Investigating Nuisances in DCNN-Based Face Recognition

Claudio Ferrari, Giuseppe Lisanti, Stefano Berretti, *Senior Member, IEEE*,
and Alberto Del Bimbo, *Senior Member, IEEE*

***Abstract*—Face recognition "in the wild" has been revolutionized by the deployment of deep learning-based approaches. In fact, it has been extensively demonstrated that deep convolutional neural networks (DCNNs) are powerful enough to overcome most of the limits that affected face recognition algorithms based on hand-crafted features. These include variations in illumination, pose, expression, and occlusion, to mention some. The DCNNs discriminative power comes from the fact that low- and high-level representations are learned directly from the raw image data. As a consequence, we expect the performance of a DCNN to be influenced by the characteristics of the image/video data that are fed to the network, and their preprocessing. In this paper, we present a thorough analysis of several aspects that impact on the use of DCNN for face recognition. The evaluation has been carried out from two main perspectives: the network architecture and the similarity measures used to compare deeply learned features; and the data (source and quality) and their preprocessing (bounding box and alignment). The results obtained on the IARPA Janus Benchmark-A, MegaFace, UMDFaces, and YouTube Faces data sets indicate viable hints for designing, training, and testing DCNNs. Considering the outcomes of the experimental evaluation, we show how competitive performance with respect to the state of the art can be reached even with standard DCNN architectures and pipeline.**

***Index Terms*—Face recognition, deep learning, CNN architecture, distance measures.**

I. Introduction

THE idea of using images of the face as biometric signatures to perform identity recognition dates back to the mid '60s, with the pioneering work of Bledsoe *et al.* [1]. Since that, most of the research on face recognition focused on the definition of hand-crafted features (also referred to as "shallow" features) capable of capturing the traits of the face that best discriminate one subject from the others. For many years, these methods have been experimented on images acquired in cooperative contexts (indoor laboratories in most of the cases), with controlled conditions and a quite limited variability in terms of different identities, image resolution, pose and illumination changes, etc. The shift from cooperative to uncooperative datasets, acquired in the wild [2], contributed to substantially advance the research in this field. However, solutions based on classical learning methods and shallow features showed to be still quite not ready to cope with the large variability that occurs in the reality.

The success of a Deep Convolutional Neural Network architecture (DCNN) in the ImageNet 2012 Large Scale Visual Recognition Challenge has radically changed the scenario [3]. Though CNNs were known since mid '80s, their effective deployment in real application contexts has been not possible till massive computation infrastructures and large quantities of data were available for training. In fact, one substantial innovation of DCNNs is the idea of letting the deep architecture to automatically discover low-level and high-level representations from labeled and/or unlabeled training data, which can then be used for detecting and/or classifying the underlying patterns. After such re-discovering, CNNs have found application in an ever increasing number of different Computer Vision problems, such as object detection and image classification at large scale [3], [4], scene [5] and action [6], [7] recognition, significantly improving the state-of-the-art.

Quite a large consensus has been also reached in the research community that DCNNs can provide the right tool to perform face recognition in real and challenging conditions. Indeed, breakthrough results have been obtained using such technology on most of the existing benchmark datasets [8]–[12]. However, though the proliferation of deep learning based solutions for face recognition, there are several aspects of their behavior that remain not completely understood or that have not been investigated at all. These concern the networks architecture as well as the source and preprocessing of training and testing data. On the one hand, designing a DCNN is by itself a complicated task since many decisions are involved, for example: how many layers and filter banks (*e.g.*, how much deepening the multi-scale analysis through subsequent convolutions is important)? which loss function (*e.g.*, triplet- or center-loss)? On the other, data and their preprocessing are crucial in both training and testing. Training of DCNNs is data driven so it is important to understand: how many data (*e.g.*, more subjects or more instances per subject)? which variability (*e.g.*, pose, illumination, resolution, distractors and label noise)? which sources

Manuscript received January 8, 2018; revised June 25, 2018 and July 25, 2018; accepted July 25, 2018. Date of publication July 31, 2018; date of current version August 21, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jie Liang. *(Corresponding author: Claudio Ferrari.)*

C. Ferrari, S. Berretti, and A. Del Bimbo are with the Department of Information Engineering, University of Florence, 50139 Florence, Italy (e-mail: claudio.ferrari@unifi.it; stefano.berretti@unifi.it; alberto.delbimbo@unifi.it).

G. Lisanti is with the Department of Electrical, Computer and Biomedical Engineering, University of Pavia, 27100 Pavia, Italy (e-mail: giuseppe.lisanti@unipv.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2018.2861359

(*e.g.*, 2D still, video frames, both)? which preprocessing (*e.g.*, alignment, bounding-box size)? which approach for data augmentation (*e.g.*, synthetic data generation [13], [14])? Just a few works provided preliminary answers to some of the above questions [15]–[17].

In this work, we present a thorough study on DCNNs based face recognition, which intends to demonstrate that, by accounting for some aspects that in the literature are somewhat taken for granted, we can boost the performance of CNN architectures trained with a simple softmax classifier, reaching comparable recognition accuracy with respect to the state-of-the-art using a standard recognition pipeline. We start by describing the face recognition pipeline, which employs the networks as face descriptor extractors. For the sake of generality, we consider diverse CNN architectures, namely, Vgg-vd16 [10], Vgg-vd19 [18], ResNet-101 [19] and AlexNet [3]. We use the four architectures above to evaluate the effects of data characteristics (resolution), data preprocessing (bounding-box and alignment of the face), and data sources (still images or video frames) on the networks training and testing, and on the ultimate recognition accuracy. Furthermore, we propose and evaluate different solutions for comparing DCNN features in the case of face recognition from image sets or videos. Experimentations have been performed on image and video datasets acquired in the "wild" (IJB-A, MegaFace, UMDFaces, YouTube faces). In summary, the main contributions and outcomes of this work are: *(i)* we select diverse DCNN architectures for train and test, and compare them in combination with different distance measures for face recognition in the "wild"; *(ii)* we explore the effect induced by data characteristics, data preprocessing and data source on the different DCNN architectures; *(iii)* we introduce and experiment a new protocol for face recognition from heterogeneous data, *i.e.*, still images and videos on the UMD dataset. The points above extend and improve over our preliminary ideas and results presented in [16], where we mainly focused on the effect of preprocessing on DCNN-based face recognition.

The rest of the paper is organized as follows: in Sect. II, we revise some of the most influencing works that use deep learning to perform face recognition; The DCNN-based recognition pipeline, the network architectures and the datasets used in this work are reported in Sect. III; Sect. IV presents the distance measures used to match DCNN descriptors; In Sect. V, we discuss how the bounding box dimension, alignment and resolution affect the recognition, while in Sect. VI, we present an extensive analysis of different data sources; A comprehensive comparison with respect to state-of-the-art solutions is given in Sect. VII; Finally, in Sect. VIII, we report discussion and conclusions.

II. Related Work

In the last few years, the scenario in face recognition has been drastically changed by the combined availability of increasing computational resources and of very large datasets that made possible the effective training of neural networks with a deep architecture. In the following, we revise some recent works that use DCNN architectures for face recognition.

Taigman *et al.* [8] proposed DeepFace, a DNN architecture for face recognition. DeepFace comprised more than 120 million parameters using locally connected layers without weight sharing, rather than the standard convolutional layers. This network was trained on an identity labeled dataset of 4 million facial images belonging to more than $4,000$ identities. Explicit 3D face modeling was used to align the images using a piecewise affine transformation. The learned representations, coupling the accurate model-based alignment with the large facial database, generalized well to faces in unconstrained environments, even with a simple classifier. Sun *et al.* [9] proposed to learn a set of high-level feature representations through deep learning for face verification. These features, referred to as Deep hidden IDentity features (DeepID), were learned through multi-class face identification tasks, whilst they can be generalized to other tasks (such as verification) and new identities unseen in the training set. DeepID features were taken from the last hidden layer neuron activations of DCNN. When learned as classifiers to recognize about $10,000$ face identities in the training set, and configured to keep reducing the neuron numbers along the feature extraction hierarchy, these DCNNs gradually form compact identity-related features in the top layers with only a small number of hidden neurons. These features were extracted from various face regions to form complementary and overcomplete representations. The FaceNet system proposed by Schroff *et al.* [11] learned a mapping from face images to a compact Euclidean space, where distances directly correspond to a measure of face similarity. Once this space is obtained, tasks such as face recognition, verification and clustering were implemented using standard techniques with FaceNet embedding as feature vectors. A DCNN was trained to directly optimize the embedding itself, rather than an intermediate bottleneck layer as in previous deep learning approaches. Triplets of roughly aligned matching / non-matching face patches generated using an online triplet mining method were used for training, with the main benefit of a better representation efficiency. State-of-the-art face recognition performance was obtained using only 128-bytes per face. In the work of Parkhi *et al.* [10], a much simpler and yet effective network architecture (called VggFace) achieving near state-of-the-art results on all popular image and video face recognition benchmarks was proposed. On the one hand, they showed how a very large scale dataset (2.6M images of over 2.6K people) can be assembled by a combination of automation and human in the loop, and discussed the trade off between data purity and time. On the other, they traversed through the complexities of deep network training and face recognition.

The work of Masi *et al.* [20], addressed unconstrained face recognition in the wild focusing on the problem of extreme pose variations. As opposed to other techniques that either expect a single model to learn pose invariance through massive amounts of training data, or normalize images to a single frontal pose, this method explicitly tackled pose variation by using multiple pose specific models and rendered face images. DCNNs were used to learn discriminative representations, called Pose-Aware Models (PAMs) using 500K images from the CASIA WebFace dataset [21]. In a comparative evaluation, PAMs achieved better performance than commercial products also outperforming methods that are specifically fine-tuned on the target dataset. Hu *et al.* [22] explored how the fusion of face recognition features and facial attribute features can enhance face recognition performance in various challenging scenarios. This is obtained by developing a tensor-based framework, which formulates feature fusion as a tensor optimisation problem. Due to the large number of parameters to optimise, a theoretical equivalence is established between low-rank tensor optimisation and a two-stream gated neural network. This equivalence allows tractable learning using standard neural network optimisation tools, leading to accurate and stable optimisation. Experimental results show that the fused feature works better than individual features. Wang *et al.* [23] addressed the challenge of face recognition in a large collection of unconstrained face images by proposing a system which combines, in a cascaded framework, a fast search procedure with a state-of-the-art commercial off the shelf (COTS) matcher. Given a probe face, first a large gallery of photos (containing 80M web-downloaded face images) is filtered to find the top-k most similar faces using features learned by a CNN. Then, the k retrieved candidates are re-ranked by combining similarities based on deep features and those output by the COTS matcher.

Rather than focusing on the development of novel architectures and methods, Ghazi and Ekenel [15] presented a comprehensive study that evaluated the performance of deep learning based face representation under several conditions, including the varying head pose angles, upper and lower face occlusion, changing illumination of different strengths, and misalignment due to erroneous facial feature localization. Face representations were extracted using two successful and publicly available deep learning models, namely, VggFace [10] and Lightened CNN [24]. Images acquired in controlled conditions were used in the experiments. The obtained results showed that although deep learning provides a powerful representation for face recognition, it can still benefit from preprocessing, for example, for pose and illumination normalization. From this study it emerged that if variations included in test images were not included in the dataset used to train the deep learning model, the role of preprocessing became more important. Experimental results also showed that deep learning based representation is robust to misalignment and can tolerate facial feature localization errors up to 10% of the inter-ocular distance. Banerjee *et al.* [25], focused on the need of frontalization in face recognition with a VggFace pre-trained architecture. They concluded that the usefulness of frontalization to pre-process test set faces can be dependent on the facial recognition system used, how it was trained, and the failure threshold set; a simple 2D-alignment might be more productive in some cases.

A set of questions that are critical to face recognition research using CNNs have been explored also by Bansal *et al.* [17]: *(i)* Can we train on still images and expect the systems to work on videos? *(ii)* Are deeper datasets better than wider datasets? *(iii)* Does adding label noise lead to improvement in performance of deep networks? *(iv)* Is alignment needed for face recognition? The authors investigated on these questions considering a single architecture derived from AlexNet.

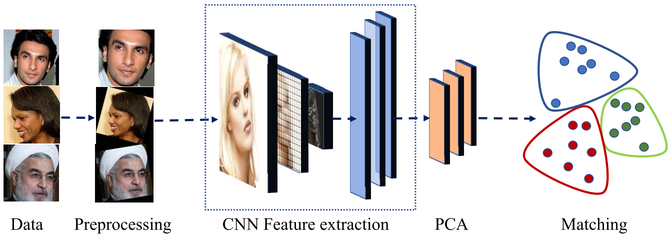


Fig. 1. The recognition pipeline used in this work.

III. Face Recognition With DCNN

To perform face recognition, we followed the reference pipeline depicted in Fig. 1, in which a DCNN is used as feature extractor; specifically, the output of the last fully connected layer is used as face descriptor, which is derived from the images and their horizontally flipped version. The final descriptor is obtained as the average of the two. Training set descriptors are used to compute a PCA projection and apply a dimensionality reduction on the test set. It is worth noting that applying a PCA to the learned descriptors, though not mandatory, is a quite well established practice in the face-recognition literature [8], [26]. This serves to improve the discriminative power of the final descriptor by removing noisy components, while also reducing the complexity. Finally, matching is performed employing the *cosine* distance between descriptors. Depending on the dataset and the protocols, matching may be either between single images, full video sequences or templates (mixed sets of both images and frames of the same subject). Consequently, from all the possible image pairs, a single distance (or score) must be determined in order to classify the identity.

In the following, we aim to investigate the effects that different choices in each component of the above pipeline have on face recognition. In particular, we will focus on the network architectures, the distance measures used to compare DCNN descriptors, the data and their preprocessing. We start by giving more details on the reference network architectures used in this work (Section III-A), and the datasets (Section III-B).

A. Network Architectures

First we need to identify network architectures that are well known and that provide good results for face recognition. In addition, we desire architectures that can be easily modified and retrained. To this end, we chose AlexNet [3], Vgg-vd19 [18] and ResNet-101 [19]. The networks were trained as face classifiers considering the data of 2, 623 unique individuals as collected in [10] and the empirical *softmax log-loss* function was used to train the classifier. In order to explore a wider range of networks and generalize the following claims, two pre-trained, publicly available models were also tested, namely, VggFace [10] and the version of ResNet-101 presented in [14], that we will refer to as ResNet-101-Synth.

1) AlexNet: The architecture of this network takes a 227 × 227 image as input, and is made up of 8 layers, 5 convolutional (Conv) and 3 fully connected (FC), each one followed by a rectification layer (ReLU). Max pooling is applied after the second and the fifth Conv layers. Three FC layers follow, and the output of the final FC is fed to a 2, 623-way softmax, which produces a distribution over the classes. Augmentation based on both randomly flipping and cropping the images was applied during training. We trained several versions of this network from scratch.

2) VggFace/Vgg-vd19: We also considered the VggFace pre-trained network built upon the Vgg-vd16 architecture that has been released by Parkhi *et al.* [10]. This network takes a 224 × 224 input image and has 8 convolutional blocks, each one followed by a ReLU. Max pooling is applied every 2 Conv layers till layer 10, then every 3. The last 3 blocks are FC layers and, similarly to AlexNet, the output of the last FC is fed to the softmax layer. No alignment has been applied to the face images used for training. Augmentation based on both randomly flipping and cropping the images was applied during training. The Vgg-vd19 architecture is a deeper version of the VggFace; it comprises 3 more Conv layers, placed in correspondence of the third, fourth and fifth block of the 16-layers version. We trained the Vgg-vd19 from scratch as well.

3) ResNet-101/ResNet-101-Synth: This network presents a peculiar architecture; it is built upon the work of [19] and grounds on the idea of fitting a residual mapping with respect to the input, rather than the input itself. Basically, the input is passed through a set of 3 Conv layers and, before going further through the layers, the result is summed with the input. The particular implementation used in this study has 101 layers and the residual is computed every 3 layers. Differently from the previous, we fine-tuned the ImageNet pre-trained version on 1M face images from [10]. Other than the latter, the pre-trained ResNet-101-Synth from [14] has also been experimented. The main reason that guided us in choosing this particular network is that it was trained on renderings, rather than on real images. Such renderings (224 × 224) were constructed considering a set of 3D shapes and textures that were mixed and rendered at various poses and expressions. This strategy permitted to both increase the intra-class variability in the training set and to artificially generate examples when real data is hardly available, as for extreme poses.

The choice of the different networks reported above was mainly intended to confirm that our subsequent analysis and conclusions can be generalized to very diverse architectures. In particular, we explored both shallower or deeper architectures in terms of number of layers, and lower or greater complexity in terms of number of parameters, as summarized in Table I. Indeed, exploring these two dimensions, *i.e.*, number of layers and parameters, permitted us to evidence that they act quite independently on the network learning capabilities [19] (*i.e.*, comparable face recognition performance can be achieved with less layers, but more parameters, or *vice versa*). Finally, such different designs also demonstrated us to have an impact on the practice of networks training, inducing considerable variations in the required computational resources, both in terms of memory and time.

TABLE I
NUMBER OF LAYERS AND PARAMETERS OF THE NETWORKS

Network	**# Layers**	**# Parameters**
AlexNet	8	66.8M
VggFace	16	144.9M
Vgg-vd19	19	149.6M
ResNet-101	101	47.5M

B. Datasets

To stress the networks dependence on the data, and derive a reasonable certainty that our results and conclusions are generalizable, we performed experiments on four datasets with different characteristics: the IARPA Janus Benchmark-A (IJB-A) [27], the YouTube Faces (YTF) [28], the University of Maryland (UMDFaces) [29], and the MegaFace [30] datasets.

1) IJB-A: Released by IARPA, this dataset is specifically designed to push the challenges of face recognition, including face imagery coming both as still images and video frames, captured under severe variations of imaging conditions, focusing on the extreme cases. The dataset comprises a total of 25, 800 images of 500 individuals. Two main protocols are defined: face identification (1:N) and face verification (1:1). In both, the identities to be matched or retrieved are expressed by means of *templates*, i.e., collections of images (or video frames) of the same individual. In the identification protocol, identities in the *probe* set have to be retrieved among the ones in the *gallery* set. In the gallery set, each identity is associated to a single template, while in the probe set each identity can have more than one template.

2) YouTube Faces: The YTF dataset collects videos from YouTube and it is specifically designed to study the problem of video based face verification. The dataset contains 3, 425 videos of 1, 595 subjects, and the task is to decide whether two video sequences contain the same subject or not.

3) UMDFaces: The UMDFaces dataset is composed of still images and contains 367,888 face annotations for 8,277 subjects. These are divided in three batches, named *Batch-1*, *Batch-2* and *Batch-3*, and a 1:1 verification protocol, *i.e.*, single image versus single image is defined on Batch-3. This protocol is, in turn, divided in three sub-protocols: *easy*, *moderate* and *difficult*. The latter two are defined based on the absolute difference between the pose of the subjects; the easy protocol considers less than 5 degrees yaw difference between images in each pair, the moderate considers a yaw range that goes from 5 to 20 degrees, while the difficult one, differences of over 20 degrees. The dataset has been subsequently extended to include video frames. In particular, it contains over 3.7 million annotated video frames from over 22,000 videos of 3,100 subjects. Exploiting these, we devised a video against image verification protocol, that will be described in Sect. VI-A.

4) MegaFace: MegaFace is a large scale dataset with a test protocol designed explicitly for evaluating the robustness of a face recognition approach when distractors are included in the gallery. In particular, tests are conducted on the FaceScrub [31] dataset, or on the FG-NET [32] dataset, and the gallery of these two datasets are augmented with an increasing number of distractors, from 10 to 1M. These distractors are selected among 1.02M images of $690,572$ individuals, collected from Flickr. Subjects are disjoint from the ones of the two test sets.

The investigations presented in Sect. IV, V and VI will be conducted mainly on the IJB-A dataset. This choice is motivated by the fact that, among all the current available datasets, it embraces a large spectrum of variabilities; indeed, for each subject, it contains both still images and video frames, along with large resolutions, pose, and expression differences. We argue that such characteristics make the conclusions gathered on this dataset reasonably general. In Sect. VII, a comparative evaluation on the other three datasets is also reported.

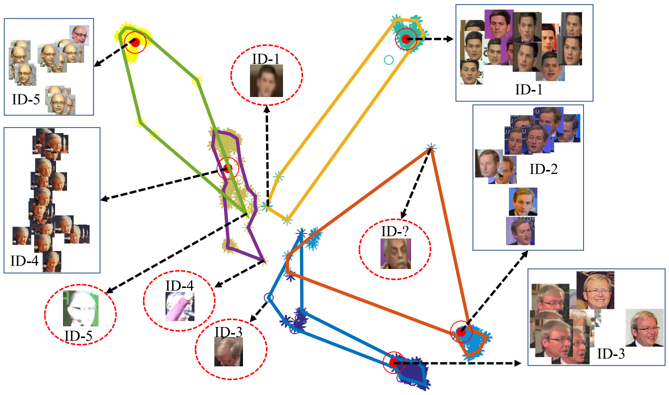


Fig. 2. T-sne plot [34] of the descriptors for 5 identities from the IJB-A dataset, with associated images of critical cases, *e.g.*, extreme illumination, very low resolution, blurring, wrong labels, extreme poses.

IV. Distance Measures Analysis

In most of the recent datasets used for face recognition, the trend is that of representing each individual by a set of images (i.e., a *template*), rather than by a single image. While for nearest-neighbor matching between two images a distance definition is sufficient, when two templates are compared more elaborated matching strategies can be used accounting for the templates data distribution. In the following, we propose and evaluate some template-to-template distance providing evidence of their behavior in combination with different network architectures.

A template (or video sequence) $\mathcal{T}$ is a collection of images (or frames) of the same individual. Indicating with $x_{i,k}$ the k-th face image of the i-th subject, the related template can be written as $\mathcal{T}_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,N_i}\}$, where each template can include a variable number of images $N_i = |\mathcal{T}_i|$. However, the final distance d between two templates must be a scalar value and must be derived from the set of images:

$$d_{i,j} = \phi(\mathcal{T}_i, \mathcal{T}_j), \tag{1}$$

where ϕ is the distance function. Thus, different matching strategies can be used depending on the scenario. To better understand the implications of this choice, we might want to consider the following fact: the softmax-loss used to train a network and classify identities tries to maximize the conditional probability of all the examples in the training mini-batches. In doing so, it tends to fit well to high quality faces, while difficult ones are ignored so that their uncertainty weighs as little as possible in the final cost. As a result, descriptors associated to hard examples eventually share a very low $L2$-norm, while good examples for which the classifier is confident, have high $L2$-norm [33]. For this reason, in the descriptors space, hard examples tend to be randomly displaced, usually in a common "uncertainty" area far away from the centroid of the belonging distribution, *i.e.*, identity, as shown in Fig. 2. Hence, in case of randomly displaced descriptors, *i.e.*, hard examples, correctly or wrongly matching two templates considering a nearest neighbor classifier ends up to be a matter of chance. On the contrary, if the network learns an effective representation, descriptors of the same class should be clustered close and, on average, farther from descriptors of different classes. Inspired by this, we considered diverse distance measures.

Let $\mathbf{f}$ be the vectorial representation of the image x. We consider the *cosine* distance as our function ϕ:

$$\phi(\mathbf{f}_i, \mathbf{f}_j) = 1 - \frac{\mathbf{f}_i \cdot \mathbf{f}_j}{\|\mathbf{f}_i\|_2 \, \|\mathbf{f}_j\|_2}. \tag{2}$$

Exploiting this notation, we devise the following distance measures between templates:

- *Min* – the minimum of the distances between templates:

$$d_{i,j} = \min_{k \in \mathcal{T}_i, h \in \mathcal{T}_j} \phi(\mathbf{f}_{i,k}, \mathbf{f}_{j,h}), \tag{3}$$

- *Mean* – the average distance between templates:

$$d_{i,j} = \frac{1}{N_i N_j} \sum_{k=1}^{|\mathcal{T}_i|} \sum_{h=1}^{|\mathcal{T}_j|} \phi(\mathbf{f}_{i,k}, \mathbf{f}_{j,h}), \tag{4}$$

- *Min-Mean* – the sum of Eq. (3) and (4);

- *Avg-Descr* – the distance is computed between the average template descriptors $\bar{\mathbf{f}}_\mathbf{i}$ and $\bar{\mathbf{f}}_\mathbf{j}$:

$$d_{i,j} = \phi(\bar{\mathbf{f}}_i, \bar{\mathbf{f}}_j). \tag{5}$$

The latter has a particular meaning: it gives clues about the goodness of the learned representation. Referring to Fig. 2, we can see that the centroids (red circled dots) of the different identities are well separated, while some outliers make the regions (polygons) intersect. Consider the situation in which some images, for whatever reason, are projected near the centroid of another class; we can take as example the "pale-blue" points (ID-1) located near the centroid of the "yellow-ocher" class (ID-4) in Fig. 2. If those points are enough close to each other, the minimum distance will classify them correctly. On the other hand, this means that the related descriptors have been generated as if they belonged to a different class (the yellow-ocher class); in other words, the network did not correctly model the distribution of the data and the images have been misclassified. In this sense,

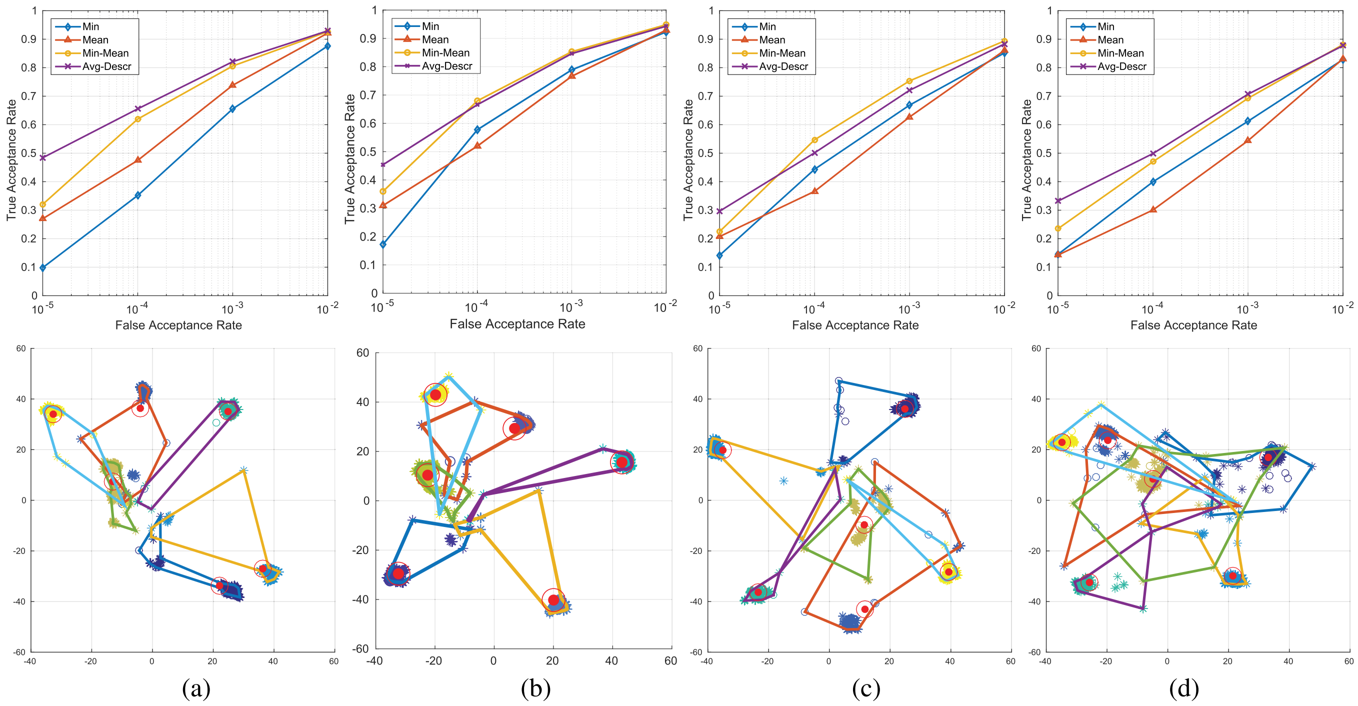


Fig. 3. (Top row) TAR at different FAR rates on the IJB-A dataset for the different distance measures. (Bottom row) T-sne plot of the descriptors of 5 identities from the IJB-A dataset. Different colors indicate the identity, while the red circled dots represent the centroid of each set of descriptors. Intuitively, the less the polygons intersect, the better is the representation. (a) VggFace. (b) Vgg-vd19. (c) AlexNet. (d) ResNet-101-Syn.

the accuracy gap between the different distances can help in understanding this behavior. Intuitively, good results obtained using the *Avg-Descr* imply that the largest part of descriptors of the same identity reside in a small bounded region, and the number of outliers is reduced. Thus, if we select "random" subsets, the probability of having centroids that are closer to the belonging distribution than to other classes is higher. In this scenario, recalling that a template is likely to include only a subset of the subject's images, good recognition accuracy implies that the majority of the descriptors have been correctly displaced, *i.e.*, the images have been correctly classified. Considering the example discussed just now, the *Min* distance can still give good results; however, we argue that it could be not totally faithful in reflecting the reliability of the network in as much as outliers have a strong impact on such measure. Similar observations apply for the other distances considered.

Results: There is a rather strong relationship between the distance measures described above and the effectiveness of the face representation learned by the networks. To stress this relationship, in Fig. 3, for each network, we report results obtained on IJB-A using the various distance measures (top row) along with the T-sne plot of some descriptors from the same dataset (bottom row). The main outcome of the experiment is that the distances give precious hints on the networks behavior. Figure 3 shows that employing the *Avg-Descr* or the *Mean* distance improves the accuracy only if the network has learned a good representation; for instance, we can note that the VggFace and Vgg-vd19 descriptors (Fig. 3(a)-(b) bottom) are well separated in the descriptors space, with a little amount of outliers. This is reflected in the quantitative results (Fig. 3(a)-(b) top), where we witness a large increase of accuracy using the *Avg-Descr* or *Min-Mean* distance. On the contrary, AlexNet (Fig. 3(c)) or ResNet-101-Synth (Fig. 3(d)) show a bit more confused classes distribution with more outliers; consistently, the accuracy gap between different distances is evidently less pronounced.

Another aspect that is worth to notice is that results obtained using the *Min* distance are very similar across all the networks. On the other hand, the other distances have very diverse trends and lead to much different curves. A more detailed inspection reveals that this may be due to the outliers that eventually affect all the networks and make the *Min* distance produce debatable results. This demonstrates the unreliability of this measure and that caution should be put in the evaluation. Furthermore, choosing the right distance can boost the results by a large margin; VggFace at very low false acceptance rates (TAR@10^{-5}FAR) gets around 40% of increased accuracy when using the *Avg-Descr*, which is remarkable.

However, we may observe that, in the optimal case, results of different measures should converge to a similar value. Gaps between such values can help in choosing the most suitable matching strategy or even set the training in the right path. For instance, bad results for the *Min* distance, as seen for VggFace, are a symptom of outliers; bad results for the *Mean* might indicate that different classes intersect with rather large margin and, on average, image pairs get mismatched, which is a sign that possibly the network did not learn to discriminate between subjects effectively.

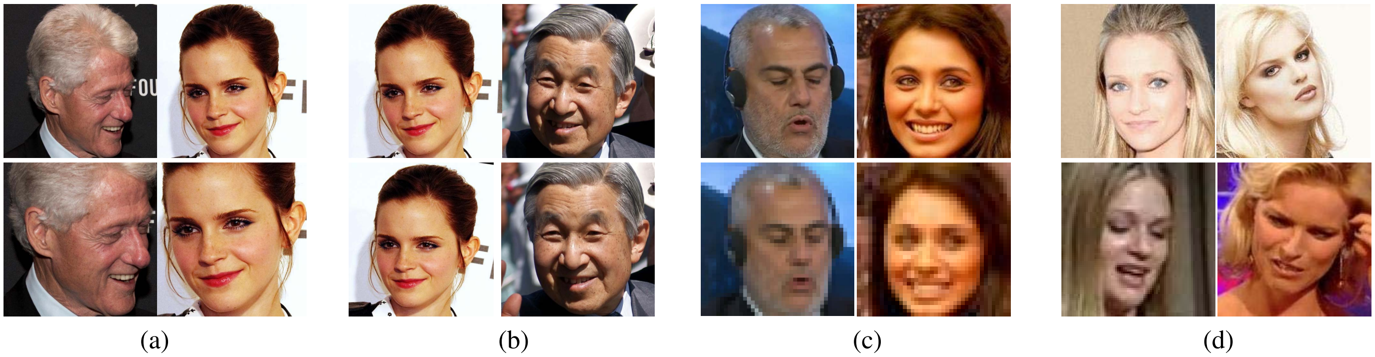


Fig. 4. (a) Examples of large *(top)* and tight *(bottom)* bounding boxes; (b) non-aligned *(top)* and aligned *(bottom)* face images; (c) images at the original resolution *(top)* and rescaled to 40 × 40 *(bottom)*; (d) still images *(top)* and video frames *(bottom)* of the same subject.

Overall, we found the behavior of the *Min-Mean* as the most stable across the different configurations, and used it in all the subsequent evaluations on the IJB-A dataset.

V. Data Preprocessing Analysis

In the following, we focus on the effect the data input to the network and their preprocessing have on DCNNs performance. Preprocessing operations constitute a stage that aims to prepare the data for the network in a proper format. These operations include the detection and clipping of the interested area, the compensation of nuisances such as in-plane or out-of-plane rotations, misalignments and scale differences. In addition to this, data by themselves come with intrinsic characteristics, most of which cannot be changed by preprocessing. First, the visual content to be learned and represented, *i.e.*, the face, can undergo many variations, such as changes in illumination, pose, expression, aging, facial hair and others. This is a crucial aspect that indeed has been extensively studied in controlled conditions [15]. Second, the data characteristics that determine the quality of the images themselves. Among them, we can think of the resolution, artifacts generated by manipulating the image, blurring and others.

Referring to the preprocessing operations and data characteristics, we considered the effect of bounding-box dimension and alignment (Sect. V-A), and of resolution (Sect. V-B).

A. Bounding Box Dimension and Alignment

The dimension of the bounding box that contains the face is relevant inasmuch as it works as a tradeoff between the amount of useful information (the face), and non-useful information (background) that will be fed to the network. Tighter bounding boxes try to minimize the amount of background included but, on the other hand, will eventually reduce the amount of facial information and vice versa. Since the choice of the dimension of the face crops is actually arbitrary, it can be beneficial to understand how their differences impact on the representation obtained through the CNN, so as to choose the best configuration. Our main goal here is to evaluate to what extent the presence or absence of background information and boundary face areas (hair, ears, etc.) influence the performance. To this aim, two different bounding box sizes have been considered to train the networks:

- *Tight*: this bounding box is a square that goes from the chin to just above the forehead. This is the dimension that qualitatively maximizes the ratio between face and background. However, it is hard to define a fixed rule to obtain such a bounding box; there is indeed a number of different face or landmark detectors that might have different outputs. In this circumstance, one may implement its own strategy to obtain a similar bounding box. Examples are shown in Fig. 4(a) bottom row;
- *Large*: this bounding box is taken so as to include the whole head and all the boundary areas, regardless the amount of background, which may vary depending on the head position, *i.e.*, pose. These have been coarsely obtained enlarging the tight boxes by 15% on each side (see Fig. 4(a) top row).

The alignment process, instead, consists in bringing all the faces to the same relative position inside the crops so as to enhance the description semantics. It is intended to compensate for in-plane rotations, translations and scaling. Although the usefulness of the alignment is well founded for engineered computer vision methods based on hand-crafted local features, it has not been fully investigated if the effort is worth with DCNN representations. Furthermore, to the best of our knowledge, such large bounding boxes with aligned face images have never been explored in the literature. To this end, we applied a standard similarity transformation to the images; it is performed using the eyes position, identified by either manual annotation (if available) or exploiting a landmark detector [35]. Following a standard procedure, the image is warped so that the line connecting the eyes is horizontal and the distance between them is 100px. The relative position of the eyes inside the image is kept fixed by setting their average horizontal location in the middle of the image. Examples of aligned and non aligned faces are shown in Fig. 4(b).

Results: Four configurations of AlexNet have been trained for this experiment:

- Aligned-Large (A-L) or Aligned-Tight (A-T): large or tight bounding boxes aligned;
- Original-Large (O-L) or Original-Tight (O-T): large or tight bounding boxes cropped on the original images.

Figure 5 shows that the best results in terms of face identification and verification have been obtained using large bounding

TABLE II

RESULTS ON IJB-A AND MEGAFACE USING DIFFERENT PREPROCESSING METHODS FOR BOTH TRAIN AND TEST IMAGES

IJB-A

		Identification 1:N				Verification 1:1	
Net	Data Type	TAR@0.01FAR	TAR@0.001FAR	Rank@1	Rank@10	TAR@0.01FAR	TAR@0.001FAR
ResNet-101	Original Large	**0.865 ± 0.016**	0.647 ± 0.037	**0.851 ± 0.013**	**0.970 ± 0.004**	**0.830 ± 0.012**	**0.666 ± 0.018**
ResNet-101	Aligned Large	0.850 ± 0.019	**0.651 ± 0.039**	0.831 ± 0.013	0.967 ± 0.006	0.806 ± 0.013	0.627 ± 0.019
AlexNet	Original Large	**0.884 ± 0.008**	**0.720 ± 0.029**	**0.872 ± 0.010**	**0.971 ± 0.005**	**0.862 ± 0.020**	0.731 ± 0.025
AlexNet	Aligned Large	0.874 ± 0.010	0.718 ± 0.031	0.855 ± 0.015	0.964 ± 0.004	0.850 ± 0.018	**0.731 ± 0.028**

MegaFace - Rank@1

Net	Data Type	10 Distractors	10^2 Distractors	10^3 Distractors	10^4 Distractors	10^5 Distractors	10^6 Distractors
ResNet-101	Original Large	**0.949**	0.778	**0.692**	**0.549**	**0.389**	**0.263**
ResNet-101	Aligned Large	0.926	**0.815**	0.660	0.512	0.371	0.249
AlexNet	Original Large	**0.871**	**0.737**	**0.590**	**0.452**	**0.315**	0.218
AlexNet	Aligned Large	0.831	0.705	0.555	0.433	0.307	0.214

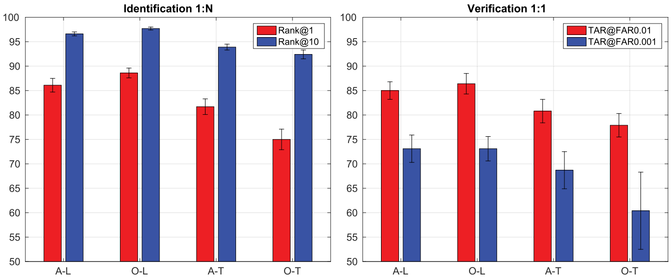


Fig. 5. Results on the IJB-A dataset using the AlexNet architecture with different train and test data preprocessing methods. Left: Rank@1/Rank@10 for the identification protocol; Right: TAR@FAR for the verification protocol.

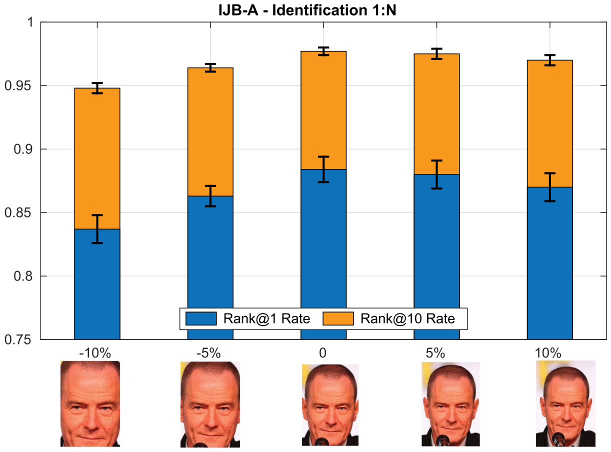
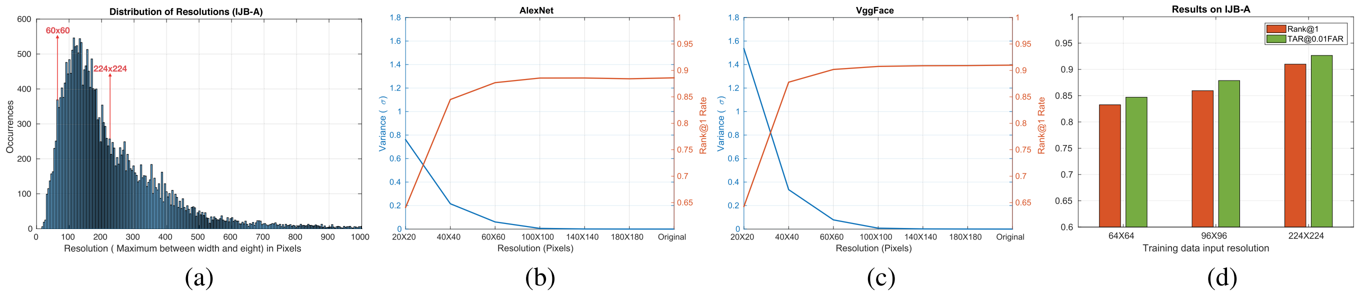


Fig. 6. Recognition performance as a function of the bounding box dimension. The central bar indicates the *large* bounding box. Results obtained for images with enlarged and reduced bounding box size of a certain percentage are given, respectively, on the right and left bars. Note that, for visualization purposes, the images reported below the horizontal axis are not square.

boxes without alignment (O-L case). We also noted that with tighter bounding boxes, a rather large accuracy improvement occurs if alignment is applied. Possibly, the inclusion of a certain amount of context due to larger crops lets the networks autonomously and more effectively learn invariance to spatial transformations. However, this cannot happen if such degrees of freedom are manually removed by the alignment procedure, or in case the bounding box is so tight that only a minimum amount of background is visible. Secondly, observing the experimental evidence, we can suppose that visual information other than the subject's face included by larger boxes can indeed help in separating between visual structures belonging to the face or to the background, which ultimately results in a more effective representation. In any case, this is an important result in as much as the alignment procedure is not trivial; it is an expensive task that implies the detection of some fiducial points on the face, which is by itself a challenging problem that can fail in many circumstances, *e.g.*, profile faces. This most likely implies the introduction of a certain amount of errors that depend on the landmark detector accuracy, but cannot be excluded. Anyhow, in case of landmark detection failures or wrong annotations, we used the original images in their place. Moreover, we argue that there is somewhat an inner incongruity in the definition of alignment; while the procedure is well-defined for near frontal faces, there is a lack of consistency when dealing with profiles or semi-profiles. As an example, assume to have a (semi)profile face and an optimal detector that perfectly locates landmarks even if they are self-occluded; the eyes distance on the image plane would result very small in this case (because of the 3D head rotation) and the alignment procedure, as it is defined, would stretch the image to fit the reference eyes location. The same applies even if we consider the triangle defined by eyes and mouth (or nose) and use a similarity (or affine) transformation to match the two triangles. The aligned images would certainly result somehow distorted. How should the alignment be performed in such cases? If we want to avoid this behavior, another strategy has to be defined, making the alignment process ill-posed. This has two main disadvantages: (1) such design choices are challenging and likely to be inconsistent, making it hard to gather a general and valid procedure, and (2) we have seen that the learning capabilities of the network are impaired and limited if variabilities are manually removed *i.e.* spatial transformations, which instead can be modeled by considering more contextual information.

An analysis on a wider range of bounding box dimensions has also been conducted. The DCNN used in this experiment is the AlexNet architecture trained on large non-aligned images. Figure 6 reports results obtained enlarging and reducing the test image bounding box of a certain percentage starting from a base dimension, that is the *large* bounding box. We observe that, being equal the percentage, the accuracy



(a) (b) (c) (d)

Fig. 7. In (a), the distribution of the resolutions of the IJB-A dataset is reported; the two red arrows indicate the points where the performance start to drop (60 × 60), and the networks input resolution (224 × 224), respectively. The peak is between 112 and 117 pixels. In (b)-(c), the recognition rate is reported in function of the resolution (red line), and the variance between the descriptors extracted at a specific resolution with respect to the original (blue line). In (d), the Rank@1 and TAR@0.01FAR of VggFace (224 × 224) are compared with Vgg-14 and Vgg-11 with input size, respectively, 96 × 96 and 64 × 64.

drop is relative when enlarging the box, while being more significant when reducing its dimension. This suggests us that the DCNN indeed takes advantage from all the available useful information, but suffers when it is missing.

In order to further verify that alignment-free images, with large bounding boxes, lead to an improved face representation, we carried out the same experiment on the large-scale MegaFace dataset considering both AlexNet and ResNet-101. We fine-tuned two versions of ResNet-101 using both the A-L and O-L configurations. The fine-tuning has been performed as described in Sect. III-A. Table II shows the same behavior for both the architectures and datasets, strengthening our thesis.

Our results might seem in contradiction with the outcomes reported by Bansal *et al.* [17]; after deepening this fact, we observed that our *tight* bounding box is with good approximation the *loose* box presented by Bansal et al., which is much smaller than our *large*. In this circumstance results become evidently consistent as they state that alignment leads to better performance, which is the same conclusion we gathered for the *tight* bounding boxes. Still, we prove that going further and enlarging the bounding box improves the performance, and in this case, alignment does not help.

B. Data Resolution

Given a bounding box of a face and its identity label, CNNs effectively extrapolate the semantic information related to the identity directly from the image pixels through convolutional operators. Considering this, the semantic information must then reside in some pattern of the image pixels, or representations derived from them. Still, the actual structures that allow the network to discriminate between different individuals are rather unknown. We investigate this issue in terms of resolution of the image. As shown in Fig. 4(c), even a drastic loss of details does not imply the structural information to be compromised, and the subject is still human-recognizable. The goal is to assess to what extent fine-grained details are relevant to infer the identity from a face image. In doing so, we resize all the images to a common resolution before providing them as input to the networks.

Results: As a preliminary analysis, in Fig. 7(a) we report the distribution of the bounding box sizes of the IJB-A dataset. It turned out that the most of the face images have a resolution between 112 and 117 pixels, only the 6% are smaller than 60 × 60, while the 39% are larger than 224 × 224. Since upscaling to resolutions greater than the network input size does not make any sense, all the images have been resized from a minimum of 20 × 20 up to 180 × 180. Note that, after this subsampling, the image must be rescaled so as to match networks' input size (*e.g.*, 224 × 224); this process obviously interpolate the information in the face image. Figures 7(b) and (c) show that up to 60×60 the Rank@1 accuracy remains pretty stable; a tolerable loss happens at 40×40, threshold over which the accuracy dramatically drops. This is consistent for both AlexNet and VggFace, which is interesting in as much as the two architectures apply different convolutional operations; VggFace uses small convolutions (3 × 3) at all layers and thus is arguably robust to details degradation. AlexNet, instead, performs sequentially 11 × 11, 5×5 and 3×3 convolutions; considering a 60×60 face image, the first set of filters actually covers a large part of the image. Again, note that upscaling the image to the network input size only replicates and interpolates information. This evidence discloses that possibly the discriminative information does not lie in the details, but rather in "higher-level" structures. Consistently, Fig. 7(b) and (c) also show that the difference in terms of variance between the face descriptors of images at the original resolution and their rescaled versions is rather small up to 60 × 60; this suggests us that the semantic information (identity in this case) is influenced by fine-grained details only to a small extent. It is also worth noticing that the face does not take up all the space inside the bounding box; for a generic image (in case of large square crops), the face should occupy no more than the 70/80%.

In light of these observations, we argued that it could have been possible to partially restructure the networks so as to adapt to the new image input size, retrain with smaller sized images and obtain roughly the same performance. With this aim, we reorganized and retrained the Vgg-vd16 architecture. We chose this network mainly because the convolutional layers have a fixed kernel size of 3 × 3 and a stride of 1 pixel so that the image size does not change until max pooling operators. This allowed us to change the network architecture slightly so that a comparison would have been reasonably meaningful. Further, the latter kernel size could have been appropriate for small images. We derived two configurations:

TABLE III

RANK@1 ON THE IJB-A USING ALL THE POSSIBLE DATA SOURCES COMBINATIONS, FOR DIFFERENT FINE-TUNED VERSIONS OF VGGFACE

Gallery vs. Probe	VggFace	Vgg-FC7	Vgg-C5	Vgg-C3	Vgg-All
A: Img. vs. Img.	0.886 ± 0.017	0.807 ± 0.016	0.879 ± 0.018	$\mathbf{0.897 \pm 0.012}$	0.895 ± 0.013
B: Img. vs. Frames	0.842 ± 0.030	0.699 ± 0.038	0.860 ± 0.026	$\mathbf{0.859 \pm 0.019}$	0.859 ± 0.023
C: Frames vs. Frames	0.846 ± 0.045	0.718 ± 0.045	0.854 ± 0.045	0.862 ± 0.041	$\mathbf{0.864 \pm 0.038}$
D: Frames vs. Img.	0.761 ± 0.041	0.679 ± 0.041	0.747 ± 0.047	0.772 ± 0.039	$\mathbf{0.786 \pm 0.035}$
E: Mixed	0.910 ± 0.014	0.814 ± 0.011	0.900 ± 0.014	0.915 ± 0.010	$\mathbf{0.917 \pm 0.008}$

Vgg-11 and *Vgg-14*. The former takes a 64 × 64 input image and was obtained by removing the first two convolutional blocks; the latter takes a 96 × 96 input image and only the first convolutional block was removed. The number of output filters has been halved since, conversely, the training could not converge, most likely for the unbalanced number of parameters. All the other training parameters are the same as VggFace. Both architectures have been trained from scratch.

With respect to VggFace, both Vgg-11 and Vgg-14 get worse results. Figure 7(d) shows that we get a Rank@1 accuracy loss of 7.7% and 5%, respectively, for Vgg-11 and Vgg-14. The same applies for TAR at different FAR rates, where an average loss of around 6% occurs. In spite of this, the model size is consistently reduced, the number of parameters is approximately halved and the training procedure took one day instead of one month on a single GeForce GTX 980 GPU. Again, we wish to stress that the amount of visual information is significantly reduced with respect to the original size, while the performance drop is relative in comparison. These findings suggest us that a higher resolution is relevant for a more effective training, but does not boost the performance at test time. The lower number of parameters could also be a concurrent cause of the worse accuracy. Anyhow, this opens the possibility of developing and exploring new and lighter architectures, grounding on the assumption that fine-grained details seem to be of marginal importance to infer the identity.

VI. DATA SOURCE ANALYSIS

Video sequences, as opposed to still images, undergo nuisances such as motion blur or compression artifacts. These differences are even more pronounced when still images are taken professionally, as it happens in most of the current datasets that largely contain celebrity face images. Furthermore, people captured in video sequences generally tend to show a wider range of spontaneous poses and expressions. An example of such differences may be appreciated in Fig. 4(d). To assess how much these differences impact on the recognition performance, it is important to vary the composition of both gallery and probe sets. In the classic identification scenario, these two sets contain both still images and frames. However, this may not always be the case. Indeed, in real scenarios usually few frames of a video or some low resolution images may be available for a subject. For these reasons, we devise a new evaluation protocol in which gallery and probe sets contain exclusively still images or frames. This protocol is made up of 4 setups (image-image, image-video, video-image, video-video), with all the combinations of gallery and probe. The general "mixed" setup, where both frames and images can be present in the gallery and probe sets is also considered.

TABLE IV

RANK@1 ON THE IJB-A USING ALL THE POSSIBLE DATA SOURCES COMBINATIONS, FOR DIFFERENT VERSIONS OF ALEXNET TRAINED FROM SCRATCH

Gallery vs. Probe	AlexNet	AlexNet-Mixed	AlexNet-Frames
A: Img. vs. Img.	$\mathbf{0.863 \pm 0.016}$	0.845 ± 0.021	0.681 ± 0.037
B: Img. vs. Frames	$\mathbf{0.817 \pm 0.022}$	0.763 ± 0.033	0.539 ± 0.042
C: Frames vs. Frames	$\mathbf{0.824 \pm 0.048}$	0.790 ± 0.039	0.654 ± 0.042
D: Frames vs. Img.	$\mathbf{0.665 \pm 0.039}$	0.618 ± 0.038	0.410 ± 0.038
E: Mixed	$\mathbf{0.872 \pm 0.010}$	0.860 ± 0.013	0.703 ± 0.020

Results: To build the above protocol, we used the IJB-A dataset, which contains both still images and frames, and selected the subset of the identities that have at least one still image and one frame. Since, in the original protocol, identities in the probe set can be missing in the gallery set, this selection was made only for the gallery so as to maintain the same set across all the setups. It resulted that 95 out of the total 112 gallery identities were retained. For the probe set, images were filtered out depending on whether still images or frames were used. Results are reported in Table III and IV, leftmost columns. This protocol has been tested with AlexNet and VggFace. For both, a performance drop is observed when gallery and probe data come from different sources, with a greater loss when the gallery is composed of video frames. These outcomes suggest us that video frames are actually more challenging to handle with respect to still images, consistently with the assumptions made at the beginning of the section, and that data homogeneity matters. The higher accuracy of the "mixed" case might instead be attributed to two facts: (1) larger number of images per subject or (2) if we have mixed media within each template, then matching homogeneous data is possible and can improve the results.

In order to deepen these aspects and assess if we can overcome this limit, VggFace and AlexNet have been, respectively, fine-tuned and re-trained with video frames as well. The still images have been taken from the Oxford dataset [10], while the video frames from the UMD dataset. These two datasets share approximately 800 identities; for those identities we then have both still images and video frames. Some of the remaining identities overlap with the ones of the YTF dataset. In order to fairly test these networks also on the YTF, we selected the 2, 942 that do not intersect. Finally, following the guidelines regarding the bounding boxes, the dataset annotations have been enlarged so as to approximately match the optimal size derived in Sect. V-A. No alignment has been applied either. Details on the training are given in the following:

- **AlexNet**: two configurations of this architecture were trained from scratch; for both, we used the same number of training images as used for the original AlexNet, *i.e.*, 2 millions, in order to be comparable. In the first, the training data contains an equal number of still images and video frames, *i.e.*, 1M-1M; we will refer to this version as "AlexNet-Mixed". In the second, the training set is composed of 2M video frames; we will refer to this as "AlexNet-Frames";
- **VggFace**: this architecture was fine-tuned considering a total of $\approx 175,000$ between images and video frames. A new fully connected layer is stacked upon and trained with the softmax supervision to classify the new identities. For the underlying layers, 4 different strategies have been devised varying the layers that were fine-tuned: (1) *"Vgg-FC7"*: only the last FC layer; (2) *"Vgg-C5"*: the last FC layer and the last Conv block; (3) *"Vgg-C3"*: the last FC layer and the last 3 Conv blocks; (4) *"Vgg-All"*: all the layers have been fine-tuned. All the above solutions have been trained for 10 epochs with a starting learning rate of 0.001, halved every 2 epochs.

The experiments on the new protocols have been repeated and results are reported in Table III and IV. The inclusion of frames resulted in different behaviors depending on the architecture and whether the training was performed from scratch or with a pre-trained network.

Training from scratch the AlexNet architecture turned out to worsen the final performance. As also reported by [17], the intuition is that, being equal the number of samples, still images carry more appearance variabilities than video frames, mainly because of the smaller changes that occur within a video sequence; thus, in proportion, many more video frames are needed to reach the same performance as of still images. We confirm this assumption showing that, being equal the number of training images, *i.e.*, 2M, a network trained on sole still images gets the best accuracy. However, note that the accuracy drop is rather small for the "AlexNet-Mixed". While Bansal *et al.* [17] show quite the opposite result, *i.e.*, mixed training data can improve the performance, we argue that such behavior is due to the unbalanced number of training samples; indeed they used 140K still images against 1.4M between still images and frames (a difference of more than 1 million samples). Under this circumstance, their result actually reinforces and completes our claims, as they show that a significantly smaller amount of still images with respect to video frames still enhances the accuracy.

Nevertheless, as stated previously and shown in Fig. 4(d), subjects in video sequences tend to show more versatile and spontaneous expressions than those portrayed in professionally captured pictures; on the one hand, this might potentially help in learning a more robust representation but, on the other, a larger number of parameters or more sophisticated training strategies might be required to model such complex variabilities. Other than that, we observe that the appearance changes within a video sequence often occur smoothly and slowly; if we consider a short period of time, the diversity in visual content is likely to be negligible. This explains the need for a larger number of frames with respect to still images.

TABLE V

BASELINE RESULTS FOR THE PROPOSED "VIDEO2TEMPLATE" PROTOCOL

	TAR@0.01FAR	TAR@0.001FAR	TAR@0.0001FAR
AlexNet	0.910 ± 0.005	0.689 ± 0.062	0.131 ± 0.089
VggFace	0.948 ± 0.004	0.704 ± 0.024	0.122 ± 0.044
Vgg-vd19	$\mathbf{0.958 \pm 0.003}$	$\mathbf{0.723 \pm 0.064}$	$\mathbf{0.139 \pm 0.038}$

The latter statement is supported by the evidence that fine-tuning VggFace with video frames led to enhanced accuracy. However, results reveal that only the two solutions "Vgg-C3" and "Vgg-All" improve upon the original VggFace. This is symptom of a diversity in the characteristics of the data and suggests us that even the information carried in the lower-level layers needs to be revisited when attempting to classify the new identities. This is consistent with the findings of Yosinski *et al.* [36] who showed that tuning all the layers in a transfer learning scenario leads to an improvement, while the opposite happens if they are kept blocked. They also showed that such improvement is larger the more the data distribution differs from the original. We then wondered if such improvement could be ascribed to the fine-tuning procedure itself. As further confirmation, we repeated the fine-tuning including exclusively still images in the training set of the "Vgg-All" configuration. This indeed improved upon the original VggFace, but the relative gain resulted inferior.

These outcomes prove that the different training data types do impact on the results, and it can be useful to include video frames in the training set, but we need to be aware that the larger complexity requires either more data, representational power or additional precautions in training.

A. UMDFaces "Video2Template" Verification Protocol

We showed above that still images and video frames have different characteristics and properties. However, the protocol that we defined on the IJB-A dataset does not allow us to have a fair comparison with other solutions. We propose here a verification protocol in which video sequences are matched with templates composed of still images only.

The protocol is defined over the "Frames" and the "Batch-1" sets of the UMDFaces dataset, which share the same $3,106$ identities. The identities are shuffled and split in 3 parts that are in turn used to build the train and test sets following a 3-fold cross validation procedure. At each turn, one fold is used as test set, while the other two as train set. Each of the $22,061$ videos in the "Frames" set is matched with two templates of images, one positive and one negative, for a total of $44,122$ pairs; each fold comprises approximately $14,000$ pairs. Templates include all the images of the related individual; the average number of images per subject is 50.[1]

In Table V, we provide some baseline results for the proposed protocol using the networks trained by us, *i.e.*, AlexNet and Vgg-vd19, and the pre-trained VggFace. The matching was performed using the *Min-Mean* distance. This protocol will allow us to specifically address the problem of matching face images from different data sources and be used

[1] Data is available at https://github.com/clferrari/UMD-video2template.

TABLE VI

RESULTS ON IJB-A USING THE ARCHITECTURES OF SECT. III-A WITH THE BEST CONFIGURATIONS AGAINST THE STATE-OF-THE-ART. BEST RESULTS ARE REPORTED IN BOLD, SECOND BEST ARE UNDERLINED

		Identification 1:N				**Verification 1:1**	
Net	Configuration	TAR@0.01FAR	TAR@0.001FAR	Rank@1	Rank@10	TAR@0.01FAR	TAR@0.001FAR
ResNet-101-Synth	A-L / Min-Mean	0.885 ± 0.013	0.706 ± 0.022	0.873 ± 0.013	0.974 ± 0.005	0.830 ± 0.024	0.661 ± 0.036
VggFace	O-L / Min-Mean	0.926 ± 0.011	0.804 ± 0.022	0.910 ± 0.014	0.983 ± 0.003	0.896 ± 0.016	0.759 ± 0.041
VggFace-All	O-L / Min-Mean	0.937 ± 0.008	0.825 ± 0.019	0.917 ± 0.008	0.984 ± 0.001	0.905 ± 0.016	0.785 ± 0.019
ResNet-101	O-L / Min-Mean	0.865 ± 0.016	0.647 ± 0.037	0.851 ± 0.013	0.970 ± 0.004	0.830 ± 0.012	0.666 ± 0.018
AlexNet	O-L / Min-Mean	0.884 ± 0.008	0.720 ± 0.029	0.872 ± 0.010	0.971 ± 0.005	0.862 ± 0.020	0.731 ± 0.025
Vgg-vd19	O-L / Min-Mean	**0.948 ± 0.007**	**0.853 ± 0.019**	0.936 ± 0.008	0.987 ± 0.002	0.913 ± 0.014	0.789 ± 0.045
PAMs [20]	–	–	–	0.840 ± 0.012	0.946 ± 0.007	0.826 ± 0.018	0.652 ± 0.037
Template Adaptation [37]	–	0.774 ± 0.050	–	0.928 ± 0.010	0.986 ± 0.003	0.939 ± 0.013	0.836 ± 0.027
TPE [38]	–	0.932 ± 0.010	0.753 ± 0.030	0.932 ± 0.010	0.977 ± 0.005	0.900 ± 0.010	0.813 ± 0.020
All-In-One CNN+TPE [39]	–	0.792 ± 0.020	–	0.947 ± 0.008	**0.988 ± 0.003**	0.922 ± 0.010	0.823 ± 0.020
NAN [40]	–	0.817 ± 0.041	–	**0.958 ± 0.005**	0.986 ± 0.003	**0.941 ± 0.008**	**0.881 ± 0.011**

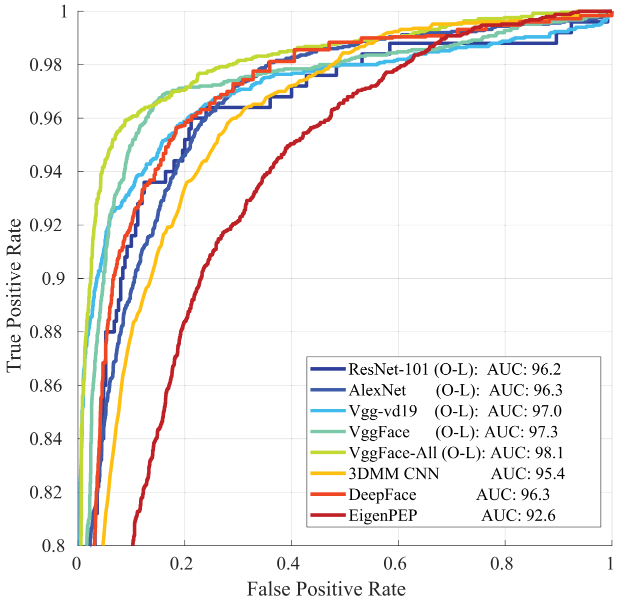


Fig. 8. ROC curves on the YTF for the architectures presented in Sect. III-A and state-of-the-art methods.

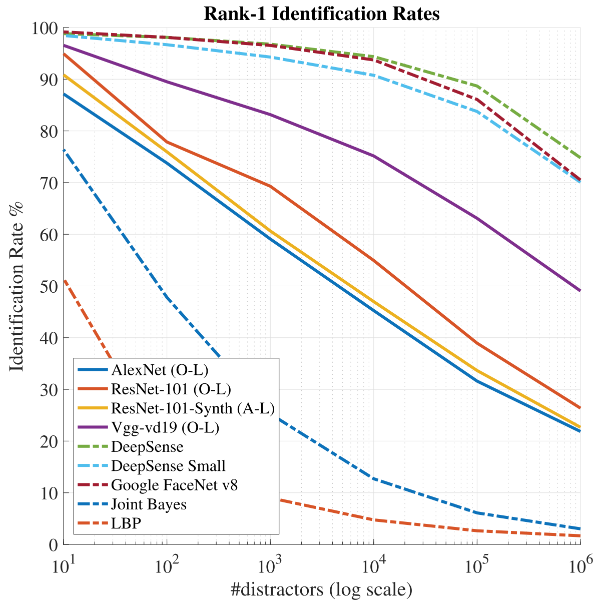


Fig. 9. Rank@1 at different number of distractors for the MegaFace dataset.

as benchmark for the specific task. The particular choice of having sets of images for both the data types in the form of full sequences or templates is motivated by the fact that, as shown in Sect. IV, in such cases choosing the right matching strategy can be crucial.

VII. COMPARATIVE EVALUATION

For the sake of completeness, we evaluated the architectures presented in Sect. III-A using their best configurations on the IJB-A, MegaFace, YTF and UMDFaces datasets. The subsequent results are intended to show that, considering all the outcomes presented so far, one can get competitive results applying a standard recognition pipeline.

A. IJB-A – Template Based Matching

Results obtained with the best configurations of the networks of Sect. III-A on the IJB-A dataset are reported in Table VI. In light of the outcomes reported in Fig. 3, for this dataset the *Min-Mean* distance was used. Results gathered by choosing large non-aligned bounding boxes and the correct distance measure are comparable with respect to state-of-the-art methods [20], [37]–[40]. The slightly lower results obtained by our ResNet-101 with respect to ResNet-101-Synth can be fairly attributed to the fact that we fine-tuned our version with 1M images, while ResNet-101-Synth has been fine-tuned with 2M. However, all the other networks get better results.

B. YouTube Faces – Video Based Verification

For what concerns the YTF, we considered the original frames (without any preprocessing) to extract the DCNN descriptors. As for the bounding boxes, the provided annotations define a crop that resembles the *tight* one shown in Fig. 4. As we found that the best option is to have larger crops, we enlarged the annotations so as to match the O-L configuration. We further observed that the video sequences included in the dataset are rather short and consequently the appearance changes of the subject within the sequence are limited; thus, we chose to employ the *Avg-Descr* measure, which resulted the best performing in most of the cases. Results for this dataset are reported in Fig. 8; the ROC curves show that the best performing network is the VggFace-All, fine-tuned on the UMD video frames, which gets a noticeable boost with respect to the original VggFace, reinforcing the assumption that data types consistency is also important.

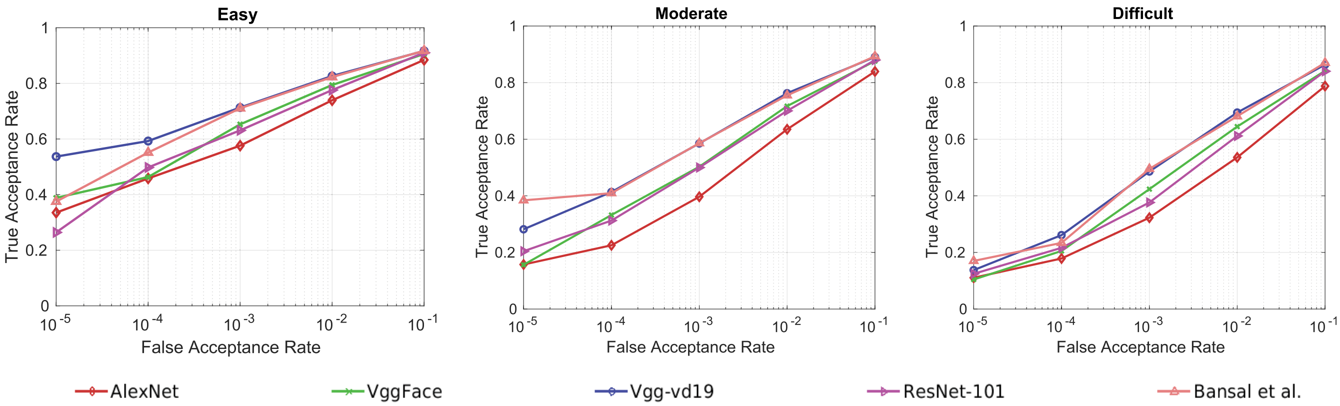


Fig. 10. TAR@ different FAR for the architectures presented in Sect. III-A on the three verification protocols defined on the Batch-3 of the UMD dataset.

Overall, almost all the tested networks obtain higher results with respect to the reported state-of-the-art methods, namely 3DMM CNN [41], DeepFace [8] and EigenPEP [42].

C. MegaFace – Large Scale Identification

Results on this dataset are reported in Fig. 9. The plot shows the Rank@1 accuracy at different number of gallery distractors, from 10 to 1M; compared to the two reference state-of-the-art methods, we see that our results are comparable for a small number of distractors (10, 100), point after which they start to degrade faster. This behavior is reasonable in as much as handling so many different identities requires very robust descriptors that can be obtained either using enormous amount of images for training (Google FaceNet v8 is trained on 500M images of 10M different people) or metric learning techniques and auxiliary loss functions (DeepSense uses three loss functions: identification loss, verification loss and triplet loss). What is important to stress here is that the assumptions made so far still apply even for a very large scale dataset. Nonetheless, contrary to the IJB-A, with respect to ResNet-101-Synth (trained and tested on aligned data), our version trained on large, non-aligned images gets higher accuracy on this dataset.

D. UMDFaces – Single Image Based Verification

The protocol defined over this dataset is face verification performed between single images, thus a simple distance ranking has been used. All the tested networks have been trained with large, non aligned crops (O-L). Results for the three different protocols are reported in Fig. 10. Here, the best performance is obtained with our Vgg-vd19, which is comparable to the state-of-the-art solution of Bansal *et al.* [17]; note that the latter is trained employing an auxiliary Triplet Probabilistic Embedding (TPE) loss function, while we employ a simple Softmax classifier.

VIII. Conclusion

In this paper, we investigated various aspects that can influence face recognition performance based on DCNN representation that are infrequently taken into account. We deepened the relationship between the representation derived with diverse DCNN architectures and the distance measures used to perform matching. We showed how different distance measures can give precious hints on the effectiveness of the learned face representation and that, depending on the circumstances, these measures can drastically change the final results. We then explored the image characteristics, including the dimension of the bounding box containing the face, the image resolution and alignment operations. From the experimental evidence, we can conclude that the final representation benefits from the inclusion of a certain amount of background. This let us also demonstrate that image normalization operations are not crucial and, rather, they seem to limit the expressive power of the networks. In this context, we also showed that even a rather drastic loss of details in the images does not irremediably impair the accuracy; possibly, the actual identity information lies in higher visual structures. We then focused on the data source, *i.e.*, still images or video frames, its peculiarities and the impact this diversity has on the final performance. A new protocol in which full video sequences have to be matched with templates of still images has also been proposed. We revealed the source of the data actually impacts on the results for various reasons; these can involve nuisances typical of video sequences like motion blur, or the stronger spontaneity of the subjects captured. Training a DCNN including video frames resulted in very different behaviors: on the one hand, it resulted more effective than using still images only in case of fine-tuning a very deep architecture, pre-trained on face images; on the other hand, it made the learning procedure more challenging if done from scratch on a shallower architecture. Finally, we showed that taking into account all the above aspects, we can achieve competitive results with respect to state-of-the-art approaches on different datasets equipped with very diverse protocols. As further and possible future work, it would be worth investigating the effect of applying our gathered conclusions to more complex frameworks including different loss functions and their combinations.

References

[1] W. W. Bledsoe, “Some results on multicategory pattern recognition,” *J. ACM*, vol. 13, no. 2, pp. 304–316, 1966.

[2] 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,” Univ. Massachusetts, Boston, MA, USA, Tech. Rep. 07-49, Oct. 2007.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[5] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 487–495.

[6] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," in *Proc. Int. Workshop Hum. Behav. Understand. (HBU)*, Nov. 2011, pp. 29–39.

[7] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[8] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1701–1708.

[9] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1891–1898.

[10] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2015, pp. 1–12.

[11] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

[12] X. Yin and X. Liu, "Multi-task convolutional neural network for pose-invariant face recognition," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 964–975, Feb. 2018.

[13] G. Hu, X. Peng, Y. Yang, T. M. Hospedales, and J. Verbeek, "Frankenstein: Learning deep face representations using small data," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 293–303, Jan. 2018.

[14] I. Masi, T. A. Trân, T. Hassner, J. T. Leksut, and G. Medioni, "Do we really need to collect millions of faces for effective face recognition?" in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 579–596.

[15] M. M. Ghazi and H. K. Ekenel, "A comprehensive analysis of deep learning based representation for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 34–41.

[16] C. Ferrari, G. Lisanti, S. Berretti, and A. Del Bimbo, "Investigating nuisance factors in face recognition with DCNN representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 81–89.

[17] A. Bansal, C. D. Castillo, R. Ranjan, and R. Chellappa, "The do's and don'ts for CNN-based face verification," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Sep. 2017, pp. 2545–2554.

[18] K. Simonyan and A. Zisserman. (Sep. 2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[19] 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.

[20] I. Masi, S. Rawls, G. Medioni, and P. Natarajan, "Pose-aware face recognition in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4838–4846.

[21] D. Yi, Z. Lei, S. Liao, and S. Z. Li. (Nov. 2014). "Learning face representation from scratch." [Online]. Available: https://arxiv.org/abs/1411.7923

[22] G. Hu *et al.*, "Attribute-enhanced face recognition with neural tensor fusion networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3764–3773.

[23] D. Wang, C. Otto, and A. K. Jain, "Face search at scale," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1122–1136, Jun. 2017.

[24] X. Wu, R. He, T. Tan, and Z. Sun. (Nov. 2015). "A light CNN for deep face representation with noisy labels." [Online]. Available: https://arxiv.org/abs/1511.02683

[25] S. Banerjee *et al.*, "To frontalize or not to frontalize: Do we really need elaborate pre-processing to improve face recognition?" in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 20–29.

[26] I. Masi *et al.*, "Learning pose-aware models for pose-invariant face recognition in the wild," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[27] B. F. Klare *et al.*, "Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1931–1939.

[28] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 529–534.

[29] A. Bansal, A. Nanduri, C. Castillo, R. Ranjan, and R. Chellappa. (Nov. 2016). "UMDFaces: An annotated face dataset for training deep networks." [Online]. Available: https://arxiv.org/abs/1611.01484

[30] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The MegaFace benchmark: 1 million faces for recognition at scale," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4873–4882.

[31] H.-W. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 343–347.

[32] (2015). *FGNET*. [Online]. Available: http://www-prima.inrialpes.fr/FGnet/html/benchmarks.html

[33] R. Ranjan, C. D. Castillo, and R. Chellappa. (Jun. 2017). "L2-constrained softmax loss for discriminative face verification." [Online]. Available: https://arxiv.org/abs/1703.09507

[34] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[35] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1867–1874.

[36] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 3320–3328.

[37] N. Crosswhite, J. Byrne, C. Stauffer, O. Parkhi, Q. Cao, and A. Zisserman, "Template adaptation for face verification and identification," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May/Jun. 2017, pp. 1–8.

[38] S. Sankaranarayanan, A. Alavi, C. D. Castillo, and R. Chellappa, "Triplet probabilistic embedding for face verification and clustering," in *Proc. IEEE Int. Conf. Biometrics Theory, Appl. Syst. (BTAS)*, Sep. 2016, pp. 1–8.

[39] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa, "An all-in-one convolutional neural network for face analysis," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May/Jun. 2017, pp. 17–24.

[40] J. Yang *et al.*, "Neural aggregation network for video face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4362–4371.

[41] A. T. Tran, T. Hassner, I. Masi, and G. Medioni, "Regressing robust and discriminative 3D morphable models with a very deep neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1493–1502.

[42] H. Li, G. Hua, X. Shen, Z. Lin, and J. Brandt, "Eigen-PEP for video face recognition," in *Proc. Asian Conf. Comput. Vis.*, 2014, pp. 17–33.