alpha_1 \\ \alpha_2 & 1 \end{smallmatrix}\right)$ uses two independent values of α_1 and α_2 between 0 and 0.1. The contrast can be randomly increased or decreased by a factor up to 2, and brightness increases or decreases up to 20% of the maximum value.

4.2.4 Robustness and stability improvement

On one hand, the robustness of convolutional models to corruptions and perturbations is affected by the quality of the training data. On the other hand, certain architectural components of the networks may influence the performance when the input data undergoes specific types of transformation. For instance, the max-pooling (or strided convolution) layers introduce aliasing in the intermediate feature maps and the networks do not provide stable predictions on translated inputs.

We evaluated the impact that targeted expansion of the input data, using the AutoAugment data augmentation technique, and a modification of the CNN architecture with the use of an anti-aliasing filter before down-sampling, have on the robustness of the existing network models. We also evaluated the combined contribution of data- and architecture-related modifications on the robustness of SOTA methods.

4.2.4.1 AutoAugment

AutoAugment is an automated procedure for determining a set of data augmentation policies for a specific image classification problem [138]. It searches augmentation policies in a space with 16 basic operations, namely shear, translate, rotate, auto-contrast, invert, equalize, solarize, posterize, contrast, color balance, brightness, sharpness, cutout, sample pairing. The augmentations are applied with a certain probability and magnitude, for a total of $15k$ policies. The search algorithm, based on Reinforcement Learning, uses a Recurrent Neural Network as controller and a Proximal Policy Optimization as training strategy. The result of the training is a set including the 25 best policies for training a network on the target data set. The authors demonstrated the improvement achieved with AutoAugment on four benchmark data sets, namely CIFAR-10, CIFAR-100, SVHN and ImageNet, and its superiority w.r.t. other augmentation techniques. A faster policy search algorithm called Fast-AutoAugment was developed by [164], based on density matching.

We evaluated the contribution that the AutoAugment data augmentation makes to improve the robustness of the considered methods to common corruptions and perturbations. We learned and made available [6] the augmentation policies on the RAF-DB data set. For searching the augmentation policies, we applied the Fast-AutoAugment method. We used a reduced version of the RAF-DB data set that includes 20% of the training data and 40% of the validation data.

We append the suffix $|_a$ to the name of the methods that are trained with AutoAugment data augmentation, e.g. VGG$|_a$ indicates the VGG method trained with AutoAugment. The need of learning a new set of AutoAugment sub-policies highlights the fact that data augmentation techniques are dataset-specific and do not generalize well to different input images.

[6] The optimized policies are public at https://github.com/MiviaLab/emotion-robustness.

4.2.4.2 Anti-aliasing filters

Although the convolution operator is translation-invariant, current CNN are not, as shown by [140]. This is caused by local pooling strategies, which introduce aliasing into intermediate representations computed inside the networks. Thus, small translations can cause dramatic performance drops. [140] demonstrated that a low-pass filter (LPF) before down-sampling reduces the aliasing in CNNs, according to the Nyquist-Shannon theorem of sampling.

We modified existing networks for face analysis and analyzed the impact that an LPF has on the robustness to several types of corruption and perturbation. We considered the three LPFs of different size: 2×2 is a rectangular filter $[1, 1]$, the 3×3 triangle filter is given by the convolution of two box filters $[1, 2, 1]$, and the 5×5 binomial filter is given by the repeated convolution of rectangular filters $[1, 4, 6, 4, 1]$.

We append the suffix $|_r$, $|_t$ and $|_b$ to the name of the methods that use the 2×2, 3×3 and 5×5 LPF, respectively.

4.3 Results

We carried out experiments to evaluate the robustness of SOTA methods for facial emotion recognition to corruptions and perturbations of the input data. In the following of the section, we report the results that we achieved on the RAF-DB, RAF-DB-C and RAF-DB-P data sets with the considered existing methods. We discuss the impact that the AutoAugment data augmentation and the insertion of anti-aliasing filters within their architecture have on the performance on corrupted and perturbed data in terms of robustness, generalization abilities and stability of the classification output.

Table 4.2: Results obtained by the considered methods on the RAF-DB, RAF-DB-C and RAF-DB-P data sets. We report the classification error on the original test set ($E_{\text{RAF-DB}}$) and on the corrupted one ($E_{\text{RAF-DB-C}}$), while $\overline{E}$, $\overline{RE}$ and $\overline{F}$ are normalized with respect to the results of VGG.

Network	$E_{\text{RAF-DB}}$	$E_{\text{RAF-DB-C}}$	$\overline{E}$	$\overline{RE}$	$\overline{F}$
VGG	14.28	26.38	1.000	1.000	1.000
SENet	13.69	27.93	1.033	1.089	1.162
DenseNet	15.91	27.43	1.027	0.863	1.355
Xception	17.35	31.94	1.208	1.179	1.789
LBP-SVM	24.92	43.14	1.364	1.341	2.452

4.3.1 Baseline results

We trained the VGG, SENet, DenseNet, Xception and the LBP-SVM methods on the RAF-DB original training set and tested on the RAF-DB, RAF-DB-C and RAF-DB-P test sets. These experiments evaluate the ground capabilities of the considered methods for facial emotion recognition when input images are subjected to corruptions and perturbations. Here, we take VGG as the baseline to compute the values of $\overline{E}$, $\overline{RE}$ and $\overline{F}$ for other methods. We report the results in Table 4.2.

SENet achieved the lowest classification error ($E = 13.69\%$) on the original RAF-DB test set. According to the typical benchmark evaluations of classification algorithms, SENet would be selected as the best performing method on the concerned data. However, the results obtained on the RAF-DB-C and RAF-DB-P test sets gave contrasting insights. We observed a degradation by more than 10% of the error for all the considered methods, as seen in Table 4.2.

On the RAF-DB-C test set, VGG achieved the lowest corruption error ($E_c = 26.38$), while SENet performed worse: $\overline{E} = 1.033$ and $\overline{RE} = 1.089$ indicate that the corruption error and the relative corruption error are 3.3% and 8.9% higher than that of VGG. DenseNet, instead, achieved a lower relative corruption er-

ror ($\overline{RE} = 0.863$). As the $\overline{RE}$ measures the degradation of the performance of a given method on corrupted input with respect to its performance on the original input, we interpret this result as an intrinsic capability of the DenseNet architecture to generalize well with respect to corruptions. We conjecture that the forward connections within a dense block of the DenseNet architecture allow to repeatedly compute feature maps that catch highly complex characteristics of the input and make the network more robust with respect to local changes in the images.

The baseline VGG method has the highest stability when dealing with perturbations, i.e. it achieved the lowest probability of flipping its prediction between consecutive perturbed frames. The normalized flip probability $\overline{F}$ achieved by the other methods is higher than that of VGG by 16.2% for SENet, 35.5% for DenseNet and 78.9% for Xception. Finally, we noted that the method based on LBP and SVM achieved performance not comparable with the CNNs.

The controversial results of this first analysis show that corruptions and perturbations are not negligible aspects and must be treated carefully when deploying a network for emotion recognition.

4.3.2 Results with AutoAugment

We analyzed the impact of the AutoAugment data augmentation strategy on the robustness and stability of the considered methods. We trained them using a set of data augmentation policies that we determined for the RAF-DB data set, according to the procedure proposed by [164]. We compare the results of the considered methods with those obtained by the corresponding variants trained with AutoAugment, which we specify by adding the suffix $|_a$ to the mehod name. We computed the $\overline{E}$, $\overline{RE}$ and $\overline{F}$ using the original methods as the baseline.

As shown in Fig. 4.3, the AutoAugment policies are effective in improving the robustness of the networks to image corruptions. All the considered methods achieved a value of $\overline{E}$ and $\overline{RE}$ lower than

Figure 4.3: Results achieved by the considered methods trained with AutoAugment. For each method, the mean error $\overline{E}$ (left plot), the relative error $\overline{RE}$ (middle plot) and the flip rate $\overline{F}$ (right plot) are computed using the corresponding method without AutoAugment as the baseline, represented as the 1.0 horizontal line.

that of the corresponding baseline. The improvement registered for VGG$|_a$ is relevant. It achieved $\overline{RE} = 0.784$, which measures a reduction of the relative corruption error with respect to the baseline by 21.6%. SENet$|_a$ benefits the most from the use of AutoAugment as it achieved $\overline{E} = 0.894$ and $\overline{RE} = 0.8$. DenseNet$|_a$ and Xception$|_a$ achieved lower results, namely $\overline{E} = 0.910$ and $\overline{RE} = 0.921$ by the former and $\overline{E} = 0.946$ and $\overline{RE} = 0.936$ by the latter. AutoAugment is effective to reduce the impact of corruptions on the performance of the existing methods.

On the RAF-DB-P data set, only SENet$|_a$ benefits from the use of AutoAugment ($\overline{F} = 0.938$), while the other methods do not improve their stability.

4.3.3 Results with anti-aliasing filters

We evaluated the impact that using anti-aliasing filters before the down-sampling layers has on the robustness of the considered methods. We computed the $\overline{E}$, $\overline{RE}$ and $\overline{F}$ for each method using

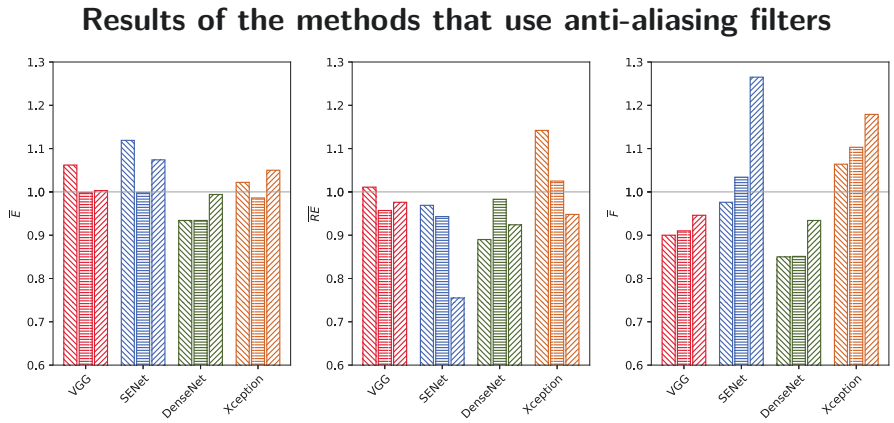

Figure 4.4: Results achieved by the considered architectures when enhanced with anti-aliasing filters. The three bars from each group bar represent the use of rectangular, triangular and binomial filters respectively; lower is better. For each method the mean error $\overline{E}$ (left plot), the relative error $\overline{RE}$ (middle plot) and the flip rate $\overline{F}$ (right plot) are computed using the corresponding method without any anti-aliasing filters as baseline, represented as the 1.0 horizontal line

as the baseline the corresponding original network.

The results that we achieved (see Fig. 4.4) show different effects on different architectures. DenseNet$|_r$ substantially benefits from the use of anti-aliasing filters. Its robustness to corruptions improved, with a reduction of the $\overline{E}$ to 0.934 and of the $\overline{RE}$ to 0.89. The stability to perturbations is also enhanced, as the $\overline{F} = 0.85$ indicates a reduction of the flip probability by 15% with respect to the baseline. VGG$|_r$ achieved $\overline{F} = 0.9$, while the results on corruptions are negligible. For SENet, the impact of the anti-aliasing filters is limited, while for Xception it is pejorative. It is worth noting that the size of the filter that contributes to the best improvement is not the same for all the methods. The use of the anti-aliasing filter is an effective strategy for improving the performance of DenseNet against corruptions and perturbations, and it is also an effective technique for increasing the stability of VGG.

4.3.4 Results with combined anti-aliasing filters and AutoAugment

The results obtained applying AutoAugment and the anti-aliasing filters showed that these modifications allow to improve the robustness of the considered methods. AutoAugment improves the robustness to corruptions, while the anti-aliasing filter is more effective to achieve better stability against perturbations.

We applied these two techniques together, and computed the $\overline{E}$, $\overline{RE}$ and $\overline{F}$ for each method using as the baseline the corresponding method without modifications. In Fig. 4.5, we report the obtained results.

DenseNet$|_{t,a}$ achieved the best overall performance in terms of classification error ($E = 13.59$), as well as robustness and generalization to the corruptions in the RAF-DB-C data set($\overline{E} = 0.887$ and $\overline{RE} = 0.676$). Its stability to perturbations also improved ($\overline{F} = 0.894$). Xception$|_{t,a}$ achieved a better classification error ($E = 15.88$) and robustness to corruptions ($\overline{E} = 1.062$ and $\overline{RE} = 0.848$) with respect to the corresponding baseline. On the perturbations in the RAF-DB-P data set, instead, the modifica-

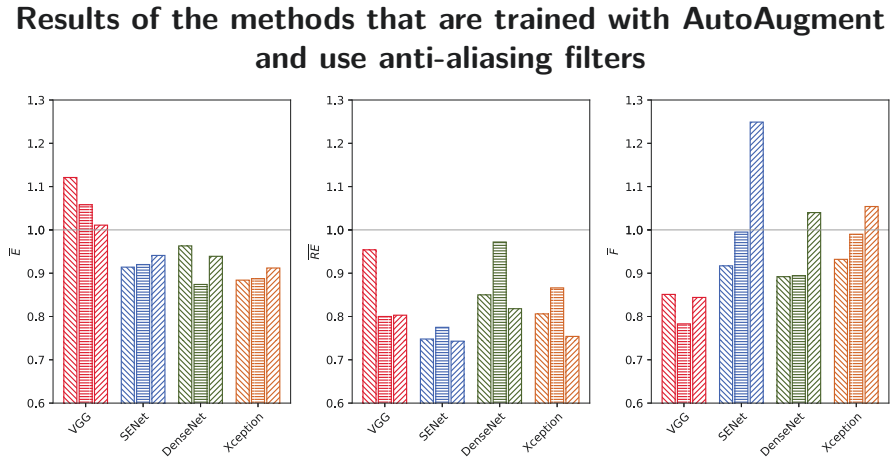

Figure 4.5: Results achieved by the considered architectures when enhanced with anti-aliasing filters and trained with AutoAugment. The three bars from each group bar represent the use of rectangular, triangular and binomial filters respectively; lower is better. For each method the mean error $\overline{E}$ (left plot), the relative error $\overline{RE}$ (middle plot) and the flip rate $\overline{F}$ (right plot) are computed using as baseline the corresponding method trained without AutoAugment and without any anti-aliasing filters, and represented as the 1.0 horizontal line.

tions to the architecture and training data worsen the stability of the method ($\overline{F} = 1.576$). This is attributable to the fact that this network requires larger input images, leading to sub-optimal performance for the problem at hand, possibly caused by the need to scale up the small images from the RAF-DB data set. For SENet, we measured an improvement of robustness to corruptions ($\overline{RE} = 0.782$ and $\overline{E} = 0.932$ achieved by SENet$|_{r,a}$), but not on the perturbations. Conversely, VGG benefited from these modifications and achieved better stability ($\overline{F} = 0.783$).

The combined use of AutoAugment and the anti-aliasing filters allows to improve the robustness of the considered models. It has a substantial impact on DenseNet, which achieved the best performance on original and corrupted images, and for on VGG that showed the best stability with respect to perturbations.

4.3.5 Robustness, generalization and stability

In Fig. 4.6, we show a scatter plot of the mean corruption error $\overline{E}$ (x-axis) and the relative corruption error $\overline{RE}$ (y-axis) achieved by the considered methods. The axis direction is inverted for visualization purposes. In order to directly compare the performance of different networks, we normalized the results reported in Fig. 4.6 using as common baseline the results of VGG.

The methods in the top-right quadrant (green region) perform better than the baseline in terms of robustness (lower $\overline{E}$) and generalization (lower $\overline{RE}$) to corruptions. It is worth pointing out that the generalization is intended as the capability of keeping the gap between the classification error on original and corrupted data very small. The points in the top-left and bottom-right quadrants (yellow regions) correspond to the methods achieving an improvement either of the $\overline{E}$ or the $\overline{RE}$ with respect to the baseline. The bottom-left quadrant (red area), instead, collects the results of the methods that perform less than the baseline on corrupted data.

The methods based on DenseNet (green markers) achieved the best performance on the RAF-DB-C data set. SENet (blue markers) benefits from the use of the AutoAugment augmentation poli-

cies and anti-aliasing filters in its architecture, and improves on the performance of its original version. The Xception-based methods (dark-yellow points) show low robustness to corruptions: their results are all located in the left quadrants of Fig. 4.6. VGG$|_a$ is the only VGG-based method (red points) that achieved better robustness and generalization to corruptions with respect the baseline.

We compared the performance of the considered methods by jointly evaluating their robustness to corruptions (i.e. the mean corruption error $\overline{E}$) and stability with respect to perturbations (i.e. the flip rate $\overline{F}$) and show a scatter plot of the results in Fig. 4.6.

The points in the upper quadrants (top-left yellow and top-right green quadrants) correspond to the methods that achieved a better stability against perturbations than the baseline. The methods based on VGG (red markers) achieved the best stability against perturbations, as they are located in the upper quadrants of the plot; however, they are less robust to corruptions than the other methods. The methods based on SENet (blue markers) and DenseNet (green markers) achieved good robustness ($\overline{E} = 0.932$ and $\overline{E} = 0.887$) to corruptions but are less stable to perturbations ($\overline{F} = 1.055$ and $\overline{F} = 1.144$, respectively). Xception-based methods (dark-yellow markers) achieved results worse than those of the other methods. Their results indeed place in the bottom-left quadrant of the plot in Fig. 4.6.

The modifications to the training data and the use of anti-aliasing filters in the network architectures generally contributed to an increase of robustness to corruptions and stability to perturbations of the methods with respect to their original implementation. However, none of the modified methods achieved at the same time higher robustness and stability than the VGG, i.e. there are no points in the green quadrant in Fig.4.6. VGG methods perform more stable classification over perturbed frame sequences and also exploit the data- and architecture-related improvements better than other existing methods.

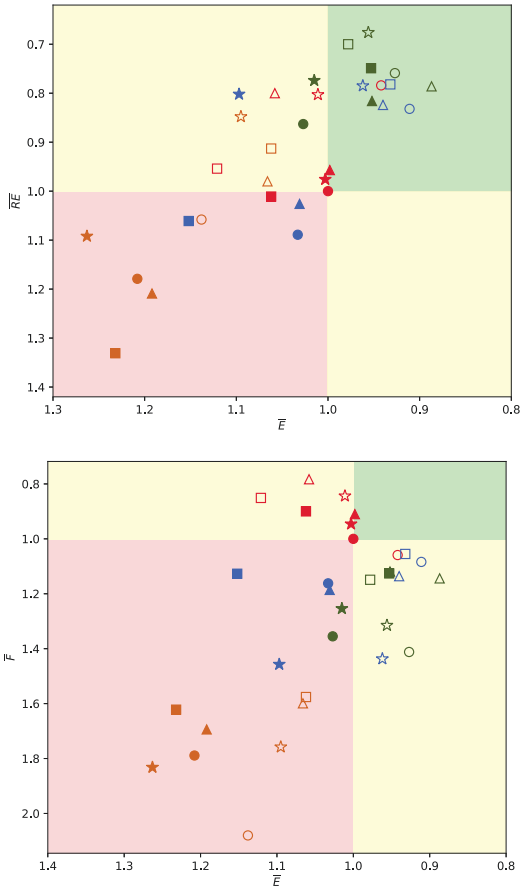

Figure 4.6: (a, top) Results of the evaluation of the robustness and generalization of the considered methods with respect to corruptions, in terms of mean error $\overline{E}$ and relative error $\overline{RE}$.
(b, bottom) Results of the evaluation of the robustness to corruptions and the stability to perturbations of the considered methods, in terms of mean error $\overline{E}$ and flip rate $\overline{F}$.
The direction of the axes is inverted so that the points in the top right quadrant (green region) correspond to the results of methods that improve their performance w.r.t. the baseline, namely VGG. Different colors represent different network architectures: red is VGG, blue is SENet, green is DenseNet and yellow is Xception.
The ● marker refers at the original methods. The ■, ▲ and ★ markers indicate the methods with anti-aliasing filters of type rectangular, triangular and binomial, respectively. The empty markers represent methods trained with the AutoAugment strategy.

4.3.6 Robustness to categories of corruption and perturbation

We carried out an analysis of the performance of the considered methods on specific categories of corruptions and perturbations. In Tables 4.3, 4.4 and 4.5, we report the $\overline{E}$, $\overline{RE}$ and $\overline{F}$ values achieved for each corruption and perturbation category respectively.

The combination of AutoAugment and anti-aliasing filters improved the robustness of the considered methods to all the categories of corruptions, except for Xception that obtained results comparable with those of the VGG baseline. The methods based on DenseNet achieved a considerable improvement of the corruption error $\overline{E}$ and of the relative corruption error $\overline{RE}$ with respect to the baseline on all the categories of corruptions. The improvement is, however, less evident on the corruptions of type *blur*; we observed that these corruptions are the most challenging for all the considered methods. This is attributable to the fact that convolutional networks tend to learn distinctive and discriminant features at high spatial frequencies, while blur preserves lower frequency patterns which are less relevant for classification [139].

Furthermore, DenseNet methods achieved the highest results in terms of stability against noise perturbations, obtaining a reduction of the flip rate $\overline{F}$ up to more than 40% with respect to the baseline. Most of the improvement is attributable to the use of the anti-aliasing filters, while training the methods with AutoAugment has lower impact. In scenarios in which the recorded images are affected by noise, the DenseNet network architecture with anti-aliasing filters can be deployed to ensure robust recognition performance.

The impact of the anti-aliasing filters on the stability of the VGG methods against perturbations is positive and evident for the blur, noise and transformation types of perturbation. These methods achieved a reduction of the flip rate $\overline{F}$ between about 10% and 25% with respect to the baseline. Digital perturbations, instead, are more challenging. The VGG-based methods

also achieved good robustness with respect to noise, digital and mixed corruptions.

The SENet methods benefited the most from the use of AutoAugment and the anti-aliasing filters to improve the performance of the original network model with respect to corruptions of the type *noise*. The values of the errors $\overline{E}$ and $\overline{RE}$ improved respectively by about 30% and 45% with respect to the baseline. However, the results of SENet methods are negatively influenced by the other types of perturbations, while showing slight improvements of robustness with respect to the other categories of corruption.

For the methods based on Xception, instead, the use of AutoAugment and the anti-aliasing filters did not contribute to substantially improve their robustness to corruptions and perturbations, apart a small reduction of the relative corruption error $\overline{RE}$ on the corruptions of type digital.

Table 4.3: Results on the RAF-DB-C data set in terms of $\overline{E}$. Green color gradients indicate an improvement w.r.t. the baseline, while yellow and red color gradients indicate comparable or lower results than the baseline.

net	Corruption	Models							
		net	**net**$\|_r$	**net**$\|_t$	**net**$\|_b$	**net**$\|_a$	**net**$\|_{r,a}$	**net**$\|_{t,a}$	**net**$\|_{b,a}$
VGG	blur	1.000	1.049	0.998	0.901	1.092	1.248	1.196	1.133
	noise	1.000	1.254	1.188	1.189	0.943	1.143	0.938	0.889
	digital	1.000	0.999	0.951	1.000	0.875	1.039	1.019	1.015
	mixed	1.000	1.083	0.990	1.014	0.916	1.125	1.050	0.956
SENet	blur	1.078	1.151	1.019	1.122	0.932	0.966	1.036	0.951
	noise	1.218	1.290	1.273	0.921	0.720	0.725	0.729	0.903
	digital	0.967	1.103	0.993	1.178	0.933	0.957	0.929	0.971
	mixed	1.017	1.166	0.997	1.033	0.940	0.952	0.963	0.981
DenseNet	blur	0.933	0.939	0.984	1.012	0.973	1.036	0.996	0.979
	noise	1.227	0.975	0.965	1.032	0.909	0.821	0.800	0.867
	digital	1.038	0.987	0.942	1.045	0.932	1.017	0.868	0.990
	mixed	1.007	0.909	0.936	0.966	0.890	0.940	0.862	0.927
Xception	blur	1.134	1.157	1.120	1.177	1.076	1.112	1.052	1.140
	noise	1.203	1.404	1.234	1.477	1.105	0.962	1.157	1.149
	digital	1.173	1.183	1.155	1.250	1.140	1.026	1.032	1.052
	mixed	1.318	1.293	1.284	1.264	1.197	1.110	1.089	1.097

Table 4.4: Results on the RAF-DB-C data set in terms of $\overline{RE}$. Green color gradients indicate an improvement w.r.t. the baseline, while yellow and red color gradients indicate comparable or lower results than the baseline.

net	Corruption	Models							
		net	**net**$\|_r$	**net**$\|_t$	**net**$\|_b$	**net**$\|_a$	**net**$\|_{r,a}$	**net**$\|_{t,a}$	**net**$\|_{b,a}$
VGG	blur	1.000	1.035	1.004	0.795	1.169	1.293	1.186	1.126
	noise	1.000	1.352	1.285	1.280	0.906	1.116	0.798	0.766
	digital	1.000	0.840	0.841	0.982	0.577	0.702	0.607	0.747
	mixed	1.000	1.094	0.950	0.991	0.717	0.970	0.761	0.637
SENet	blur	1.203	1.105	1.040	0.947	0.885	0.890	1.056	0.819
	noise	1.348	1.343	1.409	0.735	0.588	0.567	0.584	0.812
	digital	0.954	0.892	0.945	0.878	0.837	0.760	0.727	0.731
	mixed	1.085	1.149	0.975	0.606	0.882	0.811	0.869	0.822
DenseNet	blur	0.724	0.785	0.938	0.850	0.904	0.893	1.059	0.808
	noise	1.283	0.927	0.935	0.969	0.846	0.650	0.724	0.729
	digital	0.844	0.748	0.730	0.757	0.728	0.681	0.705	0.656
	mixed	0.832	0.651	0.791	0.658	0.651	0.592	0.708	0.579
Xception	blur	1.036	1.165	1.061	0.988	0.943	1.068	0.992	1.046
	noise	1.197	1.531	1.267	1.543	1.056	0.863	1.180	1.107
	digital	1.068	1.244	1.129	1.015	1.041	0.765	0.881	0.672
	mixed	1.444	1.507	1.414	1.101	1.174	1.017	1.028	0.834

Table 4.5: Results on the RAF-DB-P data set in terms of $\overline{F}$. Green color gradients indicate an improvement w.r.t. the baseline, while yellow and red color gradients indicate comparable or lower results than the baseline.

net	**Perturbation**	**Models**							
		net	**net**$\|_r$	**net**$\|_t$	**net**$\|_b$	**net**$\|_a$	**net**$\|_{r,a}$	**net**$\|_{t,a}$	**net**$\|_{b,a}$
VGG	blur	1.000	0.845	0.881	0.917	1.107	0.881	0.845	0.881
	noise	1.000	1.247	1.186	1.200	0.691	0.595	0.530	0.778
	digital	1.000	0.917	0.958	1.000	1.197	1.087	0.981	0.955
	transformation	1.000	0.746	0.762	0.806	1.150	0.845	0.779	0.802
SENet	blur	1.036	1.000	1.238	1.310	1.036	1.000	1.238	1.310
	noise	1.241	1.381	1.180	1.040	0.854	1.017	0.932	0.938
	digital	1.023	1.068	1.155	1.481	1.023	1.068	1.155	1.481
	transformation	1.255	1.094	1.177	1.728	1.255	1.094	1.177	1.728
DenseNet	blur	1.500	1.226	1.226	1.345	1.500	1.226	1.226	1.345
	noise	0.577	0.569	0.589	0.572	0.865	0.688	0.714	0.878
	digital	1.500	1.330	1.284	1.417	1.500	1.330	1.284	1.417
	transformation	1.598	1.252	1.248	1.468	1.598	1.252	1.248	1.468
Xception	blur	1.810	1.500	1.738	1.845	1.810	1.655	1.571	1.810
	noise	0.796	1.687	1.706	1.767	1.345	0.992	1.308	1.224
	digital	1.674	1.523	1.477	1.610	1.777	1.496	1.432	1.561
	transformation	2.333	1.699	1.775	1.968	2.734	1.868	1.843	2.098

Chapter 5

A distillation approach for age estimation

5.1 Background

As previously discussed in Section 1.3.1.2, we find two main limitations in the state of the art, which are, in a way: the absence of a large and reliably annotated dataset for age recognition and the absence of a handy procedure for training effective and efficient methods for age estimation. In this chapter, we propose the application of a tailored knowledge distillation approach to overcome those limitations.

Knowledge distillation [69] is a technique used to train small, efficient convolutional neural networks with reduced need of resources (i.e. processing time, memory, and so on) transferring the knowledge learned by a more complex model. The method, in its general form, consists in the extraction of the class probability vectors produced by a large model, also called *teacher*, and the adoption of these vectors as a target for training the smaller model, known as *student*. An alternative, naive approach, would be to train the small network directly on the same dataset that was used to train the large model; however, it has been demonstrated that for complex problems the student network can achieve higher accuracy when trained with knowledge distillation than if it is directly trained with the labels of the original dataset [165]. The intuition behind distillation, i.e. the supposed advantage, is that the larger *teacher* model is able to better fit the dataset and encode its peculiarity due to its higher representative power, in a way that the smaller model just could not; the *student* model may be however able to leverage the knowledge that has been pre-digested and encoded into a simpler annotation, namely the output probability vectors of the teacher.

Recent literature demonstrated the effectiveness of knowledge distillation in various pattern recognition tasks, even related to face analysis. In [69] Hinton et al. showed that knowledge distillation allows a 2% accuracy improvement of a student model for speech recognition with respect to one trained using the original labels of the dataset; with this technique the simple student model performs similarly to the much more elaborate teacher model. In

[166] the authors demonstrated that a network trained with the distillation approach makes a CNN more robust to perturbations by a considerable amount; this effect is explained considering that the *student* network sees its training input in a clearer way and is able to organize its weights around a more representative manifold. In [167] Low et al. applied distillation on selected, most informative faces, to train a face recognition network that achieves good performance on images with low resolution. In [168] the authors trained different convolutional neural networks (CNNs) for facial expression recognition with incomplete labeling; they find that the *student* model often outperforms the *teacher* on the considered task.

The method proposed here is a variant of the standard knowledge distillation technique. We apply it to the problem of age estimation, to address its peculiar limitations, namely the absence of a large dataset with reliable annotation and the lack of a handy procedure allowing to train effective and efficient CNNs for age estimation applicable in real scenarios. We take the popular large scale dataset VGGFace2 [32], which is not natively annotated with age labels, and we run the most accurate method in literature, winner of the LAP 2016 competition; this method consists of a large and complex ensemble of 14 CNNs that analyze 8 versions of each input image [57], and is trained on *IMDB-Wiki-cleaned*. We use the resulting predictions as target labels to train a variety of different CNN architectures, requiring about 15 times less operations. Therefore, we obtain the two-fold advantage of having a large dataset annotated for age estimation enabling the possibility to perform a standard (and fast) training procedure of smaller student models.

We show that our approach allows to achieve state-of-the-art results on multiple relevant public benchmarks (including the LAP dataset) with much simpler and faster methods composed only by a single CNN, outperforming other complex methods, that tipically employ large ensembles. We show that using our own cleaned version of IMDB-WIKI as training dataset, the accuracies reached by the same CNNs are much lower, thus proving the effectiveness

of the proposed approach with respect to the traditional procedure. We also show that the student models are even able to outperform the teacher in presence of the strong image corruptions described in Chapter 4.

5.2 Methodology

Our aim is to train standard and efficient CNNs that are able to perform accurate age estimation. Reliable datasets for this task are not big enough to effectively train a deep neural network, while large datasets such as *IMDB-Wiki* contain spurious annotation that will cause inefficiencies in the training process [147]. As described in Section 5.1, previous work focused on mitigating the effect of annotation errors and getting the most value out of small datasets with the use of ensembling techniques. However, we aim to build a dataset that is large, depicting a variety of conditions, identities and faces, reliably annotated and without requiring extensive human effort for annotation.

In our approach, we achieve this aim by knowledge distillation. Indeed, we automatically annotate VGG-Face2 Dataset by means of a teacher method, namely a pre-trained ensemble of CNNs for age estimation. We call the dataset VGG-Face2 Mivia Age (VMAGE). We use this dataset to train a variety of simpler student models, with the aim to achieve more or less the same accuracy of the teacher model, but with a substantially lower computational burden.

By design, student models will have some advantages over the teacher, namely smaller size and lower inference time. The authors of the teacher method report an average execution time of 6.3 seconds per image, while each one of the architectures that we employ can be executed in a fraction of a second even on embedded systems with low processing resources [131].

We prove that using the generated VMAGE dataset in the task of age estimation implies a significant advantage over the baseline procedure, i.e. training on the standard *IMDB-Wiki* dataset. The

experimental analysis of the student models in Section 5.4 reveals that the proposed procedure allows to achieve state of the art results for all the most widely used test benchmarks, overcoming the results of most methods while cutting down on complexity.

The VMAGE dataset provides an estimation of the apparent age for each face. The teacher method [169] that we used is the best performing method in the state of the art according to the ChaLearn LAP 2016 benchmark. We believe that this benchmark provides an accurate estimation of the performance of different methods in realistic scenarios due to two main reasons. Firstly, the annotation is obtained by crowdsourcing, so that an accurate estimation of the apparent age is used as target rather than the real age: this is arguably an advantage for developing systems that aim to replicate the human ability to estimate age from the appearance. Secondly, the benchmark uses a metric which weights the errors to match the human perception. We give more detail about the LAP benchmark and its metrics in Section 5.3.3.

5.2.1 Teacher method

In this section we describe how the teacher method works. More details can be found on the original paper [169].

The "Head Hunter" face detector [36] is applied to the input image to determine the position of the face; it is then aligned using a similarity transform based on the Multi-view Facial Landmark Detector [170] and resized to 224x224 pixels. A total of 8 variants are obtained from each input sample: the original, the horizontal mirror, two rotated versions ($\pm 5°$), two horizontally shifted versions ($\pm 5\%$) and two scaled versions ($\pm 5\%$). Each of these images is processed by the classification core of the method.

The classification core is composed by an ensemble of 14 CNN models; each model is based on the VGG-16 architecture, 3 are trained to recognize age in children (0-12 years old) while the others are trained for general age estimation (0-99). Each model outputs a vector of 100 age probabilities and a soft voting rule is used to determine the consensus between the 11 generic models

executed on the 8 variants of the image. If the age is higher than 12, the result is determined by the 11 generic models; otherwise, the 3 children models are executed on the 8 variants and the 3×8 vectors of 13 age probabilities are aggregated to produce the final apparent age estimation.

Each model in the classification core was trained by the authors in multiple steps using the well-known fine-tuning technique. The VGG-16 architecture is pre-trained on the VGG-Face dataset for the task of identity recognition and then it is fine-tuned on the *IMDB-Wiki* dataset; each of the 11 general age estimation models is then fine-tuned again using a different 11-fold partition of the LAP training set using distribution label encoding as loss function. The children-specialized models are trained starting from the *IMDB-Wiki* model described above, then fine-tuned again on a children-only private dataset and finally fine-tuned on the children images from the LAP training and validation set using 0/1 classification encoding. Three different checkpoints are chosen to be the 3 members of the children-specialized ensemble. All CNNs are optimized using gradient descent with momentum 0.9 and batch size of 32. Each image is repeated 5 times in a data-augmentation fashion, using horizontal mirroring, random rotation, random shift and random scale.

The teacher method achieved an impressive 0.2433 ϵ-error in the ChaLearn LAP 2016 competition, winning by a large margin on the second classified. However, its accuracy is payed in terms of processing time, which is 6.3 seconds for each image.

5.2.2 The VMAGE dataset

The VMAGE dataset is the intermediate product of the proposed knowledge distillation process. We create it in the first step of our procedure in order to transfer the knowledge of the teacher to the student models in the training phase.

The dataset is built upon the image data collected for the task of image identification in the VGG-Face2 dataset [32]. It includes 9,116 identities among actors, athletes and other public figures

Table 5.1: Absolute distribution of the samples in VMAGE, IMDB-Wiki, LFW+, LAP 2016 and Adience within the age groups 0-15, 16-25, 26-35, 36-45, 46-60 and 61-100.

Age	**# of samples**				
	VMAGE	**IMDB-Wiki**	**LFW+**	**LAP**	**Adience**
0-15	18,864	7,813	52	887	6,983
16-25	451,999	50,216	372	1,829	1,655
26-35	1,342,493	103,240	1,855	2,376	4,950
36-45	702,677	86,688	1,822	2,350	2,350
46-60	589,558	58,087	3,661	943	830
61-100	131,401	21,566	2,385	387	875

with a total count of 3.3 million faces.

In Table 5.1 we give details about the composition of the dataset, while the histogram in Figure 5.1 allows to note that the distribution of labels by age group is similar to other datasets from literature. In particular ages in the range 25-35 are the most represented, while there are few elders (60-100) and even fewer children (ages 0-15). LAP 2016 and (especially) Adience are exceptions to this rule since they focus on those less represented classes; LFW+ is skewed towards older ages, with most faces in the 45-60 and 60-100 ranges.

We notice that, in absolute numbers, the VMAGE dataset is larger than every other dataset, and this is still true across every considered age group, even the least represented one. Due to the class imbalance, we observe that the VMAGE dataset is most useful as a pre-training tool, while countermeasures can be taken if the target application includes children; fine-tuning on a more balanced dataset is a suitable strategy for fixing the imbalance problem, as we show in our experimentation.

The VMAGE dataset includes age labels for the images that were artificially annotated by the teacher ensemble. The implementation of the teacher method that we used is based on the Caffe framework [171] and was kindly provided by the authors under the GNU GPL-3.0 license. The execution took 962 GPU

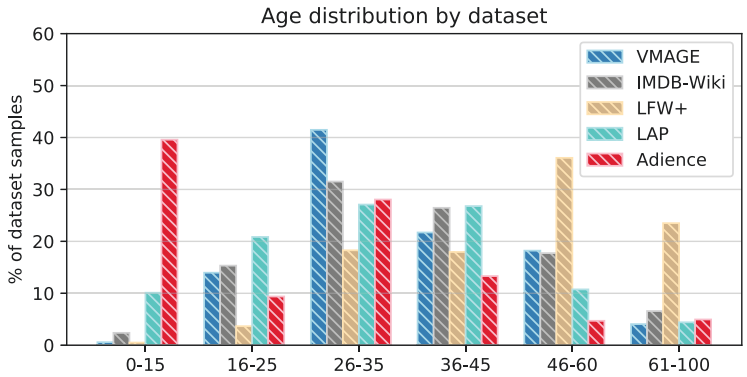

Figure 5.1: Relative distribution of the samples in VMAGE, IMDB-Wiki, LFW+, LAP 2016 and Adience within the age groups 0-15, 16-25, 26-35, 36-45, 46-60 and 61-100.

hours for 3.3 million images and was performed over 2 weeks using 3 NVIDIA Titan X GPUs. We make the labels for the VMAGE dataset available for research purposes[1].

For each face the exact predicted age is given, which is the most versatile approach. Datasets that are annotated with an age class (e.g. child, adult, elder) are useful only if the problem is posed as a classification task, with the exact same class boundaries. We argue that this position does not fit all the possible applications and so we report the exact predicted age, allowing for the reuse of the dataset in different types of contexts as shown in Section 5.3.

We retained the intermediate predictions by the teacher ensemble; the agreement between ensemble members may be used a metric of difficulty of each sample, reproducing a multi-annotator labeling scheme like the one used for the construction of the LAP dataset, allowing for additional considerations. Although we did not use those in this work, we believe that this information will be useful for future work on the topic and make them publicly available.

[1] https://mivia.unisa.it/datasets/vmage

5.3 Experimental framework

In order to evaluate the effectiveness of the proposed approach, we train a number of well known CNN architectures, representative of different families among the most commonly used for face analysis. As a baseline for comparison we train the same architectures using a prominent strategy from the state of the art, namely we apply a data-cleaning procedure to the large scale dataset IMDB-Wiki and use the resulting corpus. In Section 5.3.1 we describe the considered architectures, while in Section 5.4 we show how the CNNs trained with our knowledge distillation methodology consistently outperform the corresponding neural networks trained directly on the IMDB-Wiki cleaned corpus.

In order to compare our results with the ones published in the literature and the teacher network, we evaluate our method on the LAP 2016 (a.k.a. APPA-REAL) dataset [63]. In addition, we evaluate the accuracy of the considered CNNs over LFW+ [172] and Adience [148]. The datasets have different characteristics in terms of age distribution, face appearance and a different evaluation protocol. We will describe all these datasets and protocols in detail in Subsection 5.3.3.

Finally we will evaluate the robustness of our method to corruptions of the input images: it has been shown that images acquired in real operating environments exhibit a significant amount of diverse types of corruptions, such as gaussian noise, motion blur, compression artifacts and so on. We will discuss these corruptions in Section 5.3.4 and we will show in Section 5.4 that our training procedure allows the student networks to overcome the accuracy of the teacher in such challenging conditions.

5.3.1 CNN architectures

In our analysis, we selected 4 different convolutional neural networks, widely adopted in several face analysis tasks: VGG, SENet, DenseNet and MobileNet, each with different characteristics.

VGG, introduced in [132], is the family of CNNs most widely

used for face analysis tasks, especially for the availability of a version of VGG-16, namely VGG-Face [144], pre-trained for face recognition by using the VGG-Face2 dataset. Such network, fine tuned on specific datasets, achieved state of the art performance in gender, ethnicity and emotion recognition. The peculiarity of this CNN architecture is the adoption of 3×3 filters to build larger filters (5×5) in order to obtain a more effective receptive field while reducing the number of weights and the cost of adding convolutional layers. This choice demonstrated to give VGG the capability to achieve good generalization even when the dataset is quite small. In this paper, we use the VGG-16 version, which consists of 13 convolutional and 3 fully connected layers, resulting in 138M weights and more than 15G of operations with 224×224 input size.

SENet, proposed in [158], is based on the well known ResNet-50 architecture [130], with the addition of the *Squeeze and Excitation* modules. The ResNet architecture has been designed with the idea to increase the number of layers for achieving higher accuracy. Therefore, a shortcut module learns the residual mapping to solve the problem of vanishing gradients happening in very deep networks (especially in the earlier layers during backpropagation). In addition, it adopts the bottleneck approach by using 1×1 filters to capture cross-channel correlation and reduce the number of weights. The original ResNet-50 consists of 1 convolutional layer, 16 shortcut modules and 1 fully connected layer, resulting in 25.5M weights and 3.9G operations with 224×224 input size. The addition of the modern *Squeeze and Excitation* modules, namely a particular type of depthwise convolution with dynamic weights, allows to learn a function for giving more importance to specific channels of the input feature map by reducing the magnitude of the activations in the other channels. This choice demonstrated to increase the accuracy in various computer vision tasks [158].

DenseNet, proposed in [160], is a family of CNNs designed according to the experimental evidence that a CNN can be more accurate and efficient to train if it contains direct connections between input and output layers. In DenseNet, each layer is con-

nected to every other layer (dense blocks) to favour the propagation and the reuse of the feature maps; this concept, widely investigated in recent years, is also known as *feature map aggregation*. To solve the problem that feature maps with different spatial resolution can not be aggregated, DenseNet complements the use of dense blocks with the adoption of transition layers, which normalize the size of the feature maps computed by the different layers through specific pooling operations. In this paper, we use the DenseNet-121 version, resulting in 7M weights and about 3G operations with 224×224 input size.

MobileNet, described in [173], is a family of CNNs among the most efficient available in the literature, designed for running on board of mobile and embedded devices. It includes the more modern devices for reducing the number of weights and operations while holding a high accuracy, namely residual blocks, depthwise convolutions followed by pointwise convolutions and bottleneck layers. In this paper, we use the newest MobileNet V3 Large and Small versions [129], which also include squeeze and excitation modules, swish nonlinearities and hard sigmoid and are globally optimized through the NetAdapt algorithm. MobileNet-Large requires 5.4M weights and around 219M operations with 224×224 input size, while MobileNet-Small 2.5M weights and about 54M operations with 96×96 input size. Hereinafter, we will refer to these CNNs with the names *MN3-Large* and *MN3-Small*.

5.3.2 Training

In our experiments, we train all the architectures starting from the ImageNet pre-trained weights. Using pre-trained weights from a large-scale generic dataset is a common strategy in many applications of deep learning, since it allows to alleviate overfitting and improve convergence [31].

In our training pipeline, as a first step the bounding rectangle of the face is localized; for face detection and localization we use a lightweight face detector based on the SSD framework [38]. The face rectangle is expanded to have a square aspect ratio and

Table 5.2: Augmentation policies and parameters used for training. Parameters are randomly computed using the bounded normal distribution $\overline{\mathcal{N}}$, defined as follows
$\overline{\mathcal{N}}(\mu, \sigma) = \min(\mu + 2\sigma, \max(\mu - 2\sigma, \mathcal{N}(\mu, \sigma)))$.

Policy	**Parameter**	**Value**
Crop	Δ_x, Δ_y, Δ_w, Δ_h	$\Delta_x, \Delta_y \sim \overline{\mathcal{N}}(-\frac{\sigma}{10}, \sigma)$ $\Delta_w, \Delta_h \sim \overline{\mathcal{N}}(\frac{\sigma}{4}, 2\sigma)$
Horiz. Flip	probability P	0.5
Rotation	degrees q	$\sim \overline{\mathcal{N}}(0, 5)$
Skew	s_x, s_y	$\sim \lvert\overline{\mathcal{N}}(0, 0.05)\rvert$
Brightness	b	$\sim \overline{\mathcal{N}}(0, 24)$
Contrast	c	$\sim \overline{\mathcal{N}}(1, 0.5)$

the image is cropped and resampled with the bilinear algorithm to match the input size of the network. As a final step, from each image we subtract the average value computed separately for each channel on the VGG-Face dataset by the authors [144]; this step allows for the input distribution to be 0-centered on average, allowing to take full advantage of the ReLu non linearity and achieve faster convergence.

During the training process every sample image is perturbed using one of more random augmentation policies. The policies include random crop and horizontal flip, rotation, skew, brightness and contrast. The parameters for these transformations are chosen randomly according to the distributions reported in Table 5.2; we chose the parameters empirically, ensuring that the augmented images are representative for the dataset.

The training is carried out for 70 epochs and the SGD optimizer is used. The learning rate is initialized to 0.005 and reduced with a factor of 0.2 every 20 epochs. For the VGG-16 network we use 0.00005 as initial learning rate, since it needs lower learning rates for ensuring convergence; this is due to the architectural peculiarities of this network, namely the absence of batch normal-

ization.

When needed, the CNNs are possibly fine-tuned according to the official evaluation protocol for each considered benchmark, as explained in the following Section 5.3.3.

5.3.3 Datasets

LFW+ [172] is the dataset that we chose for testing the performance of the student networks in the task of real age estimation. It consists of 15,699 face images belonging to 8,000 different subjects. The dataset is not partitioned in training and test set, so we decided to use the whole dataset for our experiments without fine tuning. This procedure of testing without fine tuning has been used on the same LFW+ dataset in different tasks such as gender recognition [57, 174]; it is called *cross-dataset evaluation* and allows to assess the generalizability of the features that can be learned through the training dataset.

The evaluation metric we adopt for this dataset is the mean absolute error (MAE). Let us denote with a_i the age predicted on the i-th sample and with r_i the corresponding real label, the MAE is the average error over the K test samples. Being $e_i = |a_i - r_i|$ the error on the i-th sample:

$$MAE = \frac{\sum_{i=1}^{K} e_i}{K} \tag{5.1}$$

Testing without fine tuning allows us to investigate the cross-dataset generalization capability of the networks.

LAP 2016 a.k.a. APPA-REAL [63] is a dataset for estimating the apparent age of people, whose age annotations have been collected through crowdsourcing. It contains 7,591 samples, already divided in training (4,113), validation (1,500) and test (1,978) sets. The experimental protocol requires a standard training or fine tuning of the neural networks by using the proposed partition. This dataset contains a small number of samples, but it is considered one of the most challenging in terms of face variations and reliable regarding the age annotations. To weight differently

the errors done by the neural networks on images annotated with difficulties also by humans, the organizers of the Chalearn Looking at People challenge [175, 63] designed a specific metric for apparent age estimation, namely the ϵ-*error*. Being m_i and v_i^2 the mean and the variance of the distribution of the predictions a_i done by the annotators for the i-th sample, the estimation error ϵ_i is computed as:

$$\epsilon_i = 1 - e^{-\frac{(a_i - m_i)^2}{2{v_i}^2}} \tag{5.2}$$

According to this metric, the error on the i-th sample is normalized by the corresponding variance, in order to penalize less the errors done on samples with high variance. The ϵ-error is finally computed as the mean of the ϵ_i over the K samples of the test set.

Being the dataset already divided in training, validation and test set, we perform the fine tuning of our CNNs with the same procedure described in Section 5.3.2, by starting from the weights pre-trained on VMAGE of IMDB-Wiki.

Adience [148] is a dataset that we use for age group classification. It is very challenging, produced by automatically extracting images from about 200 Flickr albums, thus collected in uncontrolled conditions and including variations in pose, lighting and image quality. The whole dataset is composed by $26,580$ face images, of which only about one half are almost frontal. A subset of the face images ($17,643$) is annotated with 8 unbalanced age categories: 0-2, 4-6, 8-13, 15-20, 25-32, 38-43, 48-53, 60+. Adience is probably the dataset containing more children images in percentage than the other benchmarks publicly available. The standard experimental protocol is a 5-fold cross validation, with the folds already provided by the authors. Being a classification problem, the performance of the neural networks tested on this dataset are evaluated in terms of accuracy, namely the ratio between the number of correct classifications and the total number of samples. Since the dataset is very challenging, the protocol requires the computation of two variants: the *top-1* and the *1-off*. For computing

the accuracy top-1, a classification is considered correct whether it corresponds to the true age group; as for accuracy 1-off, the evaluation metric considers correct also the predictions for age groups which are adjacent to the one in groundtruth.

Since the benchmark protocol recommends fine tuning on predefined folds, we fine tune our networks using the procedure explained in Section 5.3.2, except that the starting learning rates are 10 times smaller than the ones used for pre-training. To choose the parameters, we ran a first experiment in which we trained on 3 folds and use the 4th for validation for 70 epochs, while the fifth was never used in the training procedure and was saved for testing; with this procedure we established that the optimal number of epochs was about 35 for all the models. Following the approach taken by our predecessors [176], we train our final fine tuned models on 4 folds for 35 epochs and test on the fifth. Intuitively, given the small size of the Adience dataset we may assume that training on 4 folds will be significantly advantageous over training on 3 folds and using the fourth for validation. Experimental results confirm this intuition, so we report in Section 5.4 the results achieved by the models trained on 4 folds.

Since our networks are pre-trained as regressors, we need a small architectural adjustment for our fine-tuned networks: we remove the last fully connected layer with its one neuron that predict the age and replace it with a fully connected layer with 8 neurons (one for each age group) and add softmax activation. This means that we explicitly convert the network into a classifier and optimize that specifically. All the layers of the network are fine tuned, since we have empirically found this approach to be more effective with respect to training only the topmost layers.

5.3.4 Corruptions

Recent studies [135] demonstrate that the modern convolutional neural networks suffer a drop of the accuracy when the input images are affected by strong corruptions, which are common in real environments. Applications of age estimation such as digital sig-

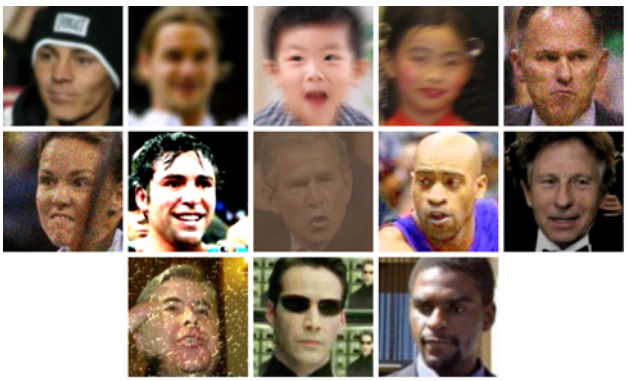

Figure 5.2: A collection of 13 samples from the LFW+C dataset, each of them perturbed with a different kind of corruption. More details about the corruption categories, their severity and their mathematical definition are reported in Section 4.2.1.1.

nage, access control and social robotics require the use of a network that is robust to these perturbations. In [166] it was shown that a student network trained with knowledge distillation was more robust to image corruptions than the teacher; therefore, we aim to evaluate the performance drop of the CNNs trained with the proposed approach when applied on corrupted images.

In particular, we reproduce the experimental framework described in [135] and apply 13 different types of corruptions with 5 levels of severity on the LFW+ dataset. The resulting test set, hereinafter LFW+C, is composed of $1,020,435$ samples. Examples of images extracted from the dataset are depicted in Figure 5.2, while more detailed information about the implementation of the image corruptions and the parameters for each severity value are reported in Section 4.2.1.1. In the following we describe the considered blur, noise and digital corruptions.

Blur Corruptions. Various types of blur can affect the images acquired for real applications, especially in social robotics. *Gaussian blur* may be artificially applied by modern cameras to reduce the negative effect of the acquisition noise. *Defocus blur* can happen when the environment is characterized by a depth

of field larger than the limit of the camera. *Zoom blur* appears whether a person moves towards the camera; this corruption can happen in access control applications. *Motion blur* occurs when a person suddenly changes the pose of the face; this category of blur is very common in digital signage and social robotics applications.

Noise Corruptions. Cameras used for surveillance or on board of a social robot are subjected to overheating, due to 24 hours working or to the external temperature, and may be installed in places characterized by high exposure. These environmental issues cause the presence of random speckles on the acquired images, which can be categorized as two categories of noise. *Gaussian noise* happens when the temperature of the sensor increases over a certain threshold, while *shot noise* occurs in case of high exposure.

Digital Corruptions. This category incorporates all the digital modifications that can appear on the acquired image due to contrast, brightness, occlusions, compression and rescaling. In particular, *contrast increase*, *contrast decrease*, *brightness increase* and *brightness decrease* happen when the modern cameras apply image corrections such as dynamic contrast and automatic gain control to improve the quality of the acquired images. *Spatter* is instead a corruption introduced to reproduce partial occlusions of the face, which can be due to scarves, glasses, sunglasses, masks, parts of the body or other people; this effect is obtained by adding bright random patterns on the image for low corruption severity and dark elements for higher corruption severity. *JPEG compression* is often applied in real applications running server side to reduce the bandwidth consumption; this effect is reproduced by reducing the compression quality with a value inversely proportional to the severity of the corruption. Finally, *pixelation* is the corruption introduced to reproduce the effect of upscaling, which is typically necessary when the input size of the neural network is higher than the size in pixels of the face image. Considering that the input size of the adopted convolutional neural network is 224×224, this corruption can happen very often when the person is not very close to the camera.

Table 5.3: MAE achieved by the considered convolutional neural networks over the test set of VMAGE, IMDB-Wiki and LFW+. The best results for each dataset are highlighted in bold. The methods are sorted in ascending order of the MAE over LFW+, that can be considered an impartial benchmark, since it was not used for training.

Method	Training set	MAE		
		VMAGE	IMDB-Wiki	LFW+
SENet	VMAGE	**1.75**	7.20	**5.58**
VGG	VMAGE	1.82	7.20	**5.58**
MN3-Large	VMAGE	1.84	7.23	5.65
MN3-Small	VMAGE	2.02	7.27	5.69
DenseNet	VMAGE	1.90	7.44	5.89
VGG	IMDB-Wiki	5.56	7.14	6.20
MN3-Small	IMDB-Wiki	4.84	7.17	6.45
DenseNet	IMDB-Wiki	4.82	7.16	6.48
SENet	IMDB-Wiki	5.17	7.23	6.88
MN3-Large	IMDB-Wiki	5.40	**7.11**	7.27

5.4 Experimental results

5.4.1 Results on LFW+

As a first experiment we compare the MAE achieved by each architecture when trained on the distilled VMAGE dataset and on the previously described IMDB-Wiki dataset. In Table 5.3 we present those results sorted in ascending order of the MAE over the LFW+ dataset, that has not been used for training, thus being a fair benchmark. The results show the higher generalization capability obtained by the networks trained with VMAGE; in fact, they achieve a MAE around 1 year lower than the corresponding CNNs trained with IMDB-Wiki.

In the same table we also report the results on the VMAGE test set and on the IMDB-Wiki test set; as expected [31], the trend is that the performance on the test portion of the dataset used for

training is better than the one obtained on an external independent dataset. This is the main reason why most of the competitions allow for fine-tuning on the target dataset. On the VMAGE test set, the models trained on VMAGE achieve significantly lower MAE (up to 3 years) than their IMDB-Wiki-trained rivals, as expected. On the other hand, the advantage of IMDB-Wiki-trained architectures is negligible over the IMDB-Wiki test set (less than 0.1 year in every case). This proves the superior representativeness of the VMAGE dataset with respect to the IMDB-Wiki dataset: the networks trained with the former are able to provide better performance on all the datasets, while the networks trained on the latter are comparable only when tested on the IMDB-Wiki but do not generalize as well.

This comparison confirms the effectiveness of the proposed knowledge distillation technique over the naive approach of training with the standard procedure over the IMDB-Wiki dataset.

Among the different architectures, SENet and the VGG are the CNNs achieving the best performance over LFW+ (5.58), even if the former obtains a slightly smaller MAE on VMAGE (1.75 vs 1.82). The two versions of MN3, Large and Small, achieve a similar MAE (5.65 and 5.69), while DenseNet is at the 5th place (5.89). We also notice that the ranking of the CNNs trained with the proposed technique is the same on IMDB-Wiki and LFW+, while the trend of the others is more random over the different test sets.

5.4.2 Results on LAP 2016

The results achieved in terms of ϵ-error over the LAP 2016 dataset are reported in Table 5.4. The student model based on SENet obtains a notable 0.3033, which is the best performance on this dataset except for the one obtained by the teacher network during the competition [169]. This result is even more relevant if we consider that this CNN overcomes the performance achieved by complex and bulky CNNs or ensembles of them, such as the ones described in [177], [178], [179] and [180]. Examples of face images analyzed by this model are reported in Fig. 5.3.

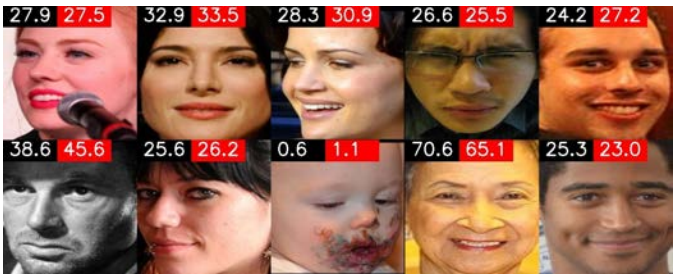

Figure 5.3: Examples of LAP 2016 images analyzed by the proposed student model based on SENet. The apparent age in groundtruth is reported in the black box, while the age estimated by the CNN is annotated in the red box.

In general, all the CNNs trained with the proposed knowledge distillation technique achieve result very close to the performance obtained by substantially more computationally expensive deep neural networks. VGG and MN3-Large (0.3362 and 0.3404) achieve a performance higher than DenseNet and MN3-Small (0.3589 and 0.3601), but the gap with respect to SENet is significant.

The corresponding CNNs trained with IMDB-Wiki achieve a performance substantially lower. The highest gap can be noted over SENet (0.3033 vs 0.4351) and VGG (0.3362 vs 0.4543), but also on MN3-Large (0.3404 vs 0.3944), DenseNet (0.3589 vs 0.4029) and MN3-Small (0.3601 vs 0.4284) and it is substantial. It is interesting to note that all the CNNs trained with this procedure are not able to achieve performance comparable with the ones obtained by the methods who participated in the competition; this experimental evidence demonstrates that state of the art performance are not easily achievable with standard CNNs through the standard pre-training procedure with IMDB-Wiki and further confirms the utility of the proposed technique.

Table 5.4: ϵ-error achieved by the considered convolutional neural networks over LAP 2016. The methods are sorted in descending order of the ϵ-error, so that the best result is at the top.

Method	**Pre-Training**	**ϵ-error**
Antipov et al. [169]	IMDB-Wiki+Private	0.2411
SENet	**VMAGE**	**0.3033**
Tan et al. [177]	Augmented IMDB-Wiki	0.3100
Dehghan et al. [178]	Private	0.3190
Huo et al. [179]	IMDB-Wiki	0.3214
Uricar et al. [180]	IMDB-Wiki	0.3361
VGG	**VMAGE**	**0.3362**
MN3-Large	**VMAGE**	**0.3404**
DenseNet	**VMAGE**	**0.3589**
MN3-Small	**VMAGE**	**0.3601**
Malli et al. [181]	IMDB-Wiki	0.3668
Duan et al. [182]	IMDB-Wiki	0.3679
Gurpinar et al. [183]	VGG-Face	0.3740
MN3-Large	**IMDB-Wiki**	**0.3944**
DenseNet	**IMDB-Wiki**	**0.4029**
MN3-Small	**IMDB-Wiki**	**0.4284**
SENet	**IMDB-Wiki**	**0.4351**
VGG	**IMDB-Wiki**	**0.4543**

5.4.3 Results on Adience

The results analyzed so far demonstrated the capability of the proposed training procedure to produce effective CNNs for real and apparent age estimation. In this experiment, whose results are reported in Table 5.5, we show that the procedure also allows to achieve remarkable performance in age group classification.

In fact, the proposed student model based on SENet holds the 3rd top rank (top-1 65.0%, 1-off 97.1%), followed closely by MN3-Large (top-1 64.1%, 1-off 97.0%), VGG (top-1 64.0%, 1-off 96.9%), DenseNet (top-1 63.5%, 1-off 96.2%) and MN3-Small (top-1 62.5%, 1-off 96.6%). The high accuracy 1-off for all the CNNs pre-trained with VMAGE demonstrates that these models make a negligible mistake, confusing the exact age group with an adjacent one in most cases.

The significant superiority with respect to the corresponding CNNs pre-trained with IMDB-Wiki is a further confirmation of the effectiveness of the proposed technique compared to that typically used in literature. Indeed, it allows to achieve an accuracy higher or very close to the ones obtained by CNNs more complex, as the Residual of Residual network (RoR) with 152 layers adopted by Zhang et al. [184] or the already described ensemble of 20 CNNs used by Rothe et al. [67], or architectures tailored for the purpose, such as the VGG-16 modified by Hou et al. [185] with smoothed adaptive activation functions (SAAF) for reducing the regression bias.

5.4.4 Robustness to image corruptions

In our last experiment we evaluate the robustness of the considered models to generalize to the image corruptions described in Section 5.3.4. This experiment allows to estimate the performance of these models on images acquired in real scenarios and to compare the robustness of the student models with the one of the teacher.

The results summarized in Fig. 5.4 confirm the experimental findings reported in [166]. In fact, we notice that three of the student models, namely SENet, MN3-Small and VGG (MAE of 7.87,

Table 5.5: Accuracy top-1 and 1-off achieved by the considered CNNs over Adience. The methods are sorted in descending order of the accuracy top-1, so that the best result is at the top.

Method	**Pre-Training**	**top-1**	**1-off**
Zhang et al. [184]	IMDB-Wiki	67.3	97.5
Hou et al. [185]	IMDB-Wiki	67.3	97.4
SENet	**VMAGE**	**65.0**	**97.1**
MN3-Large	**VMAGE**	**64.1**	**97.0**
VGG	**VMAGE**	**64.0**	**96.9**
Rothe et al. [67]	IMDB-Wiki	64.0	96.6
DenseNet	**VMAGE**	**63.5**	**96.2**
Lapuschkin et al. [176]	IMDB-Wiki	62.8	95.8
MN3-Small	**VMAGE**	**62.5**	**96.6**
DenseNet	**IMDB-Wiki**	**61.1**	**95.5**
Hou et al. [186]	ImageNet	61.1	94.0
VGG	**IMDB-Wiki**	**60.7**	**94.5**
MN3-Large	**IMDB-Wiki**	**60.6**	**94.3**
Liu et al. [187]	ImageNet	60.2	93.7
SENet	**IMDB-Wiki**	**59.9**	**94.4**
Qawaqneh et al. [188]	VGG-Face	59.9	90.6
MN3-Small	**IMDB-Wiki**	**57.6**	**92.8**
Chen et al. [189]	Mixed	52.9	88.4
Levi et al. [190]	No	50.7	84.7
Eidinger et al. [148]	No	45.1	80.7

Table 5.6: MAE achieved by the considered CNNs on the corruption categories in LFW+C. The columns are divided in three blocks, one for each corruption category (blur, noise, digital). The methods are sorted in ascending order of the MAE over LFW+C, so that the best result is at the top, while the best MAE for each corruption category is highlighted in bold.

Method	**LFW+C**	**Blur**				**Noise**		**Digital**						
		Gaussian	Defocus	Zoom	Motion	Gaussian	Shot	Cont.Inc.	Cont.Dec.	Brig.Inc.	Brig.Dec.	Spatter	JPEG Comp.	Pixelation
SENet	7.87	7.05	16.63	**6.42**	11.29	6.83	7.08	**6.20**	**5.91**	**6.23**	**6.21**	10.83	5.76	5.82
MN3-Small	7.96	7.23	16.54	6.44	11.42	7.48	7.69	6.55	6.36	6.67	6.41	**8.91**	5.89	5.92
VGG	8.03	7.04	16.58	**6.42**	**11.27**	7.05	7.45	6.23	5.97	6.38	6.22	11.93	5.91	5.95
Antipov et al. [169]	8.07	**7.01**	18.53	6.47	11.41	**6.40**	**6.49**	6.97	6.31	6.66	6.31	11.28	**5.50**	**5.59**
DenseNet	8.19	7.66	**16.19**	7.02	12.46	7.11	7.54	6.58	6.76	6.61	6.71	9.70	6.00	6.15
MN3-Large	8.97	7.69	19.24	6.72	11.92	7.62	8.27	6.42	6.51	6.65	6.30	10.29	5.80	5.81

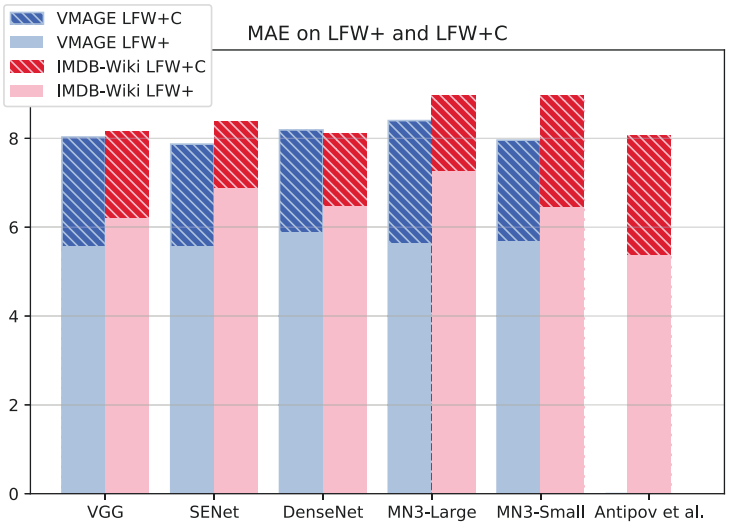

Figure 5.4: MAE achieved by the considered CNNs on the LFW+ dataset (light bar) and its corrupted version LFW+C (dark bar). We compare the results achieved when the networks are pre-trained using IMDB-Wiki (orange) and VMAGE (blue).

7.96 and 8.03) are more robust to corruptions than the teacher (MAE=8.07). In particular, SENet achieves a MAE 0.2 years lower than the teacher, which in turn obtained a MAE 0.2 years lower on the original LFW+ dataset; this result demonstrates that the proposed distillation technique allows to provide some of the student models with a higher generalization capability than the teacher over corrupted images. In all the cases, the CNNs trained with VMAGE achieve lower MAE (around 0.5 years for SENet and MN3-Large, 1 year for MN3-Small and 0.1 years for VGG) over LFW+C than the corresponding ones adopting IMDB-Wiki, except for DenseNet (8.19 vs 8.12). A noteworthy result is that obtained by MN3, whose Small version achieves a better performance than the Large one; this result can be explained by the fact that a smaller CNN may generalize better over images very different from the ones used for training.

The analysis can be further extended by evaluating the MAE achieved by the considered CNNs over images perturbed with specific corruption categories; the detailed results of this experiment are reported in Table 5.6. We can note that the teacher is more robust to gaussian blur, gaussian and shot noise, JPEG compression and pixelation, while it suffers in case of brightness and contrast variations, spatter (11.28) and, significantly, in case of defocus blur (18.53, almost 2 years more than the average of the student models). SENet achieves very balanced performance over the different categories and the best MAE when dealing with zoom blur (6.42, as well as VGG), contrast increase (6.20) and decrease (5.91), brightness increase (6.23) and decrease (6.21). MN3-Small is substantially more resilient than other CNNs to spatter (8.91), while VGG obtains the best MAE over motion blur (11.27) and DenseNet over defocus blur (16.19).

In general, we can note that spatter, motion blur and defocus blur are the corruptions causing more problems to the considered CNNs. This evidence can be explained by the fact that these corruptions strongly reduce the facial details, substantially more than the other categories.

Chapter 6

Conclusions

The foundation of this thesis is the observation of the usefulness of contextual clues in the context of social robotics: studies prove that human-like behaviour is key to generate in the interlocutor the feeling of empathy that allow him to subconsciously perceive the robot as his peer. From the analysis of the faces of the people around, the robot can gather information that allows to personalize the interaction and enhance the feeling of empathy given by the robot. Such information, including age, gender, ethnicity, emotion and more is called "soft biometrics" because it does not allow unique, perfect, identification of a person, but it is nevertheless used by humans to distinguish their peers; soft biometrics is not the only tool a social robot will need to function, but it is arguably one of the main ones.

We observe that tasks in the domain of facial soft biometrics are extensively studied in literature but the application to realistic environments introduces some constraints that require specific attention, namely resource constraints and robustness constraints. Resource constraints are limitations due to the actual hardware that runs the prediction systems; such constraints for example require the memory footprint to be confined to what the hardware can handle and require the inference time to be limited as well, in order for the information to be available in good time to be used in a naturally paced iteration. Robustness concerns the ability of the system to produce correct predictions based on input images that are affected by all kinds of corruptions and perturbation that are present on images acquired in unconstrained conditions using typical hardware from the considered application; for instance embedded cameras produce noisy images with limited resolution and dynamic range.

In the thesis we tackle all those themes in the context of Deep Learning. We design and evaluate efficient and effective CNN-based methods for the tasks of gender recognition, ethnicity recognition, age estimation and emotion classification.

As a first contribution, we design an efficient CNN architecture for the task of gender recognition. We recognize that all the CNN architectures that are typically applied to gender recognition

are designed for the task of recognizing generic objects; we argue that gender recognition networks need a smaller input size and a more shallow hierarchy of layers than what is needed for object recognition. Following the design of the MobileNetV2 architecture, propose an architecture with 3 hyperparameters, namely the number of feature maps (width), the number of residual blocks (depth) and the input size. We study the inference time and the accuracy of the proposed architecture with different values of the hyperparameters and we find that our proposed architecture is able to recognize gender with an accuracy of 98.1% in just 56ms on an embedded device without any neural network acceleration. We compare our results with the ones publicly available in the state of the art and we find that our proposed method is up to 1% more accurate than existing efficient architectures with comparable inference time, proving the effectiveness of our method.

As a second contribution, we observe that the task of Ethnicity recognition is held back by the absence of a large dataset. We effectively design a dataset by having people of different ethnicities annotate the images of 9000 famous people with the ethnicity they recognize. The resulting dataset, that we call VMER, is unbiased according to the other race effect thanks to this annotation procedure; it uses our annotated information along with 3.3 million images of the annotated 9000 people taken from the large scale VGG dataset. We train multiple commonly used neural network architectures and evaluate them on public independent benchmarks. The accuracy achieved by the architectures trained on the VMER dataset is higher than the one achieved by training the same architectures on different datasets from the literature: this proves that our proposed dataset is more representative than others in existing literature. We make our labels publicly available and we believe that such labels will allow for futher advancement of the state of the art in automatic ethnicity recognition from face images.

As a third contribution, we evaluate the robustness of CNNs to corruptions of the input face images, and the stability of the predicitons when the input is subject to perturbations. We eval-

uate 4 CNN architectures (VGG, SENet, DenseNet and Xception) and we evaluate the effect of Autoaugment and antialiased downsampling on those architectures, the first being a technique for effectively augmenting the training data and the second being an architectural modification that adds low pass filters inside the networks wherever a downsampling happen (e.g. max-pooling or strided convolutions). For our evaluation, we construct a benchmark data set on top of the RAF-DB test set that includes images with corruptions that typically occur when the recognition systems are deployed in real scenarios. Corruptions include different kinds of blur (motion blur, lens blur, zoom blur, gaussian blur), of noise (gaussian noise, shot noise), pixelation, jpeg compression, changes in brightness and contrast and combination of those. For evaluating the stability of the predictions, we generate the RAF-DB-P dataset, that includes versions of the testing images where we perturb the brightness, the position, the scale, the rotation, the quantity of blur and the pattern of noise. We find that the combined use of antialiasing and Autoaugment substantially contributes to the improvement of the robustness to corruptions, especially to those of the noise and digital type, of SENet and DenseNet. The VGG architecture instead showed the highest classification stability with respect to perturbations that affect subsequent frames of a sequence, especially when combined with the use anti-aliasing filters. The Xception methods are not suitable for facial emotion analysis in the wild since they are expecially affected by corruptions and perturbations. In conclusion our experiments demonstrated that the common corruptions and perturbations are important aspects to take into account when evaluating methods to be deployed in real scenarios. However, none of the existing methods, which we modified with anti-aliasing filters and trained with extensive data augmentation, showed robustness to all the considered corruptions and perturbations, thus this aspect requires future investigation.

As a fourth contribution, we propose and experiment a simple procedure to train CNNs for an age estimation method that is both efficient and accurate. We observe that the training of such

a method has been hindered by the absence of a large-scale reliably annotated dataset. The commonly used IMDB-Wiki dataset in fact is automatically annotated and its annotation is extremely noisy, while other datasets may be more accurately annotated but their size is insufficient for the purposes of Deep Learning. Previous work was able to overcome those limitations by cleaning the dataset with fairly long and expensive manual procedures (the results of which are not public) and by implementing large and slow ensembles of neural networks. We propose to use Knowledge Distillation to transfer the knowledge from such a large and slow method (the teacher) to lightweight CNNs (students): we annotate a large scale dataset of face images using the large teacher method and then we use that dataset to train some commonly used architectures. We experiment our method on various public benchmarks, where we find that our student CNNs surpass the accuracy of most pre-existing methods, even ones that are much more complex. We prove the effectiveness of our procedure by training the same architectures on our distilled VMAGE dataset and on a cleaned version of the IMDB-Wiki dataset from the literature, proving that using VMAGE is consistently and significantly beneficial. We make VMAGE publicly available for other researchers to use it in their work. The student architectures are all able to be executed in under 100 milliseconds on embedded hardware (Nvidia Jetson TX1) while the execution of the teacher method on a single image takes more than 6 seconds. We finally find that the student models show comparable or better performance with respect to the teacher methods when the input images subjected to the corruptions described before. This robustness advantage is an additional proof of the effectiveness of the training approach proposed.

Overall we were able, for each of the four tasks, gender, age, ethnicity, emotion, to design a CNN-based system able to achieve state of art performance while being able to perform in the target social robotic environment, with limited inference time and memory requirements and able to work in reasonably "wild", uncontrolled settings.

6.1 Outlook

In this section we aim to give an outlook on what is missing in this thesis, thus what can be done in future work to further advance the state of the art concerning facial soft biometrics for social robotic applications.

We find though that robustness of CNNs to strong corruptions still needs to be improved, expecially with respect to random noise; this aspect is relevant because such noise is caused by thermal effects and it appears strong with small cameras in not well lit environment. New architectural components or training strategies that take into account the occurrence of perturbations between subsequent frames may be designed to reduce the performance drop of the existing methods when dealing with corruptions and perturbations.

We found that more mature tasks, such as gender recognition and age estimation, struggle with children and elders as well as asians. We believe that such a situation is due to the underrepresentation of such categories in the widespread public datasets. Future work should definitely address the problem from a dataset design perspective; however, the issue can be mitigated by appropriately designing a training procedure to counteract the imbalance, for example via a weighted loss function or a custom sampling of the images that compose each minibatch, making sure that enough variability is considered into each training iteration.

We found that the data imbalance has non negligible effect on the behaviour of the trained network. Most literature disregards this aspect, but we believe that this effect should be better studied in future work, integrating those considerations into the standard benchmarks. For example the LFW+ benchmark is imbalanced with respect to gender, with a prominence of males; this benchmark thus provides a skewed representation of neural network performance because a network that is trained to reflect the a-priori distribution of the benchmark itself will be unfairly judged to be more accurate while an on-the-field test would show that they more often mistake females for males than vice-versa. The issue

is even more strongly felt with emotion recognition, since some classes are more easily recognizable, thus better represented in the datasets (namely happy, angry and the catch-all class neutral) while others are more subtle (for instance sad or disgust).

Concerning the efficiency of network architecture, in this work we focused on the architectural features, but we are well aware that the proposed architectures could be further improved by using techniques such as quantization of weights and activations. Since different architectures respond to quantization in different ways, the architectures should be evaluated with respect to this observation, and design principles should be identified in a way that minimizes both the performance drop and the time and memory required at inference time.

A further way to improve effciency is to look at the overall system and combine multiple predictors into a multitask system: a multitask neural network is composed by a shared stack of layers that extract common low-level features, and distinct classification branches, where specific features are derived from the low-level ones and the final predictions are performed. Such networks are efficient in that they perform multiple tasks at once with a computational burden that is roughly the same of a single task network. If they are trained properly, their accuracy is comparable or even higher than single task equivalents, due to the fact that the neural network is able to extract more generic features and even to learn inter-dependencies between Combining such multitask techniques with the techniques developed in this work would be useful to the development of the soft biometric subsystem of a social robot, allowing for better efficiency, thus leaving more space for other subsystems.

Finally, for emotion recognition in particular, future development of this work will definitely include the inclusion of temporal analysis: facial expressions in fact, happen through time, with a sequence of phases that represent the onset, peak and offset of the emotion. Recurrent neural CNNs may be a natural extension of the forward CNN approach shown in this work, and literature on their use for the problem at hand has existed for years now. Their

robustness though is yet to be measured, and different challenges may arise in their design process.

Bibliography

[1] D. Fischinger, P. Einramhof, K. Papoutsakis, W. Wohlkinger, P. Mayer, P. Panek, S. Hofmann, T. Koertner, A. Weiss, A. Argyros *et al.*, "Hobbit, a care robot supporting independent living at home: First prototype and lessons learned," *Robotics and Autonomous Systems*, vol. 75, pp. 60–78, 2016.

[2] E. Mordoch, A. Osterreicher, L. Guse, K. Roger, and G. Thompson, "Use of social commitment robots in the care of elderly people with dementia: A literature review," *Maturitas*, vol. 74, no. 1, pp. 14–20, 2013.

[3] B. Scassellati, H. Admoni, and M. Matarić, "Robots for use in autism research," *Annual review of biomedical engineering*, vol. 14, 2012.

[4] R. Triebel, K. Arras, R. Alami, L. Beyer, S. Breuers, R. Chatila, M. Chetouani, D. Cremers, V. Evers, M. Fiore *et al.*, "Spencer: A socially aware service robot for passenger guidance and help in busy airports," in *Field and service robotics*. Springer, 2016, pp. 607–622.

[5] Z.-H. Tan, N. B. Thomsen, X. Duan, E. Vlachos, S. E. Shepstone, M. H. Rasmussen, and J. L. Højvang, "isociobot: A multimodal interactive social robot," *International Journal of Social Robotics*, vol. 10, no. 1, pp. 5–19, 2018.

[6] A. Saggese, M. Vento, and V. Vigilante, "Miviabot: A cognitive robot for smart museum," in *International Conference on Computer Analysis of Images and Patterns*, M. Vento and G. Percannella, Eds., Springer. Cham: Springer International Publishing, 2019, pp. 15–25.

[7] C. Schroeter, S. Mueller, M. Volkhardt, E. Einhorn, C. Huijnen, H. van den Heuvel, A. van Berlo, A. Bley, and H.-M. Gross, "Realization and user evaluation of a companion robot for people with mild cognitive impairments," in *2013 IEEE International Conference on robotics and automation*. IEEE, 2013, pp. 1153–1159.

[8] B. Graf, M. Hans, and R. D. Schraft, "Care-o-bot ii - development of a next generation robotic home assistant," *Autonomous robots*, vol. 16, no. 2, pp. 193–205, 2004.

[9] R. Kittmann, T. Fröhlich, J. Schäfer, U. Reiser, F. Weißhardt, and A. Haug, "Let me introduce myself: I am care-o-bot 4, a gentleman robot," *Mensch und computer 2015–proceedings*, 2015.

[10] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "Experiences with an interactive museum tour-guide robot," *Artificial intelligence*, vol. 114, no. 1-2, pp. 3–55, 1999.

[11] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, "Speech recognition using deep neural networks: A systematic review," *IEEE Access*, vol. 7, pp. 19 143–19 165, 2019.

[12] E. Battenberg, J. Chen, R. Child, A. Coates, Y. G. Y. Li, H. Liu, S. Satheesh, A. Sriram, and Z. Zhu, "Exploring neural transducers for end-to-end speech recognition," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 206–213.

[13] Á. Miklósi, P. Korondi, V. Matellán, and M. Gácsi, "Ethorobotics: A new approach to human-robot relationship," *Frontiers in psychology*, vol. 8, p. 958, 2017.

[14] C. Breazeal, K. Dautenhahn, and T. Kanda, "Social robotics," in *Springer handbook of robotics*. Springer, 2016, pp. 1935–1972.

[15] M. Mori, K. F. MacDorman, and N. Kageki, "The uncanny valley [from the field]," *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 98–100, 2012.

[16] S. Meyer, *My robot friend: service robotics for elders, a response to the demographic change? (Mein Freund der Roboter: Servicerobotik für ältere Menschen - eine Antwort auf den demographischen Wandel?) ; Studie im Auftrag von VDE - Verband der Elektrotechnik, Elektronik, Informationstechnik, VDI - Verein Deutscher Ingenieure e.V., BMBF/VDE Innovationspartnerschaft AAL, DKE - Deutsche Kommission Elektrotechnik, Elektronik, Informationstechnik im DIN und VDE.* VDE-Verl., 2011, document in German.

[17] E. Wiese, G. Metta, and A. Wykowska, "Robots as intentional agents: using neuroscientific methods to make robots appear more social," *Frontiers in psychology*, vol. 8, p. 1663, 2017.

[18] B. R. Duffy, C. Rooney, G. M. O'Hare, and R. O'Donoghue, "What is a social robot?" in *10th Irish Conference on Artificial Intelligence & Cognitive Science, University College Cork, Ireland, 1-3 September, 1999*, 1999.

[19] Z. Cai, B. Goertzel, and N. Geisweiller, "Openpsi: Realizing dorner's "psi" cognitive model in the opencog integrative agi architecture," in *International Conference on Artificial General Intelligence*. Springer, 2011, pp. 212–221.

[20] D. Hart and B. Goertzel, "Opencog: A software framework for integrative artificial general intelligence," in *AGI*, 2008, pp. 468–472.

[21] B. Goertzel, H. De Garis, C. Pennachin, N. Geisweiller, S. Araujo, J. Pitt, S. Chen, R. Lian, M. Jiang, Y. Yang *et al.*, "Opencogbot: achieving generally intelligent virtual agent control and humanoid robotics via cognitive synergy," in *Proceedings of ICAI*, vol. 10. Citeseer, 2010, pp. 1–12.

[22] K. Inoue, D. Lala, K. Yamamoto, S. Nakamura, K. Takanashi, and T. Kawahara, "An attentive listening system with android erica: Comparison of autonomous and woz interactions," in *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2020, pp. 118–127.

[23] S. A. Abdul-Kader and J. Woods, "Survey on chatbot design techniques in speech conversation systems," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 7, 2015.

[24] K. Jokinen, "Dialogue models for socially intelligent robots," in *International Conference on Social Robotics*. Springer, 2018, pp. 127–138.

[25] A. Dantcheva, C. Velardo, A. D'angelo, and J.-L. Dugelay, "Bag of soft biometrics for person identification," *Multimedia Tools and Applications*, vol. 51, no. 2, pp. 739–777, 2011.

[26] A. K. Jain, S. C. Dass, and K. Nandakumar, "Soft biometric traits for personal recognition systems," in *International conference on biometric authentication*. Springer, 2004, pp. 731–738.

[27] K. Niinuma, U. Park, and A. K. Jain, "Soft biometric traits for continuous user authentication," *IEEE Transactions on information forensics and security*, vol. 5, no. 4, pp. 771–780, 2010.

[28] E. Gonzalez-Sosa, J. Fierrez, R. Vera-Rodriguez, and F. Alonso-Fernandez, "Facial soft biometrics for recognition in the wild: Recent works, annotation, and cots evaluation," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 2001–2014, 2018.

[29] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition*. IEEE Computer Society, 1991, pp. 586–587.

[30] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[31] V. Carletti, A. Greco, G. Percannella, and M. Vento, "Age from faces in the deep learning revolution," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 9, pp. 2113–2132, 2019.

[32] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 2018, pp. 67–74.

[33] J. Holler, M. Casillas, K. H Kendrick, and S. C Levinson, *Turn-taking in human communicative interaction*. Frontiers Media SA, 2016.

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[35] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Conf. on CVPR*, vol. 1, 2001, pp. I–I.

[36] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool, "Face detection without bells and whistles," in *IEEE Conf. ECCV*. Springer, 2014, pp. 720–735.

[37] S. Liao, A. K. Jain, and S. Z. Li, "A fast and accurate unconstrained face detector," *IEEE Trans. on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 211–223, 2016.

[38] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[39] M. Najibi, P. Samangouei, R. Chellappa, and L. S. Davis, "Ssh: Single stage headless face detector," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4875–4884.

[40] T. Hassner, S. Harel, E. Paz, and R. Enbar, "Effective face frontalization in unconstrained images," in *IEEE Conf. on CVPR*, 2015, pp. 4295–4304.

[41] P.-H. Lee, S.-W. Wu, and Y.-P. Hung, "Illumination compensation using oriented local histogram equalization and its application to face recognition," *IEEE Transactions on Image processing*, vol. 21, no. 9, pp. 4280–4289, 2012.

[42] A. Mustapha, A. Oulefki, M. Bengherabi, E. Boutellaa, and M. A. Algaet, "Towards nonuniform illumination face enhancement via adaptive contrast stretching," *Multimedia Tools and Applications*, vol. 76, no. 21, pp. 21 961–21 999, 2017.

[43] C. Wang, Y. Li, and C. Wang, "An efficient illumination compensation based on plane-fit for face recognition," in *2008 10th International Conference on Control, Automation, Robotics and Vision*. IEEE, 2008, pp. 939–943.

[44] G. Azzopardi, A. Greco, A. Saggese, and M. Vento, "Fast gender recognition in videos using a novel descriptor based on the gradient magnitudes of facial landmarks," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2017, pp. 1–6.

[45] B. Moghaddam and M.-H. Yang, "Learning gender with support faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 707–711, 2002.

[46] H.-C. Lian and B.-L. Lu, "Multi-view gender classification using local binary patterns and support vector machines," in *International Symposium on Neural Networks*. Springer, 2006, pp. 202–209.

[47] G. Azzopardi, A. Greco, and M. Vento, "Gender recognition from face images using a fusion of svm classifiers," in *International Conference on Image Analysis and Recognition*. Springer, 2016, pp. 533–538.

[48] G. Rhodes, C. Hickford, and L. Jeffery, "Sex-typicality and attractiveness: Are supermale and superfemale faces superattractive?" *British journal of psychology*, vol. 91, no. 1, pp. 125–140, 2000.

[49] G. Guo, C. R. Dyer, Y. Fu, and T. S. Huang, "Is gender recognition affected by age?" in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*. IEEE, 2009, pp. 2032–2039.

[50] J. E. Tapia and C. A. Perez, "Gender classification based on fusion of different spatial scale features selected by mutual information from histogram of lbp, intensity, and shape," *IEEE transactions on information forensics and security*, vol. 8, no. 3, pp. 488–499, 2013.

[51] S. Jia and N. Cristianini, "Learning to classify gender from four million images," *Pattern recognition letters*, vol. 58, pp. 35–41, 2015.

[52] A. Halevy, P. Norvig, and F. Pereira, "The unreasonable effectiveness of data," *IEEE Intelligent Systems*, vol. 24, no. 2, pp. 8–12, 2009.

[53] G. Azzopardi, A. Greco, and M. Vento, "Gender recognition from face images with trainable cosfire filters," in *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2016, pp. 235–241.

[54] 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*, 2015, pp. 34–42.

[55] J. van de Wolfshaar, M. F. Karaaba, and M. A. Wiering, "Deep convolutional neural networks and support vector machines for gender recognition," in *2015 IEEE Symposium Series on Computational Intelligence*, 2015, pp. 188–195.

[56] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[57] G. Antipov, M. Baccouche, S.-A. Berrani, and J.-L. Dugelay, "Effective training of convolutional neural networks for face-based

gender and age prediction," *Pattern Recognition*, vol. 72, pp. 15–26, 2017.

[58] G. Antipov, S.-A. Berrani, and J.-L. Dugelay, "Minimalistic cnn-based ensemble model for gender prediction from face images," *Pattern recognition letters*, vol. 70, pp. 59–65, 2016.

[59] P. Foggia, A. Greco, G. Percannella, M. Vento, and V. Vigilante, "A system for gender recognition on mobile robots," in *Proceedings of the 2nd International Conference on Applications of Intelligent Systems*. ACM, 2019, p. 9.

[60] S. M. J. Jalali, S. Ahmadian, P. M. Kebria, A. Khosravi, C. P. Lim, and S. Nahavandi, "Evolving artificial neural networks using butterfly optimization algorithm for data classification," in *International Conference on Neural Information Processing*. Springer, 2019, pp. 596–607.

[61] P. M. Kebria, A. Khosravi, S. M. Salaken, I. Hossain, H. D. Kabir, A. Koohestani, R. Alizadehsani, and S. Nahavandi, "Deep imitation learning: The impact of depth on policy performance," in *International Conference on Neural Information Processing*. Springer, 2018, pp. 172–181.

[62] P. M. Kebria, A. Khosravi, S. M. Salaken, and S. Nahavandi, "Deep imitation learning for autonomous vehicles based on convolutional neural networks," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 82–95, 2019.

[63] S. Escalera, M. Torres Torres, B. Martinez, X. Baró, H. Jair Escalante, I. Guyon, G. Tzimiropoulos, C. Corneou, M. Oliu, M. Ali Bagheri *et al.*, "Chalearn looking at people and faces of the world: Face analysis workshop and challenge 2016," in *Proc. of IEEE Conf. on CVPR Workshops*, 2016, pp. 1–8.

[64] Y. Fu, G. Guo, and T. S. Huang, "Age synthesis and estimation via faces: A survey," *IEEE Trans. on PAMI*, pp. 1955–1976, 2010.

[65] A. Othmani, A. R. Taleb, H. Abdelkawy, and A. Hadid, "Age estimation from faces using deep learning: A comparative analysis," *Computer Vision and Image Understanding*, p. 102961, 2020.

[66] P. Punyani, R. Gupta, and A. Kumar, "Neural networks for facial age estimation: a survey on recent advances," *Artificial Intelligence Review*, vol. 53, no. 5, pp. 3299–3347, 2020.

[67] R. Rothe, R. Timofte, and L. Van Gool, "Deep expectation of real and apparent age from a single image without facial landmarks," *Int. Journal of Computer Vision*, Aug. 2016.

[68] B.-C. Chen, C.-S. Chen, and W. H. Hsu, "Cross-age reference coding for age-invariant face recognition and retrieval," in *Proc. of Springer ECCV*, 2014.

[69] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[70] I. Anwar and N. U. Islam, "Learned features are better for ethnicity classification," *Cybernetics and Information Technologies*, vol. 17, no. 3, pp. 152–164, 2017.

[71] H. Han, A. K. Jain, F. Wang, S. Shan, and X. Chen, "Heterogeneous face attribute estimation: A deep multi-task learning approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 11, pp. 2597–2609, 2017.

[72] Z. Zhang, Y. Song, and H. Qi, "Age progression/regression by conditional adversarial autoencoder," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5810–5818.

[73] K. Kärkkäinen and J. Joo, "Fairface: Face attribute dataset for balanced race, gender, and age," *arXiv preprint arXiv:1908.04913*, 2019.

[74] S. Fu, H. He, and Z.-G. Hou, "Learning race from face: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2483–2509, 2014.

[75] L. Seidenari, A. Rozza, and A. Del Bimbo, "Real-time demographic profiling from face imagery with fisher vectors," *Machine Vision and Applications*, vol. 30, no. 2, pp. 359–374, 2019.

[76] J. K. Wagner, J.-H. Yu, J. O. Ifekwunigwe, T. M. Harrell, M. J. Bamshad, and C. D. Royal, "Anthropologists' views on race, ancestry, and genetics," *American Journal of Physical Anthropology*, vol. 162, no. 2, pp. 318–327, 2017.

[77] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The feret database and evaluation procedure for face-recognition algorithms," *Image and Vision Computing*, vol. 16, no. 5, pp. 295–306, 1998.

[78] M. J. Lyons, J. Budynek, and S. Akamatsu, "Automatic classification of single facial images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 12, pp. 1357–1362, 1999.

[79] A. Bastanfard, M. A. Nik, and M. M. Dehshibi, "Iranian face database with age, pose and expression," *Machine Vision*, pp. 50–55, 2007.

[80] W. Gao, B. Cao, S. Shan, X. Chen, D. Zhou, X. Zhang, and D. Zhao, "The cas-peal large-scale chinese face database and baseline evaluations," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 38, no. 1, pp. 149–161, 2007.

[81] K. Ricanek and T. Tesafaye, "Morph: A longitudinal image database of normal adult age-progression," in *Proc. of IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 2006, pp. 341–345.

[82] C. E. Thomaz and G. A. Giraldi, "A new ranking method for principal components analysis and its application to face image analysis," *Image and Vision Computing*, vol. 28, no. 6, pp. 902–913, 2010.

[83] N. Kumar, A. Berg, P. N. Belhumeur, and S. Nayar, "Describable visual attributes for face verification and image search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 1962–1977, 2011.

[84] S.-Y. Fu, G.-S. Yang, and Z.-G. Hou, "Spiking neural networks based cortex like mechanism: A case study for facial expression recognition," in *Int. Conf. on Neural Networks*. IEEE, 2011, pp. 1637–1642.

[85] H. Zawbaa and S. A. Aly, "Hajj and umrah event recognition datasets," *arXiv preprint arXiv:1205.2345*, 2012.

[86] D. Riccio, G. Tortora, M. De Marsico, and H. Wechsler, "Ega-ethnicity, gender and age, a pre-annotated face database," in *IEEE Workshop on BIOMS*. IEEE, 2012, pp. 1–8.

[87] V. LoBue and C. Thrasher, "The child affective facial expression (cafe) set: validity and reliability from untrained adults," *Frontiers in Psychology*, vol. 5, p. 1532, 2015.

[88] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. of IEEE ICCV*, 2015, pp. 3730–3738.

[89] M. Demirkus, K. Garg, and S. Guler, "Automated person categorization for video surveillance using soft biometrics," in *Biometric Technology for Human Identification VII*, vol. 7667. International Society for Optics and Photonics, 2010, p. 76670P.

[90] Y. Xie, K. Luu, and M. Savvides, "A robust approach to facial ethnicity classification on large scale face databases," in *Int. Conf. on Biometrics: Theory, Applications and Systems*. IEEE, 2012, pp. 143–149.

[91] S. M. M. Roomi, S. Virasundarii, S. Selvamegala, S. Jeevanandham, and D. Hariharasudhan, "Race classification based on facial features," in *Conf. on computer vision, pattern recognition, image processing and graphics*. IEEE, 2011, pp. 54–57.

[92] B. Wu, H. Ai, and C. Huang, "Facial image retrieval based on demographic classification," in *Int. Conf. on Pattern Recognition*, vol. 3. IEEE, 2004, pp. 914–917.

[93] S. Hosoi, E. Takikawa, and M. Kawade, "Ethnicity estimation with facial images," in *IEEE Int. Conf. on Automatic Face and Gesture Recognition*. IEEE, 2004, pp. 195–200.

[94] H. Lin, H. Lu, and L. Zhang, "A new automatic recognition system of gender, age and ethnicity," in *Congress on Intelligent Control and Automation*, vol. 2. IEEE, 2006, pp. 9988–9991.

[95] S. H. Salah, H. Du, and N. Al-Jawad, "Fusing local binary patterns with wavelet features for ethnicity identification," in *World Academy of Science, Engineering and Technology*, no. 79. World Academy of Science, Engineering and Technology (WASET), 2013, p. 471.

[96] G. Muhammad, M. Hussain, F. Alenezy, G. Bebis, A. M. Mirza, and H. Aboalsamh, "Race classification from face images using local descriptors," *Int. J. on Artificial Intelligence Tools*, vol. 21, no. 05, p. 1250019, 2012.

[97] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikainen, X. Chen, and W. Gao, "Wld: A robust local image descriptor," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1705–1720, 2010.

[98] A. Ahmed, K. Yu, W. Xu, Y. Gong, and E. Xing, "Training hierarchical feed-forward visual recognition models using transfer learning from pseudo-tasks," in *European Conf. on Computer Vision*. Springer, 2008, pp. 69–82.

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

[100] G. Guo and G. Mu, "A framework for joint estimation of age, gender and ethnicity on a large database," *Image and Vision Computing*, vol. 32, no. 10, pp. 761–770, 2014.

[101] C. A. Corneanu, M. O. Simón, J. F. Cohn, and S. E. Guerrero, "Survey on rgb, 3d, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications," *IEEE Trans Pattern Anal Mach Intell*, vol. 38, no. 8, pp. 1548–1568, 2016.

[102] H. Gunes and H. Hung, "Is automatic facial expression recognition of emotions coming to a dead end? the rise of the new kids

on the block." *Image and Vision Computing*, vol. 55, pp. 6–8, 2016.

[103] A. Greco, A. Roberto, A. Saggese, M. Vento, and V. Vigilante, "Emotion analysis from faces for social robotics," in *IEEE SMC*, 2019, pp. 358–364.

[104] D. Kollias, P. Tzirakis, M. A. Nicolaou, A. Papaioannou, G. Zhao, B. Schuller, I. Kotsia, and S. Zafeiriou, "Deep affect prediction in-the-wild: Aff-wild database and challenge, deep architectures, and beyond," *International Journal of Computer Vision*, vol. 127, no. 6-7, pp. 907–929, 2019.

[105] D. Kollias, S. Cheng, E. Ververas, I. Kotsia, and S. Zafeiriou, "Deep neural network augmentation: Generating faces for affect analysis," *International Journal of Computer Vision*, pp. 1–30, 2020.

[106] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee *et al.*, "Challenges in representation learning: A report on three machine learning contests," in *NeurIPS*, 2013, pp. 117–124.

[107] A. Mollahosseini, B. Hasani, and M. H. Mahoor, "Affectnet: A database for facial expression, valence, and arousal computing in the wild," *IEEE Transactions on Affective Computing*, vol. 10, no. 1, pp. 18–31, 2017.

[108] E. Barsoum, C. Zhang, C. C. Ferrer, and Z. Zhang, "Training deep networks for facial expression recognition with crowdsourced label distribution," in *ACM ICMI*, 2016, pp. 279–283.

[109] S. Li, W. Deng, and J. Du, "Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild," in *IEEE CVPR*, 2017, pp. 2584–2593.

[110] Y. Kim, B. Yoo, Y. Kwak, C. Choi, and J. Kim, "Deep generative-contrastive networks for facial expression recognition," *arXiv:1703.07140*, 2017.

[111] Y. Fan, J. C. Lam, and V. O. Li, "Multi-region ensemble convolutional neural network for facial expression recognition," in *ICANN*, 2018, pp. 84–94.

[112] Y. Li, J. Zeng, S. Shan, and X. Chen, "Occlusion aware facial expression recognition using cnn with attention mechanism," *IEEE Trans Image Process*, vol. 28, no. 5, pp. 2439–2450, 2018.

[113] D. Acharya, Z. Huang, D. Pani Paudel, and L. Van Gool, "Covariance pooling for facial expression recognition," in *IEEE CVPR Workshops*, 2018, pp. 367–374.

[114] T. S. Ly, N.-T. Do, S.-H. Kim, H.-J. Yang, and G.-S. Lee, "A novel 2d and 3d multimodal approach for in-the-wild facial expression recognition," *Image and Vision Computing*, vol. 92, p. 103817, 2019.

[115] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[116] A. Canziani, A. Paszke, and E. Culurciello, "An analysis of deep neural network models for practical applications," *arXiv preprint arXiv:1605.07678*, 2016.

[117] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 116–131.

[118] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6848–6856.

[119] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2820–2828.

[120] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[121] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[122] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[123] A. Gholami, K. Kwon, B. Wu, Z. Tai, X. Yue, P. Jin, S. Zhao, and K. Keutzer, "Squeezenext: Hardware-aware neural network design," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1638–1647.

[124] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4820–4828.

[125] A. Zhou, A. Yao, Y. Guo, L. Xu, and Y. Chen, "Incremental network quantization: Towards lossless cnns with low-precision weights," *International Conference on Learning Representations (ICLR), Volume abs/1702.03044*, 2017.

[126] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European conference on computer vision*. Springer, 2016, pp. 525–542.

[127] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2704–2713.

[128] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[129] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1314–1324.

[130] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[131] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64 270–64 277, 2018.

[132] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[133] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv:1412.6572*, 2014.

[134] S. Zheng, Y. Song, T. Leung, and I. Goodfellow, "Improving the robustness of deep neural networks via stability training," in *IEEE CVPR*, 2016, pp. 4480–4488.

[135] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *ICLR*, 2019.

[136] I. Vasiljevic, A. Chakrabarti, and G. Shakhnarovich, "Examining the impact of blur on recognition by convolutional networks," *arXiv:1611.05760*, 2016.

[137] R. Geirhos, C. R. Temme, J. Rauber, H. H. Schütt, M. Bethge, and F. A. Wichmann, "Generalisation in humans and deep neural networks," in *NeurIPS*, 2018, pp. 7538–7550.

[138] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *IEEE CVPR*, 2019, pp. 113–123.

[139] D. Yin, R. G. Lopes, J. Shlens, E. D. Cubuk, and J. Gilmer, "A fourier perspective on model robustness in computer vision," *arXiv:1906.08988*, 2019.

[140] R. Zhang, "Making convolutional networks shift-invariant again," *ICML*, 2019.

[141] N. Strisciuglio, M. Lopez-Antequera, and N. Petkov, "Enhanced robustness of convolutional networks with a push–pull inhibition layer," *Neural Computing and Applications*, pp. 1–15, 2020.

[142] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *IEEE CVPR*, 2017, pp. 1251–1258.

[143] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Int. Conf. on AIS*, 2010, pp. 249–256.

[144] O. M. Parkhi, A. Vedaldi, A. Zisserman *et al.*, "Deep face recognition." in *BMVC*, vol. 1, no. 3, 2015, p. 6.

[145] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Tech. Rep. 07-49, University of Massachusetts, Tech. Rep., 2007.

[146] V. Carletti, A. Greco, A. Saggese, and M. Vento, "An effective real time gender recognition system for smart cameras," *Journal of Ambient Intelligence and Humanized Computing*, 2019.

[147] R. Rothe, R. Timofte, and L. V. Gool, "Deep expectation of real and apparent age from a single image without facial landmarks," *International Journal of Computer Vision (IJCV)*, vol. 126, no. 2-4, pp. 144–157, Jul. 2016.

[148] 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.

[149] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[150] V. Carletti, A. Greco, G. Percannella, and M. Vento, "Age from faces in the deep learning revolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.

[151] S. Li and W. Deng, "Deep facial expression recognition: A survey," *arXiv preprint arXiv:1804.08348*, 2018.

[152] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[153] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," *University of Montreal*, vol. 1341, no. 3, p. 1, 2009.

[154] C. Shan, S. Gong, and P. W. McOwan, "Facial expression recognition based on local binary patterns: A comprehensive study," *Image and Vision Computing*, vol. 27, no. 6, pp. 803–816, 2009.

[155] A. Dhall, O. Ramana Murthy, R. Goecke, J. Joshi, and T. Gedeon, "Video and image based emotion recognition challenges in the wild: Emotiw 2015," in *ACM ICMI*, 2015, pp. 423–426.

[156] P. Ekman, E. R. Sorenson, and W. V. Friesen, "Pan-cultural elements in facial displays of emotion," *Science*, vol. 164, no. 3875, pp. 86–88, 1969.

[157] H. Ding, S. K. Zhou, and R. Chellappa, "Facenet2expnet: Regularizing a deep face recognition net for expression recognition," in *FG 2017*. IEEE, 2017, pp. 118–126.

[158] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE CVPR*, 2018, pp. 7132–7141.

[159] S. Albanie, A. Nagrani, A. Vedaldi, and A. Zisserman, "Emotion recognition in speech using cross-modal transfer in the wild," *arXiv:1808.05561*, 2018.

[160] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE CVPR*, 2017, pp. 4700–4708.

[161] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *IEEE ICPR*, vol. 1, 1994, pp. 582–585.

[162] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans Pattern Anal Mach Intell*, vol. 24, no. 7, pp. 971–987, 2002.

[163] Z. Yu and C. Zhang, "Image based static facial expression recognition with multiple deep network learning," in *ACM ICMI*, 2015, pp. 435–442.

[164] S. Lim, I. Kim, T. Kim, C. Kim, and S. Kim, "Fast autoaugment," *arXiv:1905.00397*, 2019.

[165] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Advances in Neural Information Processing Systems*, 2014, pp. 2654–2662.

[166] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2016, pp. 582–597.

[167] S. Ge, S. Zhao, C. Li, and J. Li, "Low-resolution face recognition in the wild via selective knowledge distillation," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 2051–2062, 2018.

[168] D. Deng, Z. Chen, and B. E. Shi, "Multitask emotion recognition with incomplete labels," in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)(FG)*, 2020, pp. 828–835.

[169] G. Antipov, M. Baccouche, S.-A. Berrani, and J.-L. Dugelay, "Apparent age estimation from face images combining general and children-specialized deep learning models," in *Proc. of IEEE Conf. on CVPR Workshops*, 2016, pp. 96–104.

[170] M. Uřičář, V. Franc, D. Thomas, A. Sugimoto, and V. Hlaváč, "Real-time multi-view facial landmark detector learned by the structured output svm," in *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, vol. 2. IEEE, 2015, pp. 1–8.

[171] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv:1408.5093*, 2014.

[172] I. Rafique, A. Hamid, S. Naseer, M. Asad, M. Awais, and T. Yasir, "Age and gender prediction using deep convolutional neural networks," in *2019 International Conference on Innovative Computing (ICIC)*, 2019, pp. 1–6.

[173] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation," *arXiv*, 2018.

[174] A. Greco, A. Saggese, M. Vento, and V. Vigilante, "A convolutional neural network for gender recognition optimizing the accuracy/speed tradeoff," *IEEE Access*, vol. 8, pp. 130 771–130 781, 2020.

[175] S. Escalera, J. Fabian, P. Pardo, X. Baró, J. Gonzalez, H. J. Escalante, D. Misevic, U. Steiner, and I. Guyon, "Chalearn looking at people 2015: Apparent age and cultural event recognition datasets and results," in *Proc. of IEEE ICCV*, 2015, pp. 1–9.

[176] S. Lapuschkin, A. Binder, K.-R. Müller, and W. Samek, "Understanding and comparing deep neural networks for age and gender classification," in *Proc. of IEEE ICCV*, 2017.

[177] Z. Tan, J. Wan, Z. Lei, R. Zhi, G. Guo, and S. Z. Li, "Efficient group-n encoding and decoding for facial age estimation," *IEEE Trans. on PAMI*, 2017.

[178] A. Dehghan, E. G. Ortiz, G. Shu, and S. Z. Masood, "Dager: Deep age, gender and emotion recognition using convolutional neural network," *arXiv:1702.04280*, 2017.

[179] Z. Huo, X. Yang, C. Xing, Y. Zhou, P. Hou, J. Lv, and X. Geng, "Deep age distribution learning for apparent age estimation," in *Proc. of IEEE Conf. on CVPR Workshops*, 2016, pp. 722–729.

[180] M. Uricar, R. Timofte, R. Rothe, J. Matas, and L. V. Gool, "Structured output svm prediction of apparent age, gender and smile from deep features," in *Proc. of IEEE Conf. on CVPR Workshops*, 2016, pp. 730–738.

[181] R. C. Malli, M. Aygun, and H. K. Ekenel, "Apparent age estimation using ensemble of deep learning models," in *Proc. of IEEE Conf. on CVPR Workshops*, 2016, pp. 714–721.

[182] M. Duan, K. Li, and K. Li, "An ensemble cnn2elm for age estimation," *IEEE Trans. on IFS*, pp. 758–772, 2018.

[183] F. Gurpinar, H. Kaya, H. Dibeklioglu, and A. Salah, "Kernel elm and cnn based facial age estimation," in *Proc. of IEEE Conf. on CVPR Workshops*, 2016, pp. 80–86.

[184] K. Zhang, C. Gao, L. Guo, M. Sun, X. Yuan, T. X. Han, Z. Zhao, and B. Li, "Age group and gender estimation in the wild with deep ror architecture," *IEEE Access*, pp. 22 492–22 503, 2017.

[185] L. Hou, D. Samaras, T. Kurc, Y. Gao, and J. Saltz, "Convnets with smooth adaptive activation functions for regression," in *Int. Conf. on Artificial Intelligence and Statistics*, 2017, pp. 430–439.

[186] L. Hou, C.-P. Yu, and D. Samaras, "Squared earth mover's distance-based loss for training deep neural networks," *arXiv:1611.05916*, 2016.

[187] H. Liu, J. Lu, J. Feng, and J. Zhou, "Label-sensitive deep metric learning for facial age estimation," *IEEE Trans. on Information Forensics and Security*, pp. 292–305, 2018.

[188] Z. Qawaqneh, A. A. Mallouh, and B. D. Barkana, "Deep convolutional neural network for age estimation based on vgg-face model," *arXiv:1709.01664*, 2017.

[189] J.-C. Chen, A. Kumar, R. Ranjan, V. M. Patel, A. Alavi, and R. Chellappa, "A cascaded convolutional neural network for age estimation of unconstrained faces," in *Proc. of IEEE Int. Conf. on BTAS*, 2016, pp. 1–8.

[190] G. Levi and T. Hassner, "Age and gender classification using convolutional neural networks," in *Proc. of CVPR Workshops*, 2015, pp. 34–42.

List of Figures

List of Tables