IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 21, 2020, accepted July 9, 2020, date of publication July 13, 2020, date of current version July 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3008793

A Convolutional Neural Network for Gender Recognition Optimizing the Accuracy/Speed Tradeoff

ANTONIO GRECO, (Member, IEEE), ALESSIA SAGGESE, (Member, IEEE), MARIO VENTO, (Member, IEEE), AND VINCENZO VIGILANTE
Department of Information and Electrical Engineering and Applied Mathematics, University of Salerno, 84084 Fisciano, Italy
Corresponding author: Alessia Saggese (asaggese@unisa.it)

This work was supported in part by the Italian MIUR within PRIN 2017 Grants, under Project 20172BH297 002 CUP D44I17000200005.

ABSTRACT Gender recognition has been among the most investigated problems in the last years; although several contributions have been proposed, gender recognition in unconstrained environments is still a challenging problem and a definitive solution has not been found yet. Furthermore, Deep Convolutional Neural Networks (DCNNs) achieve very interesting performance, but they typically require a huge amount of computational resources (CPU, GPU, RAM, storage), that are not always available in real systems, due to their cost or to specific application constraints (when the application needs to be installed directly on board of low-power smart cameras, e.g. for digital signage). In the latest years the Machine Learning community developed an interest towards optimizing the efficiency of Deep Learning solutions, in order to make them portable and widespread. In this work we propose a compact DCNN architecture for Gender Recognition from face images that achieves approximately state of the art accuracy at a highly reduced computational cost (almost five times). We also perform a sensitivity analysis in order to show how some changes in the architecture of the network can influence the tradeoff between accuracy and speed. In addition, we compare our optimized architecture with popular efficient CNNs on various common benchmark dataset, widely adopted in the scientific community, namely LFW, MIVIA-Gender, IMDB-WIKI and Adience, demonstrating the effectiveness of the proposed solution.

INDEX TERMS Convolutional neural network, deep learning, face analysis, gender recognition, efficiency, accuracy-speed tradeoff.

I. INTRODUCTION

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. One example is the intelligent social robotics, where the perception of soft biometric traits is used to personalize the conversation and increase the feel of intelligence perceived by the human interlocutor. Digital signage is another application where gender recognition can be profitably used, since it allows to boost the effectiveness of the advertisement campaigns; indeed, in this scenario it is possible to replace the static contents shown on the monitor with some dynamic advertisements, customized depending on the gender of the person looking at the monitor itself.

In both the examples provided above, the systems need to be capable to reliably work even ''in the wild'', where there are challenging conditions of illumination, uncontrolled pose variations, random occlusions, and even more variability of age, ethnicity, expression. Furthermore, the algorithm must be executed in real time, and often it is not possible to exploit cloud services, due to latency or the absence of a reliable connection to the internet, and a powerful server is rarely available due to its cost. Therefore, the gender recognition algorithm must run in real time on the processing units embedded in the robot, in the surveillance camera or in the digital billboard. However, the most accurate methods may need gigabytes of RAM and storage, and billions of floating point operations for a single prediction, while the available

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Zhang.

processing units, even being quite powerful, with 32 bit parallelism, vector co-processors and capabilities of floating point computation, are typically equipped with ARM processors and only low resources in terms of memory and storage. This is particularly true in applications like digital signage, in which a small smart camera, with around 512 MB of RAM and 16 MB for storing the whole application, including the model of the network, needs to perform the classification of several faces in real-time to quickly customize the promotional content.

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.

Within this context, we explicitly address both the above mentioned issues. We propose an optimal DCNN architecture specifically tuned for gender recognition. Similar challenges are nowadays faced in akin interactive, human-centered fields, such as autonomous driving [1], that require careful design of a real-time capable network architecture [2], [3].

In this work, 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 we want to build 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 well known standard benchmarks for gender recognition; in order to evaluate the performance in real environments, we chose some of them 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. Moreover, we perform a comparison with existing optimized architectures, namely Xception [4], Squeezenet [5] and Shufflenet [6], and we measure their prediction latency on a hardware architecture that is nowadays very common for middle or high-end embedded system; 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.

To summarize, the main contributions of this paper are the following: 1) we demonstrate with a comprehensive experimental analysis that it is possible to preserve the gender recognition accuracy by carefully modifying the architecture of a CNN; 2) we propose a network architecture specifically devised for gender recognition, optimized by reducing the input size, the number of feature maps and the number of layers of an existing network architecture, achieving a performance comparable with state of the art but can be suitably applied in embedded applications with real-time constraints.

II. RELATED WORK

The typical pipeline for a gender recognition system is shown in Figure 1 and consists of the following main steps: (1) face detection; (2) face normalization/alignment; (3) feature extraction and classification. In the first step, the position of the face in the image is identified with model based approaches, such as [7], [8] and [9]. The face detection is typically the processing step which requires more time. In [10] the authors propose an architecture for reducing the space of the image where looking for faces. They compute the time for detection by using the well-known Viola Jones algorithm, with an average time of 428 ms on the target embedded device. Anyway, for face classification, the time required for classifying a single face by using deep learning based approaches may vary from 40 up to 2000 ms for most accurate methods available in the literature. It is also important to specify that the processing time scales linearly with the number of people, and this factor represents a challenge for gender recognition in crowded scenarios. It implies that the classification time needs to be considered as well, since it may become a very critical part of the face analysis process.

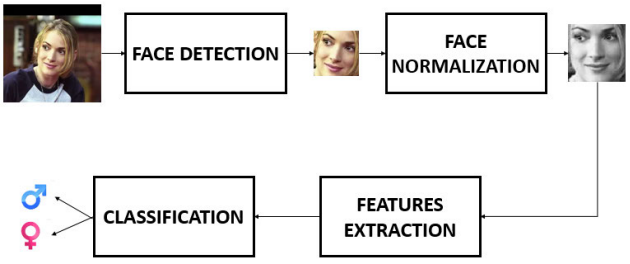

FIGURE 1. Functional processing pipeline of a typical gender recognition system. Note that some functions may be also absent or aggregated.

In the second step, the facial landmarks are found inside the face region. The facial landmarks are known points in the face that are easy to identify for a human: typically the tip of

the nose and the centers of the eyes are used. Once the facial landmarks are identified, the image is scaled and rotated to put the eyes and the nose in fixed locations; the procedure may vary and more sophisticated methods may exploit more landmarks to perform different transformations, such as a full frontalization [11] that tries to compensate for pose variations. However, such methods have significant drawbacks and are not as popular as plain affine transformation, since they are extremely slow and introduce a consistent deformation of the face. Other normalization steps may be also applied, such as contrast stretching and histogram equalization.

In the third step, the actual classification takes place; the features are extracted from a pre-processed face image and a binary decision (male/female) is the output. Three main strategies have been adopted for classification: (1) handcrafted features, (2) trainable features or (3) a combination of them.

Handcrafted features are carefully designed by humans explicitly for the problem at hand, while trainable features are general purpose meta-descriptors that can be learned from examples. The handcrafted features can take advantage of the domain-specific knowledge to be more accurate and efficient. For gender recognition, it has been shown that color [12], shape [13], texture [14] and local features [15] are typically discriminant. It has also been shown that a combination of those features would lead to significant improvements in recognition accuracy [16]. SVMs is often used [17] as classification step. Using a variant of SVM on multiscale LBP texture features, the authors of [18] achieve 96.6% performance on the well known LFW benchmark. While handcrafted-feature based systems often leverage the full pipeline, sometimes in practical application face alignment is just skipped. Indeed, the improvement in the accuracy is paid in a more significant improvement in terms of the computational burden required [19]. Furthermore, any failures of the alignment algorithm may affect the overall system performance.

Trainable features, on the contrary, do not leverage domain-specific information when they are chosen and designed, but they can themselves learn particular patterns that are not immediately evident to human designers. The approach based on trainable features includes all the techniques related to deep convolutional neural networks that learn the filters directly from the data. These techniques were proved in recent years to be very effective on all the computer vision tasks, and in particular on those related to face analysis, such as face recognition and re-identification [20], soft biometrics such as gender [21], age [22], and so on. It is worth noting that this approach was not born with deep learning, but it has been already applied to gender recognition from still images in different forms. In [23], for example, the authors use a weighted combination of shifted Gabor filters, inspired by the structure of the visual cortex: the parameters of those filters are chosen with an automatic pseudo-random training procedure in which the images from the application domain are fed into the filters.

Among the trainable-feature based methods, the authors of [24] 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. VGG architecture has been also used in [25], where the authors compare MobileNet and VGG in the field of social robotic. In a successive work [26] they train the very deep an powerful ResNet-50 CNN and obtain the state of art accuracy of 99.3% on the LFW benchmark; the network is pretrained on the problem of face recognition and then it is fine tuned on the IMDB-WIKI-cleaned dataset.

Some recent methods even perform all the three steps together (detection, aligment and classification): for instance, in [27] the authors achieve 94% accuracy on the LFW dataset, training a CNN jointly for face landmarks, pose estimation and gender recognition.

III. PROPOSED METHOD

Our method is based on a multi-purpose neural network architecture named MobileNets [28], [29]. 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. This trick allows 3×3 convolutions to require 8 to 9 times less operations, with a consequent reduction in the number of parameters [28]. In [29], 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, the residual connection from [30] 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).

According to [31] the biggest variant of MobileNets achieves a high accuracy on the problem of object recognition while keeping a low latency. The architecture has been tested in different variants to trade latency for accuracy. This network was experimented to be quantized to obtain further improvement in time and memory consumption on low power devices with negligible performance loss [32].

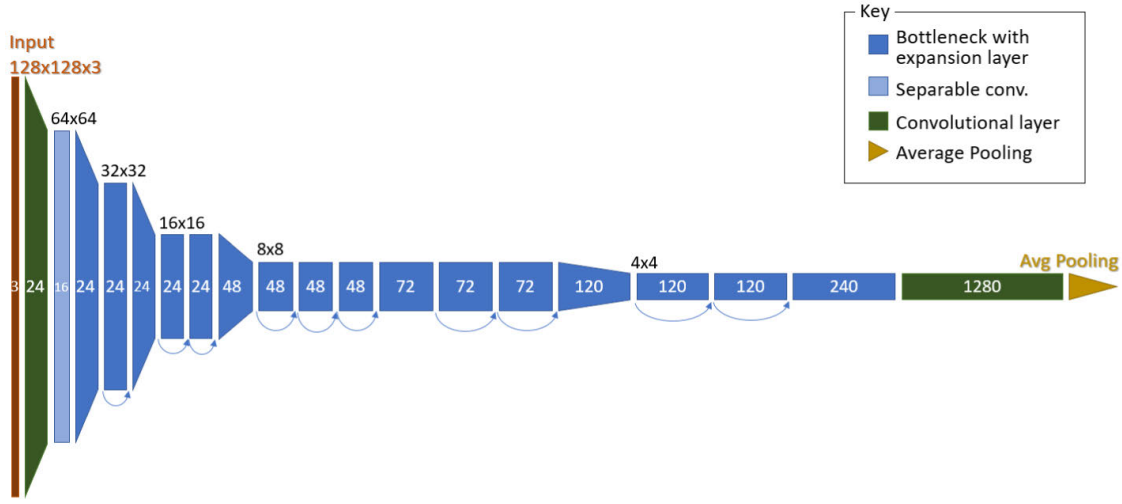

FIGURE 2. The original MobileNets v2 architecture (width multiplier = 0.5, input size = 128).

A. MINIMIZATION

We will experiment different variants of that architecture to find out how the performance is affected. The variants that we will consider, as reported in Table 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.

TABLE 1. Different changes 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

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 [28]. 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 [24] 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 [22]. 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 IV-D 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.

B. TRAINING

All the network architectures are trained from scratch. We decided to adopt the most widely used parameters initialization technique, namely the Xavier Uniform method [33], which allows the neural networks to achieve quick convergence and high accuracy in several computer vision tasks; we did not use different initialization methods, since the aim of this experimental analysis is to compare the performance of different gender recognition methods trained with the same

TABLE 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			

protocol. 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: when loading an image, it 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. We use the Adam optimizer with parameters b1=0.9, b2=0.999, decay=5e-5. Also, in order to reduce the overfitting we use a dropout rate; we experimented 9 values between 0.1 and 0.9 with step 0.1 and noticed that setting it to 0.2 allows to maximize the accuracy on the validation set. The dropout is inserted between the last convolutional layer and the last fully connected layer, as typically done in literature.

C. PREPROCESSING

In this work our focus is on the classification step; anyway, it is still important for the sake of completeness to describe the preprocessing steps, both in terms of face detection and normalization/alignment, that affect the type of images fed into our classifier and the latency of the overall system.

As for the detection step, we adopt the well-known Viola Jones face detector [7], 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: indeed, even if it can converge faster in the training phase, the performance improvement is limited since it can only fix in-plane rotation. Since the common alignment algorithms have a significant effect on latency, we choose to completely drop the alignment 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. Nearest-Neighbour resampling would produce significant artifacts on the images with lower resolution, so we decided to avoid it. Bicubic resampling would produce visually similar results in the spatial and frequency domain, so we decided to go with bilinear, that is simpler.

IV. EXPERIMENTS

We perform a comprehensive experimental analysis on several public datasets; we describe them in Section IV-A, while in Section IV-B 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 IV-C we describe, at various input resolutions, the effect of decreasing the number of feature maps; in Subsection IV-D 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 IV-E we compare our proposed solution with other architectures on the considered datasets. Finally, in Subsection IV-F we analyze the results in real environments and show how our approach is able to succeed in the target applications while different solutions fail.

A. DATASETS

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

1) VGGFACE

The VGGFace dataset [20] 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 [34], 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 purposes.

2) LFW DATASET

The LFW dataset [35] 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 [18], [24] and [26].

3) MIVIA-GENDER DATASET

The MIVIA-Gender dataset [10] 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.

4) IMDB-WIKI DATASET

The IMDB-WIKI dataset [36] 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.

5) ADIENCE DATASET

The Adience dataset [19] 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.

B. 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 [24], 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 [26] (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.

C. 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 3. For this evaluation, we will adopt the notation x_y, where x is the input size 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

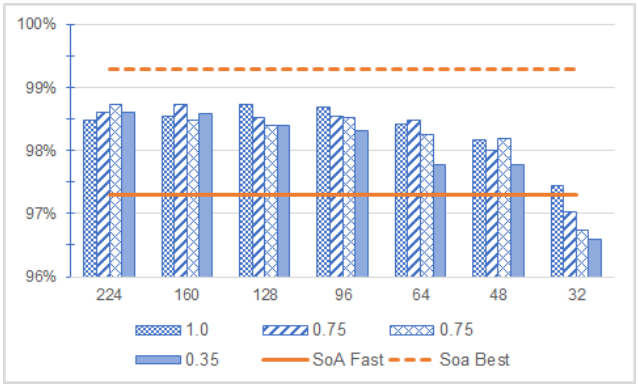

FIGURE 3. Classification accuracy vs.input size (224, 160, ...) and width multiplier (1.0, 0.75, ...) on the LFW dataset. According to the notation we used in the paper, the first bar (starting from the left) corresponds to 224_1.0, the second one to 224_0.75 etc. On the chart we also display two main results of the state of the art for comparison, namely SoA Fast [24] and SoA Best [26]. More details are reported in Section IV-C.

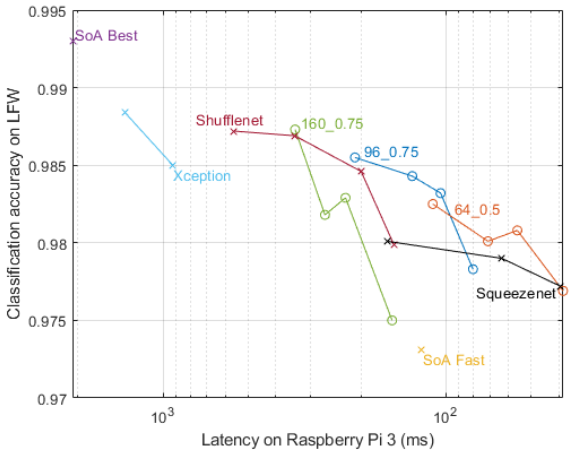

FIGURE 4. Scatter plot of latency versus accuracy on the LFW dataset. For our proposed architectures (circles), each line represents a different combination between input size and width multiplier and every point indicates a different number of blocks. The other points (crosses) represent variants of different architectures we compare with.

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.

D. 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 4 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 III-B, is not sufficient to avoid that. Another cause may be the fact that, having a larger input resolution, the last convolutional 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.

E. 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* [24] and *SoA Best* [26], 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 [4] that improves over the popular Inception architecture using depthwise convolution, like in our proposed architecture. Then, we experiment the Squeezenet architecture [5], that is thought for embedded systems, even though it does not directly optimize the processing speed with respect to the classification accuracy. Finally we

TABLE 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.

Model	Latency (ms)	Accuracy (%)							
		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

experiment ShufflenetV2 [6], 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 4 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* [26], 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* [26], in order to explain the performance gap on the LFW dataset. In our case no pretraining is performed, while the authors of [26] prove that a face recognition pretraining significantly improves classification of accuracy of the final model. Then, we use VGGFace2 as training dataset, while [26] 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 III-B. The difference is confirmed by the fact that when we train the architectures from [26] 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 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 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.

F. 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* and Xception, which use ResNet-50, 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 5 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 5. Samples of misclassifications on the LFW test set. Aside of faces with poor gender features, most of the few errors concern children, elders and Asians. Faces in the first row were miclassified as males, while the ones in the second row were mistaken for females.

V. CONCLUSION

Even if in the future we expect the presence on the market of high end embedded platforms equipped with neural network accelerator chips, in the current market most of the devices, such as smart cameras or commercial robotic platforms for social and smart applications, only rely on low power general purpose CPUs. In this work we have shown how a very accurate gender recognition system (up to 98.73% in the *wild*) can run in real time on an embedded device, without the use of dedicated hardware such as a GPU or other type of parallel computation accelerators. We leveraged many features of the modern deep learning state of the art that include separable convolution and residual blocks to train a convolutional deep neural network that would reliably recognize gender from images. We started from the MobileNet architecture that is already known and designed to be a fast and efficient CNN model and we experimented different changes of the architecture to find a trade off between processing time and recognition accuracy. The changes regarded the input size, the number of feature maps and the number of layers. The rationale behind this choice is that very low resolution images are still typically sufficient for a human to determine gender. We found out that even with few feature maps and a reduced layer hierarchy, there is no significant performance drop (up to 97.70%).

Future work will include specifically addressing the problem of accuracy drop when classifying faces of elders, children and Asians. Furthermore we plan to consider problems generated by detection errors: these mistakes are just mentioned in this work, but may represent a significant source of error in a real system. We also plan to extend our analysis to other soft biometrics, such as age, expression, emotion, ethnicity or facial attributes [37]. Finally, we will investigate other techniques to further reduce the processing time, such as pruning, weight quantization and single stage face detection and gender recognition.

REFERENCES

[1] 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 *Proc. Int. Conf. Neural Inf. Process*. Cham, Switzerland: Springer, 2019, pp. 596–607.

[2] 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 *Proc. Int. Conf. Neural Inf. Process*. Cham, Switzerland: Springer, 2018, pp. 172–181.

[3] P. M. Kebria, A. Khosravi, S. M. Salaken, and S. Nahavandi, "Deep imitation learning for autonomous vehicles based on convolutional neural networks," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 1, pp. 82–95, Jan. 2020.

[4] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.

[5] 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," 2016, *arXiv:1602.07360*. [Online]. Available: http://arxiv.org/abs/1602.07360

[6] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 116–131.

[7] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Dec. 2001, p. 1.

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

[9] S. Liao, A. K. Jain, and S. Z. Li, "A fast and accurate unconstrained face detector," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 211–223, Feb. 2016.

[10] V. Carletti, A. Greco, A. Saggese, and M. Vento, "An effective real time gender recognition system for smart cameras," *J. Ambient Intell. Hum. Comput.*, vol. 11, no. 6, pp. 2407–2419, Jun. 2020.

[11] T. Hassner, S. Harel, E. Paz, and R. Enbar, "Effective face frontalization in unconstrained images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4295–4304.

[12] B. Moghaddam and M.-H. Yang, "Learning gender with support faces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 707–711, May 2002.

[13] V. Singh, V. Shokeen, and M. B. Singh, "Comparison of feature extraction algorithms for gender classification from face images," *Int. J. Eng. Res. Technol.*, vol. 2, no. 5, pp. 1313–1318, 2013.

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

[15] 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 *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2017, pp. 1–6.

[16] G. Azzopardi, P. Foggia, A. Greco, A. Saggese, and M. Vento, "Gender recognition from face images using trainable shape and color features," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 1983–1988.

[17] G. Azzopardi, A. Greco, and M. Vento, "Gender recognition from face images using a fusion of SVM classifiers," in *Proc. ICIAR*, 2016, pp. 533–538.

[18] S. Jia and N. Cristianini, "Learning to classify gender from four million images," *Pattern Recognit. Lett.*, vol. 58, pp. 35–41, Jun. 2015.

[19] E. Eidinger, R. Enbar, and T. Hassner, "Age and gender estimation of unfiltered faces," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 12, pp. 2170–2179, Dec. 2014.

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

[21] G. Azzopardi, A. Greco, A. Saggese, and M. Vento, "Fusion of domain-specific and trainable features for gender recognition from face images," *IEEE Access*, vol. 6, pp. 24171–24183, 2018.

[22] V. Carletti, A. Greco, G. Percannella, and M. Vento, "Age from faces in the deep learning revolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Apr. 11, 2019, doi: 10.1109/TPAMI.2019.2910522.

[23] G. Azzopardi, A. Greco, and M. Vento, "Gender recognition from face images with trainable COSFIRE filters," in *Proc. 13th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2016, pp. 235–241.

[24] G. Antipov, S.-A. Berrani, and J.-L. Dugelay, "Minimalistic CNN-based ensemble model for gender prediction from face images," *Pattern Recognit. Lett.*, vol. 70, pp. 59–65, Jan. 2016.

[25] P. Foggia, A. Greco, G. Percannella, M. Vento, and V. Vigilante, "A system for gender recognition on mobile robots," in *Proc. 2nd Int. Conf. Appl. Intell. Syst. (APPIS)*, 2019, p. 9.

[26] 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 Recognit.*, vol. 72, pp. 15–26, Dec. 2017.

[27] R. Ranjan, V. M. Patel, and R. Chellappa, "HyperFace: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 121–135, Jan. 2019.

[28] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861

[29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation," 2018, *arXiv:1801.04381*. [Online]. Available: https://arxiv.org/abs/1801.04381

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[31] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64270–64277, 2018.

[32] R. Krishnamoorthi, "Quantizing deep convolutional networks for efficient inference: A whitepaper," 2018, *arXiv:1806.08342*. [Online]. Available: http://arxiv.org/abs/1806.08342

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

[34] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "VGGFace2: A dataset for recognising faces across pose and age," in *Proc. 13th IEEE Int. Conf. Automat. Face Gesture Recognit. (FG)*, May 2018, pp. 67–74.

[35] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," in *Proc. Workshop Faces Real-Life Images, Detection, Alignment, Recognit.*, 2008, pp. 1–15.

[36] R. Rothe, R. Timofte, and L. V. Gool, "Deep expectation of real and apparent age from a single image without facial landmarks," *Int. J. Comput. Vis.*, vol. 126, pp. 144–157, Jul. 2016.

[37] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3730–3738.

ANTONIO GRECO (Member, IEEE) received the Ph.D. degree in computer science and computer engineering from the University of Salerno, in 2018. He is currently an Assistant Professor with the University of Salerno. His research activity is focused on computer vision and machine learning techniques for video surveillance applications. He serves as a Referee for many journals and international conferences.

ALESSIA SAGGESE (Member, IEEE) received the Ph.D. degree in computer engineering from the University of Salerno, Italy, and the University of Caen, France, in 2014. She is currently an Assistant Professor with the University of Salerno. Her research interests include basic methodologies and applications in computer vision and pattern recognition. She is a member of the International Association for Pattern Recognition Technical Committee 15 on Graph-Based Representations in Pattern Recognition, since 2012. She is also an Associate Editor of the IEEE ACCESS JOURNAL.

MARIO VENTO (Member, IEEE) received the Ph.D. degree in computer engineering from the University of Napoli, in 1989.

He is currently a Full Professor of artificial vision, machine learning, and cognitive robotics with the University of Salerno, Italy, where he is also the Coordinator of the Artificial Vision Laboratory (MIVIA Lab). His research activities cover real-time video analysis and interpretation, cognitive robotics, classification techniques, exact and inexact graph matching, and learning methodologies for structural descriptions. He is a Fellow Scientist of the International Association Pattern Recognition (IAPR). He has served as the Chairman of the IAPR Technical Committee 15 on Graph-Based Representation in Pattern Recognition, from 2002 to 2006. He is also an Associate Editor of the *Pattern Recognition Journal*.

VINCENZO VIGILANTE received the degree *(cum laude)* in computer engineering from the University of Salerno, in September 2017, where he is currently pursuing the Ph.D. degree. His research activity is focused on computer vision and machine learning techniques for intelligent robotics and video surveillance applications.