DeepVisage: Making face recognition simple yet with powerful generalization skills

Abul Hasnat[1], Julien Bohné[2], Jonathan Milgram[2], Stéphane Gentric[2], and Liming Chen[1]

[1]Laboratoire LIRIS, École centrale de Lyon, 69134 Ecully, France.
[2]Safran Identity & Security, 92130 Issy-les-Moulineaux, France.
md-abul.hasnat@ec-lyon.fr, {julien.bohne, jonathan.milgram,
stephane.gentric}@safrangroup.com, liming.chen@ec-lyon.fr

Abstract

Face recognition (FR) methods report significant performance by adopting the convolutional neural network (CNN) based learning methods. Although CNNs are mostly trained by optimizing the softmax loss, the recent trend shows an improvement of accuracy with different strategies, such as task-specific CNN learning with different loss functions, fine-tuning on target dataset, metric learning and concatenating features from multiple CNNs. Incorporating these tasks obviously requires additional efforts. Moreover, it demotivates the discovery of efficient CNN models for FR which are trained only with identity labels. We focus on this fact and propose an easily trainable and single CNN based FR method. Our CNN model exploits the residual learning framework. Additionally, it uses normalized features to compute the loss. Our extensive experiments show excellent generalization on different datasets. We obtain very competitive and state-of-the-art results on the LFW, IJB-A, YouTube faces and CACD datasets.

1. Introduction

Face recognition (FR) is one of the most demanding computer vision tasks, due to its practical use in numerous applications, such as biometric, surveillance and human-machine interaction. The state-of-the-art FR methods [34, 29, 31, 24, 20] surpassed human performance (97.53%) and achieved significant accuracy on the standard labeled faces in the wild (LFW) [14] benchmark. These remarkable results are achieved by training the deep convolutional neural network (CNN) [10] with large databases [11, 24, 44, 2].

The facial image databases mostly provide the identity labels. These labels allow the CNN models to be easily trained with the softmax loss. FR methods generally use the trained CNN model to extract facial features and then perform verification by computing distance or recognition with a classifier. However, from our extensive study (see Sect. 2), we observe that recent methods include different additional strategies to obtain better performance, such as:

1. *train CNN with different loss functions [29, 31]:* requires carefully preparing the image pairs/triplets by maintaining certain constrains [29], because arbitrary pairs/triplets do not contribute to the training. Online triplet generation requires a larger batch size (e.g., [29] used 1.8K images in a mini-batch with 40 images/identity), which is excessive for a limited resource machine. On the other hand, using offline triplets can be critical as many of them will be useless while training progresses. The joint optimization [31] with Softmax and Contrastive losses not only requires specific training data (with identity and pair labels) but also complicates the training procedure.

2. *fine-tune CNN:* requires training on each target dataset, which restricts the ability to generalize.

3. *metric learning [28, 9]:* requires particular form of training data (e.g., triplets). Moreover, it does not always guarantee to enhance performance [37].

4. *concatenating features from multiple CNNs [31, 20]:* requires additional training data of different forms and train CNNs for each form. Besides, it is necessary to explore the particular modalities that can contribute to enhance performance.

The use of the above strategies requires significant efforts in terms of data preparation or selection and computing resources. On the other hand, recent results on the ImageNet challenge [26] indicate that deeper CNNs enhance

performance of different computer vision tasks. These observations raise the following question - *can we achieve state-of-the-art results with a single CNN model which is trained only once with the identity labels?* Our research is motivated by this question and we aim to address it by developing a simple yet robust single-CNN based FR method. Moreover, we want that our once-trained single CNN based FR method generalizes well across different datasets.

In this research, our primary objective is to discover an efficient CNN architecture. We follow the recent findings, which suggest that deeper CNNs perform better on a number of computer vision tasks [13, 10]. We construct a deep CNN model with 27 convolutional and 1 fully connected (FC) layers, which incorporates the residual learning framework [13]. Moreover, we aim to find an efficient way to train our CNN only with the identity labels. Recently, [39] achieves high FR performance with a CNN trained from the identity labels. However, they perform joint optimization using the softmax and center loss [39] (CL). CL improves the features discrimination among different classes. It follows the principle that, features learned from a deep CNN should minimize the intra-class distances. Interestingly, we observe (see Fig 3) that an equivalent representation can be achieved by normalizing the CNN features before computing the loss. Therefore, we train our CNN using the softmax loss with the normalized features.

With our single CNN model, first we evaluate on the LFW [14] benchmark and observe that it obtains 99.62% accuracy. In order to demonstrate its effectiveness, we evaluated it on different challenging face verification tasks, such as face templates matching on the IJB-A [16] dataset, video faces matching on the YouTube Faces [40] (YTF) dataset and cross age face matching on the CACD [3] dataset. Our method achieves 82.4% TAR@FAR=0.001 on IJB-A [16], 96.24% accuracy on YTF [40] and 99.13% accuracy on CACD [3]. These results indicate that our method achieves very competitive and state-of-the-art results. Moreover, it generalizes very well across different datasets.

We summarize our contributions as follows: (a) conduct extensive study and provide (Sec 2) a review and methodological comparison of the state-of-the-art methods; (b) propose (Sect. 3) an efficient single CNN based FR method; (c) conduct (Sect. 4) extensive experiments on different datasets, which demonstrate that our method has excellent generalization ability; and (d) perform (Sect. 4.3) an in-depth analysis to identify the influences of different aspects.

In the remaining part of this paper, first we study and analyze the state-of-the-art FR methods in Section 2, describe our proposed method in Section 3, present experimental results, perform analysis of our method and discuss them in Section 4 and finally draw conclusions in Section 5.

2. Related work, state-of-the-art FR methods

Face recognition (FR) in unconstrained environment attracts significant interest from the community. Recent methods exploited deep CNN models and achieved remarkable results on the LFW [14] benchmark. Besides, numerous methods have been evaluated on the IJB-A [16] dataset. We study[1] and analyze these methods based on several key aspects: (a) details of the CNN model; (b) loss functions used; (c) incorporation of additional learning strategy; (d) number of CNNs and (e) the training database used.

Recent methods tend to learn CNN based features using a *deep architecture* (e.g., 10 or more layers). This is inspired from the extraordinary success on the ImageNet [26] challenge by famous CNN architectures [10], such as AlexNet, VGGNet, GoogleNet, ResNet, etc. The FR methods commonly use these architectures as their baseline model (directly or slightly modified). For example, AlexNet is used by [27, 28, 21, 25, 1, 22, 29], VGGNet is used by [24, 8, 23, 1, 22, 9, 33] and GoogleNet is used by [42, 29]. CASIA-Webface [44] proposed a simpler CNN model, which is used by [37, 5, 9]. Several methods, such as [32, 34, 35, 39, 31] use a model with lower depth but increase its complexity with locally connected convolutional layers. Besides, [46] use 4 parallel 10 layers CNNs to learn features from different facial regions. *We follow the ResNet [13] based deep CNN model.*

FR methods often train multiple CNNs and accumulate features from all of them to construct the final facial descriptors. It provides an additional boost to the performance. Different types of inputs are used to train these multiple CNNs: (a) [32, 31, 33, 37, 9, 20] used image-crops focused on certain facial regions (eyes, nose, lips, etc.); (b) [9, 1, 22, 34] used different modality of input images, such as 2D, 3D, frontalized and synthesized faces at different poses and (c) [35, 20] used different training databases with varying number of images. *We do not follow this approach and train only one CNN.*

The CNN model parameters are learned by optimizing loss functions, which are defined based on the given task (e.g., classification, regression) and the available information (e.g., class labels, price). The softmax loss [10] is a common choice for classification tasks. It is often used by the FR methods to create good face representation by training the CNN as an identity classifier. It requires only the identity labels. The contrastive loss [10, 7] is used by [32, 34, 33, 31, 42] for face verification and requires face image pairs and similarity labels. The triplet loss [29] is used by [29, 24, 8, 20] for face verification and requires the face triplets. Recently the center loss [39] is proposed to enhance feature discrimination, which uses the identity labels.

[1] We consider only the CNN based methods. For the others, we refer readers to the recently published survey [17] for LFW and [16] for IJB-A.

We use the softmax loss.

Several methods use multiple loss functions and train CNN using joint optimization [32, 33, 31, 39, 25]. The other way is to use them sequentially [34, 24, 8, 20, 42], i.e., first train with the softmax and then train with the other loss. We observe that using multiple loss functions complicates the training data preparation task and the CNN training procedure. *Therefore, we avoid this type of strategies*.

Fine-tuning the CNN parameters is a particular form of transfer learning. It is commonly employed by several methods [37, 5, 27] on the IJB-A [16] dataset. It refines the CNN parameters from a previously learned model using a target specific training dataset. Several methods do not directly use the raw CNN features but employ an additional learning strategy. The *metric/distance learning* strategy based on the Joint Bayesian method [4] is a popular one and used by [32, 44, 37, 5, 33, 31, 9]. Recently, two different strategies [28, 28] have been proposed to learn feature embedding using face triplets. Another strategy, called template adaptation [8], exploits an additional SVM classifier. Apart from these, principal component analysis (PCA) is used by several methods [23, 1, 22] to learn a dataset specific projection matrix. [42] learns an aggregation module to compute scores among two videos. The above methods often need training data from the target datasets. Moreover, they [27, 28] may need to carefully prepare the training data, e.g., triplets. *We do not need any such learning strategies*.

The use of a large facial training dataset is important to achieve high FR accuracy [29, 46]. [46] provided an in-depth analysis and demonstrated the effect of the dataset size and the number of identities for FR. Following the high demand of a large FR dataset, several publicly available datasets have been released recently. Among them, CASIA-WebFace [44] is used by numerous methods [39, 27, 28, 21, 25, 44, 37, 5, 9, 41, 23, 1, 22]. Several researches [23, 1, 22] enlarge it by synthesizing facial images with different shapes and poses based on the 3D face models. Recently, the MSCeleb [11] dataset has been publicly released. It contains the largest collection of facial images and identities. *We exploit it to develop our FR method*.

3. Proposed Method

Our FR method, called *DeepVisage*, consists in preprocessing face image, learning CNN based facial features and computing similarity. Following the recent trend [34, 29, 31, 24, 44], we exploit the CNN as the core component. Our deep CNN model follows the residual learning framework [13]. Moreover, it intelligently exploits feature normalization, which is a crucial step, see Sect. 4.3. Our pre-processing stage consists in the detection of the face and facial landmarks, which are used to create a normalized face image. We compute the cosine similarity among the features of a pair of faces as the verification score. Below, we describe these elements.

3.1. Building blocks and deep CNN architecture

Convolutional networks: We begin with the basic ideas of CNN [18]: (a) local receptive fields with identical weights via the convolution operation and (b) spatial subsampling via the pooling operation. At a particular layer l, the convolution of the input $f_{x,y}^{Op,l-1}$ to obtain the k^{th} output feature map $f_{x,y,k}^{C,l}$, can be expressed as:

$$f_{x,y,k}^{C,l} = \mathbf{w}_k^{l\,T} f_{x,y}^{Op,l-1} + b_k^l \tag{1}$$

where, $\mathbf{w}_k^l$ and b_k^l are the shared weights and bias. C denotes convolution and Op (for $l > 1$) denotes various tasks, such as convolution, sub-sampling or activation. For $l = 1$, Op represents the input image. Sub-sampling or pooling performs a simple local operation, such as computing the average or maximum value in a local spatial neighborhood followed by reducing spatial resolution. We apply max pooling for our CNN, which has the following form:

$$f_{x,y,k}^{P,l} = \max_{(m,n)\in\mathcal{N}_{x,y}} f_{m,n,k}^{Op,l-1} \tag{2}$$

where, $\mathcal{N}_{x,y}$ denotes the local spatial neighborhood of (x, y) coordinate and P denotes the pooling operation.

In order to ensure non-linearity of the network, the feature maps are passed through a non-linear activation function, e.g., the Rectified Linear Unit (ReLU) [10, 12]: $f_{x,y,k}^{l} = max(f_{x,y,k}^{l-1}, 0)$. We apply the Parametric Rectified Linear Unit (PReLU) [12] as the activation function, which has the following form:

$$f_{x,y,k}^{A,l} = max(f_{x,y,k}^{Op,l-1}, 0) + \lambda_k min(f_{x,y,k}^{Op,l-1}, 0) \tag{3}$$

where, λ_k is a trainable parameter to control the slope of the linear function for the negative input values and A denotes activation operation.

At the basic level, a CNN is constructed by stacking series of convolution, activation and pooling layers, see LeNet-5 [18] for an example. Often a layer with full connections is placed at the end of the stacked layers, called the fully connected (FC) layer. It takes all points (neurons) from the previous layer as input and connects it to all points (neurons) of the output layer.

Residual learning framework [13]: A recent trend [10] on the ImageNet [26] challenge shows that deeper CNNs achieve better results. However, it increases the model complexity, which makes it harder to optimize the loss of the CNN model. Besides, they may generate higher training error than a shallower CNN [13]. The residual learning framework [13] provides a solution to these problems.

For a stack of a few layers, residual learning fits a mapping $\mathcal{F}(f) := \mathcal{H}(f) - f$ instead of fitting the underlying

mapping $\mathcal{H}(f)$. Therefore, the original mapping is formulated as $\mathcal{F}(f) + f$, which means directly adding the input feature map f with the output of the stacked layers $\mathcal{F}(f)$. This idea can be easily implemented with the notion of *shortcut connection*. Formally, the output of a residual block R can be expressed as:

$$f^{R,l}_{x,y,k} = f^{Op,l-q}_{x,y} + \mathcal{F}(f^{Op,l-q}_{x,y}, \{W_k\}) \quad (4)$$

where, $f^{Op,l-q}_{x,y}$ represents the input feature map, $\mathcal{F}(.)$ is the residual mapping to be learned, W_k is the parameters of the k^{th} residual block and q is the total number of stacked layers within the residual block. The flexible form of the residual function $\mathcal{F}(.)$ allows to stack multiple layers with different types of operations, such as convolution, pooling, activation etc. All of the residual blocks in our CNN consist of two convolution layers with different numbers of neurons. Each convolution is followed by a PReLU activation.

Loss function: Deep CNNs are trained by optimizing loss function. We use the softmax loss, which is widely used for classification:

$$\mathcal{L}_{Softmax} = -\sum_{i=1}^{N} log \frac{e^{\mathbf{w}^T_{y_i} f_i + b_{y_i}}}{\sum_{j=1}^{K} e^{\mathbf{w}^T_j f_i + b_j}} \quad (5)$$

where, f_i and y_i are the features and true class label of the i^{th} image. $\mathbf{w}_j$ and b_j denote the weights and bias of the j^{th} class. N and K denote the number of training samples and the number of classes.

Feature normalization (FN): It is often used as a necessary step in many learning algorithms. It ensures that all of the features have equal contribution to the cost function [36]. With deep CNNs, we cannot guarantee this by only normalizing the input image pixels, because the scale of features (from the final FC layer) may change due to a series of operations at different layers. Therefore, to avoid the influence of un-normalized features during cost computation, we provide normalized features f^{Nr}_i to the softmax loss as: $f^{Nr} = \frac{f^{Op} - \mu}{\sqrt{\sigma^2}}$, where μ and σ^2 are the mean and variance.

During training, we apply normalization by computing μ and σ from the samples of each mini-batch. Moreover, we maintain the moving average of μ and σ and use them to normalize the test samples. Note that, this is a specific case of the popular batch normalization (BN) technique [15] with scale $\gamma = 1$ and shift $\beta = 0$.

Proposed CNN architecture: Our CNN model consists of 27 convolution (*Conv*), 4 pooling (*Pool*) and 1 fully connected (*FC*) layers. Each convolution uses a 3×3 kernel and is followed by a PReLU activation function. The CNN progresses from the lower to higher depth by decreasing the spatial resolution using a 2×2 *max Pool* layer while gradually increasing the number of feature maps from 32 to 512. We use a *FC* layer of 512 neurons after the last *Conv* layer. We normalize (see ***FN*** above) the output of this *FC* layer and consider it as the desired feature representation of the input image. Finally, we use the *softmax* layer to compute the loss and optimize it during training. Our CNN model incorporates the residual learning framework [13], see Fig. 1 for the details. Overall, it comprises 40.5M parameters.

	Input	CoPr	CoPr	Pool	ResBl	CoPr	Pool	ResBl	CoPr	Pool	ResBl	CoPr	Pool	ResBl	FC	FN	Output
Filt Support	112 x 96	3	3	2	3	3	2	3	3	2	3	3	2	3	512		Softmax
Stride		1	1	2	1	1	2	1	1	2	1	1	2	1			
Pad		1	1	0	1	1	0	1	1	0	1	1	0	1			
# Filts		32	64		64	128		128	256		256	512		512			
# Replications		1	1	1	1	1	1	**2**	1	1	**5**	1	1	**3**	1	1	

Figure 1. Illustration of the proposed CNN architecture. ***CoPr*** indicates convolution followed by the PReLU activation function. ***ResBl*** is a residual block which computes $output = input + CoPr(CoPr(input))$. ***# Replication*** indicates how many times the same block is sequentially replicated in the CNN model. ***# Filts*** denotes the number of feature maps. ***FN*** denotes feature normalization.

3.2. Image pre-processing and face verification

Pre-processing: We maintain the same form of the 2D face image during training and testing. Our pre-processing steps are: (a) detect[2] face and landmarks using the MTCNN [45] detector; (b) normalize the face image by applying a 2D similarity transformation. The transformation parameters are computed from the location of the detected landmarks on the image and pre-set coordinates in a 112×96 image frame; and (c) convert to grayscale.

Face verification: We verify a pair of face images [14], templates [16] (contains multiple images and video frames) and videos [40] (given as frames) using the following steps:

1. *pre-process*: apply the pre-processing[3] stage described in the previous paragraph.

2. *extract facial feature/representation*: we use the trained CNN model to extract the facial feature descriptor. For an image i, we obtain its descriptor f_i by taking element-wise maximum of the features from its original $f_{i,o}$ and horizontally flipped version $f_{i,f}$. In order to perform verification based on template [16] and video [40], we obtain the descriptor for an identity by taking element-wise average of the features from all of the images/frames.

3. *compute verification score*: for a given pair of facial features, we compute the cosine similarity as the veri-

[2] In case of multiple faces, we take the face closer to the image center.

[3] If the landmarks detector fails we keep the face image by cropping it based on the given/detected bounding box.

fication score. We compare this score to a threshold to decide whether two images belong to the same person.

4. Experiments, Results and Discussion

Our experiments consist of first training the CNN model and then use it to extract facial features and perform different types (single-image [14, 3], multi-image or video [16, 40]) of face verification. In order to verify the effectiveness, we experiment on several datasets, namely LFW [14], IJB-A [16], YTF [40] and CACD [3].

4.1. CNN Training

We collect the training images from the cleaned[4] version of the MS-Celeb-1M [11] database, which consists of 4.47M images of 62.5K identities. We train our CNN model using only the identity label of each image. We use 95% images (4.2M images) for training and 5% images (232K images) for monitoring and evaluating the loss and accuracy. We train our CNN using the *stochastic gradient descent* method and *momentum* set to 0.9. Moreover, we apply *L2* regularization with the *weight decay* set to $5e^{-4}$. We begin the CNN training with a learning rate 0.1 for 2 epochs. Then we decrease it after each epoch by a factor 10. We stop the training after 5 epochs. We use 120 images in each mini-batch. During training, we apply data augmentation by horizontally flipping the images. Note that, during evaluation on a particular dataset, we do not apply any additional CNN training or fine-tuning and dimension reduction.

4.2. Results and Evaluation

Now we evaluate our proposed FR method, called *DeepVisage*, on the most commonly used and challenging facial image datasets based on their specified protocols.

Labeled Faces in the Wild (LFW) [14]: LFW is one of the most popular and challenging databases for evaluating unconstrained FR methods. It consists of 13,233 images of 5,759 identities. It has different evaluation protocols. We follow the *unrestricted-labeled-outside-data* protocol based on the recent trend [17]. The FR task requires verifying 6000 image pairs in 10 folds and report the accuracy. These pairs are equally divided into genuine and impostor pairs and comprises 7.7K images of 4,281 identities.

Table 1 provides the results of our method along with the other state-of-the-art methods. We observe that, our method achieves significant accuracy (99.62%) and among the top performers, despite the fact that: (a) we use single CNN, whereas Baidu [20] used 10 CNNs to obtain 99.77% and (b) we train CNN with comparatively much less amount of data and identities, whereas FaceNet [29] used 200M images of 8M identities to obtain 99.63%.

Table 1. Comparison of the state-of-the-art methods evaluated on the LFW benchmark [14].

FR method	# of CNNs	Dataset Info	Acc %
DeepVisage (proposed)	1	4.48M, 62K	99.62
Baidu [20]	10	1.2M, 1.8K	99.77
Baidu [20]	1	1.2M, 1.8K	99.13
FaceNet [29]	1	200M, 8M	99.63
Sparse ConvNet [33]	25	0.29M, 12K	99.55
DeepID3 [31]	25	0.29M, 12K	99.53
Megvii [46]	4	5M, 0.2M	99.50
LF-CNNs [38]	25	0.7M, 17.2K	99.50
DeepID2+ [32]	25	0.29M, 12K	99.47
Center Loss [39]	1	0.7M, 17.2K	99.28
MM-DFR [9]	8	0.49M, 10.57K	99.02
VGG Face [24]	1	2.6M, 2.6K	98.95
MFM-CNN [41]	1	5.1M, 79K	98.80
VIPLFaceNet [21]	1	0.49M, 10.57K	98.60
Webscale [35]	4	4.5M, 55K	98.37
AAL [43]	1	0.49M, 10.57K	98.30
FSS [37]	9	0.49M, 10.57K	98.20
Face-Aug-Pose-Syn [23]	1	2.4M, 10.57K	98.06
CASIA-Webface [44]	1	0.49M, 10.57K	97.73
Unconstrained FV [5]	1	0.49M, 10.5K	97.45
Deepface [34]	3	4.4M, 4K	97.35

The results in the Table 1 indicates saturation, because all of the methods achieve close to or more than human performance (97.53%). Besides, it is argued that matching only 6K pairs is insufficient to justify a method w.r.t. the real world FR scenario [19]. We address these issues by two ways: (a) employ more challenging evaluation metrics and (b) evaluate with the other challenging datasets. To this aim, first we follow the BLUFR LFW protocol [19] and measure the true accept rate (TAR) at a low false accept rate (FAR). BLUFR [19] protocol exploits all images of the LFW dataset and evaluates methods based on 10 trials experiments. Each trial computes 47M pair-matching scores (157K positives, 46.9M negatives), which is significantly higher than the 6K scores used in the standard protocol. Within this protocol, we compute the verification rate (VR) at FAR=0.1% and compare with the methods which reported results[5] in this protocol. We observe that: ***DeepVisage (proposed)*** $(98.65) > CenterLoss$[6] [39] $(92.97\%) > FSS$ [37] $(89.8\%) > CASIA$ [44] (80.26%) , i.e., our method obtains the best results published so far. Therefore, this result together with the Table 1 confirm the remarkable performance of *DeepVisage* on the LFW database. Next, we justify our method by evaluating it on the challenging IJB-A [16] dataset.

[4]We take the list of 5.05M faces provided by [41] and keep non-overlapping (with test set) identities which has at least 30 images after successful landmarks detection.

[5]We do not include results from Baidu [20] (VR@FAR: 99.11% for single CNN and 99.41% for 10-CNNs ensembles). The reason is that, we are not sure if they compute results based on the BLUFR protocol [19] or based on the 6K pairs. Note that, we obtain 99.7% on VR@FAR=0.1% using the 6K pair-matching scores of the standard protocol.

[6]Results computed from the features publicly provided by the authors.

Table 2. Comparison of the state-of-the-art methods evaluated on the IJB-A benchmark [16]. '-' indicates the information for the entry is unavailable. Methods which incorporates external training (ExTr) or CNN fine-tuning (FT) with IJB-A training data are separated with a horizontal line. VGG-Face result was provided by [27]. T@F denotes the *True Accept Rate at a fixed False Accept Rate (TAR@FAR)*.

FR method	ExTr	FT	T@F 0.01	T@F 0.001
DeepVisage (proposed)	N	N	0.887	0.824
VGG Face [24]	N	N	0.805	0.604
Face-Aug-Pose-Syn [23]	N	N	0.886	0.725
Deep Multipose [1]	N	N	0.787	-
Pose aware FR [22]	N	N	0.826	0.652
TPE [28]	N	N	0.871	0.766
All-In-One [25]	N	N	0.893	0.787
All-In-One [25] + TPE	Y	N	0.922	0.823
Sparse ConvNet [33]	Y	N	0.726	0.460
FSS [37]	N	Y	0.729	0.510
TPE [28]	Y	N	0.900	0.813
Unconstrained FV [5]	Y	Y	0.838	-
TSE [27]	Y	Y	0.790	0.590
NAN [42]	Y	N	0.941	0.881
TA [8]	Y	N	0.939	0.836
End-To-End [6]	N	Y	0.787	-

IARPA Janus Benchmark A (IJB-A) [16]: The recently proposed IJB-A database aims at raising the difficulty of FR by incorporating more variations in pose, illumination, expression, resolution and occlusion. It consists of 5,712 images and 2,085 videos of 500 identities. The FR task compares two templates. A template is a set of images and video-frames. The evaluation protocol requires computing the true accept rate (TAR) at a fixed false accept rate (FAR) with various values, e.g., 0.01 and 0.001.

Table 2 presents our results along with the other state-of-the-art methods. We separate the results (with a horizontal line) to distinguish two categories: (1) methods only using a pre-trained CNN; our method belongs to this category and (2) methods use additional learning, such as CNN fine-tuning and metric learning. From the comparison among the 1^{st} category of methods, we observe that, our method provides the best result for FAR at 0.001% and competitive (second best) at 0.01%. By comparing it to the 2^{nd} category we observe that, it is also very competitive and provide better results than numerous methods from this category. Besides, similar to [25, 28], it is possible to exploit our CNN features and further improve the final results with external learning, such as TA [8], NAN [42] and TPE [28].

YouTube Faces [40] (YTF): The YTF dataset is a widely used FR dataset of unconstrained videos. It consists of 3,425 videos of 1,595 identities. YTF evaluation requires matching 5000 video pairs in 10 folds and report average accuracy. Each fold consists of 500 video pairs and ensures subject-mutually exclusive property. We follow the *restricted* protocol of YTF, i.e., access to only the similarity information. We report our result in Table 3, along with the state-of-the-art methods. Results show that our method provides the best accuracy (96.24%).

Table 3 also provides the results (separated with a horizontal line) from *unrestricted* protocol, i.e., access to similarity and identity information of the test data. We observe that our method is very competitive to the best accuracy, although it follows the *restricted* protocol. The VGG Face [24] provides results with both protocols and shows that accuracy increases significantly (from *restricted*-91.6% to *unrestricted*-97.3%) when they learn their CNN feature embedding using the YTF training data. Based on this observation, we can predict that our result (96.24%) can be further enhanced by training or fine tuning with the YTF data.

Table 3. Comparison of the state-of-the-art methods evaluated on the Youtube Face [40]. ***Ad.Tr.*** denotes additional training is used.

FR method	Ad.Tr.	Accuracy (%)
DeepVisage (proposed)	N	96.24
VGG Face [24]	N	91.60
Sparse ConvNet [33]	N	93.50
FaceNet [29]	N	95.18
DeepID2+ [32]	N	93.20
Center Loss [39]	N	94.90
MFM-CNN [41]	N	93.40
CASIA-Webface [44]	Y	92.24
Deepface [34]	Y	91.40
VGG Face [24]	Y	97.30
NAN [42]	Y	95.72

Cross-Age Celebrity Dataset (CACD) [3]: CACD is a recently released dataset, which aims to ensure large variations of the ages in the wild. It consists of 163,446 images of 2000 identities with the age range from 16 to 62. CACD evaluation requires verifying 4000 image pairs in ten folds and report average accuracy. Table 4 reports the results of *DeepVisage* along with the state-of-the-art methods. It shows that our method provides the best accuracy. Moreover, it is better than LF-CNN [38], which is a recent method specialized on age invariant face recognition.

Table 4. Comparison of the state-of-the-art methods evaluated on the CACD [3] dataset. VGG [24] result is obtained from [41].

FR method	Accuracy (%)
DeepVisage (proposed)	99.13
LF-CNNs [38]	98.50
MFM-CNN [41]	97.95
VGG Face [24]	96.00
CARC [3]	87.60
Human, Avg.	85.70
Human, Voting [3]	94.20

The evaluations of *DeepVisage* (proposed method) across different challenging datasets prove that it not only achieves significant performance but also generalizes very well. It overcomes several of the difficulties which make unconstrained FR a challenging task.

Table 5. Analysis of the influences from training databases, size and number of classes. T@F denotes the *True Accept Rate at a fixed False Accept Rate (TAR@FAR)*.

Aspect	Add. info	Acc %	T@F 0.01
DB	***Size, Class***		
CASIA [44]	0.43M, 10.6K	99.00	0.988
Pose-CASIA [23]	1.26M, 10.6K	99.15	0.992
UMDFaces [2]	0.34M, 8.5K	99.15	0.992
VGG Face [24]	1.6M, 2.6K	98.40	0.975
MSCeleb [11]	4.2M, 62.5K	99.62	0.997
Min samp/id	***Size, Class***		
10	4.48M, 62.7K	99.56	0.996
30	4.47M, 62.5K	99.62	0.997
50	3.91M, 47.3K	99.60	0.997
70	3.11M, 33K	99.55	0.996
100	1.5M, 12.7K	99.23	0.991

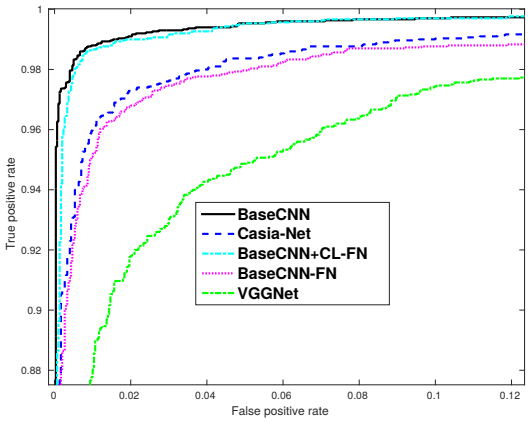

Figure 2. Illustration of the ROC plots for different CNN models evaluates on the LFW [14] dataset.

4.3. Analysis and Discussion

We perform further analysis to highlight the influences of several aspects, such as: (a) training datasets; (b) CNN models and depth; (c) normalization and (d) activation functions. Therefore, we modify and train our CNN model and observe the accuracy and TAR@FAR=0.01 on LFW. Table 5 presents the results.

First, we study the influence of training the proposed CNN with different datasets. It helps us to understand the capacity of the CNN to learn facial representation and identify the requirements to achieve better performance. The top part of Table 5 presents the analysis w.r.t. different datasets, from which we observe that: (a) CNN performance increased by training with larger number of images as well as identities, the best results are obtained with the largest dataset, i.e., MSCeleb [11]; (b) synthesized images help to enhance performance, we see this from the pose augmented CASIA [44, 23] dataset; (c) a dataset with more variations per identity helps even with a relatively lower number of images and identities, we see this by comparing the CASIA [44] and UMD [2] datasets; and (d) large number of images with smaller number of identities may not help, we see this from the VGG Face [24] dataset. Besides, we analyze the dataset uniformity or balance issue, i.e., number of images-per-identity, see bottom part of of Table 5. We use the MSCeleb [11] dataset for this experiment. We see that, while maintaining certain balance is necessary, it is equality important to train CNN with a larger dataset. We obtain the best performance by keeping only the identities with 30 images or more.

Besides the size/balance of datasets, it is also important to select an appropriate CNN model to learn from such datasets. To verify this, we trained CASIA-Net [44] (a shallower 10 layers CNN) with the MSCeleb [11] dataset and observed that it provided 98.6% accuracy, whereas we get 99.62% with the 27 layers CNN used in our method.

Next, we analyze the results based on different CNN components and models. Table 6 and Fig. 2 present the results with different forms, where we train all settings with the CASIA [44] dataset. Our observations are: (a) the proposed CNN model obtains better performance by including feature normalization (FN) before loss computation, we see this by comparing with the center loss [39] and without FN based results and (b) it obtains better accuracy than the other commonly used CNNs (for FR), such as the VGG-Net [24] and CASIA-Net [44]. Note that, we do not directly compare with other loss functions (within our CNN model) as the center loss [39] has been shown to be more efficient than those. Additionally, we trained our CNN with ReLU instead of PReLU and observe that it decreases accuracy by approximately 0.5%. In terms of complexity (measured with the number of parameters in Table 6), our model is more complex than the simpler models (Cas-Net and CN-mod). However, it is much simpler than the VGG-Net [24]. Results indicate that, while a simpler model may limit[7] the FR performance, a complex model is prone to overfitting. Perhaps this is the reason why the VGG-Net [24] requires additional fine-tuning on the target datasets. The above analyses justify the efficiency of our proposed CNN model.

Table 6. Study the influences from CNN related issues. All CNN models are trained with the CASIA [44] dataset. CL- center loss [39], FN- feature normalization. ***CN-mod*** modifies the ***Cas-Net*** [44] by replacing *Pool* layer with a *FC* layer of 512 neurons.

Settings	*# params*	Acc %	T@F 0.01
Base-CNN (***proposed***)	40.5M	99.00	0.988
Base-CNN - FN	40.5M	97.40	0.954
Base-CNN + CL - FN	44.8M	98.85	0.986
VGG-Net [24]	182M	95.15	0.883
Cas-Net [44]	6M	97.10	0.938
CN-mod	8M	97.50	0.956

[7]We train the CN-mod (see Table 6) with the MSCeleb dataset and observed that, compared to our proposed CNN model CN-mod provides lower results and generalizes poorly.

We observe that, feature normalization (FN) before the loss computation plays a significant role in the performance. In order to gain further insights, we conduct experiments and visualize the features of the MNIST digits in the 2D space. This is similar to the visualization recently shown in [39] and hence we also provide a comparison with the center loss (CL). The CNN is composed of 6 convolution, 2 pool and 1 FC (with 2 neurons for 2D visualization) layers. We optimize it using the softmax loss. Fig. 3 provides the illustration, from which we observe that: (a) FN provides a better feature discrimination in the normalized 2D space, see Fig. 3-b; (b) CL enforces the features towards its representative center and hence shows discrimination, see Fig. 3-c and (c) CL+FN does not provide much additional discrimination, see Fig. 3-b and Fig. 3-d. We also verified the usefulness of our FN trick with other CNNs and observe that it improves CN-mod (see Table 6) accuracy from 97.5% to 97.6% and VGG-Net accuracy from 95.15% to 98%. This indicates that the FN trick works better with deeper CNNs (in our CNN model, FN improves from 97.4% to 99.0%). These observations reveal that, by exploiting the FN appropriately we can ensure feature discrimination and hence no additional loss function, e.g., CL, is necessary.

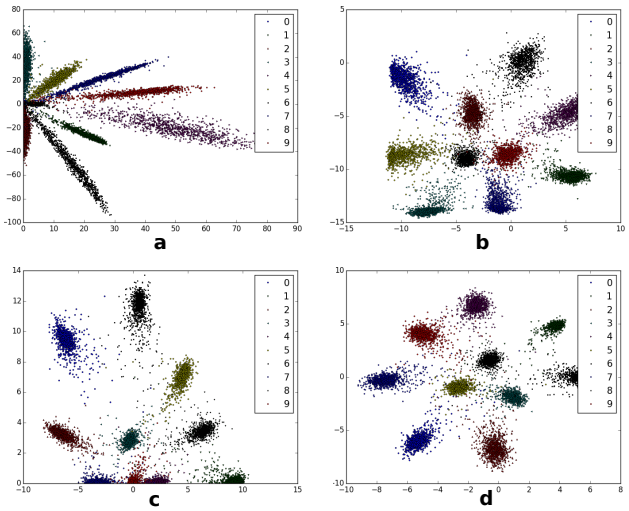

Figure 3. 2D visualization of the MNIST [18] digits features, which are obtained by using same baseline CNN model and training settings. CL [39] parameters are set to $\lambda = 0.003$ and $\alpha = 0.5$. **a.** CNN without FN and CL; **b.** CNN with FN; **c.** CNN with CL; and **d.** CNN with FN and CL.

Apart from the CNN components, we also experimented with image crop-size and color information. Our method provides similar results for both including/excluding random-crop based training. Moreover, we did not achieve any additional gain by using larger crop size and color image.

Finally, we investigate the incorrect results by observing the face image pairs in which *DeepVisage* failed. The *supplementary material* provides the illustrations of the false accept/reject cases from the different datasets. We observe that, on LFW it failed (11/20 error cases) when the eyes are occluded by glasses or a cap. Incorrect CACD results and higher false rejection rate indicate that our method (although provides best accuracy) encounters difficulties to recognize the same person from the images of different ages. Incorrect results from YTF often suffers from high pose and perhaps low image resolution. IJB-A results reveal that our method needs to take care of the face images with extreme pose variations. Indeed, during the IJB-A experiments, we are forced to keep a large number of images as un-normalized due to the failure of landmarks detection for them. Based on empirical evidences, we believe that these un-normalized faces cause the degradation of our performance. Besides, the results from YTF and IJB-A indicate that we may need to use a better distance computation strategy.

5. Conclusion

In this paper we present a single-CNN based FR method which achieves state-of-the-art performance and exhibits excellent ability of generalize across different FR datasets. Our method, called *DeepVisage*, performs face verification based on a given pair of single images, templates and videos. It consists in a deep CNN model which is simple and straightforward to train. Overall, *DeepVisage* is very easy to implement, thanks to the residual learning framework, feature normalization, softmax loss and the simplest distance. It successfully demonstrates that, in order to achieve state-of-the-art results it is not necessary to develop a complicated FR method by using complex training data preparation and CNN learning procedure. We foresee several future perspectives of this work, such as: (a) train CNN with a larger and more balanced dataset, which can be constructed by combining multiple publicly available datasets or by adopting the face synthesizing strategy [23] with the existing one; (b) enhance FR performance by incorporating failure detection based technique [30], particularly for face and landmarks detection and (c) incorporate better distance computation method for the template and video comparison, e.g., use softmax based distance [23].

Acknowledgment

This work was supported in part by the French Research Agency, l'Agence Nationale de Recherche (ANR), through the Jemime project (N° contract ANR-13-CORD-0004-02) and the PUF 4D Vision project funded by the Partner University Foundation.

References

[1] W. AbdAlmageed, Y. Wu, S. Rawls, S. Harel, T. Hassner, I. Masi, J. Choi, J. Lekust, J. Kim, and P. Natarajan. Face

recognition using deep multi-pose representations. In *IEEE WACV*, pages 1–9, 2016.

[2] A. Bansal, A. Nanduri, C. Castillo, R. Ranjan, and R. Chellappa. Umdfaces: An annotated face dataset for training deep networks. *arXiv:1611.01484*, 2016.

[3] B.-C. Chen, C.-S. Chen, and W. H. Hsu. Face recognition and retrieval using cross-age reference coding with cross-age celebrity dataset. *IEEE Trans. on Multimedia*, 17(6):804–815, 2015.

[4] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In *ECCV*, pages 566–579. 2012.

[5] J.-C. Chen, V. M. Patel, and R. Chellappa. Unconstrained face verification using deep cnn features. In *2016 IEEE WACV*, pages 1–9, 2016.

[6] J.-C. Chen, R. Ranjan, A. Kumar, C.-H. Chen, V. Patel, and R. Chellappa. An end-to-end system for unconstrained face verification with deep convolutional neural networks. In *Proc. of IEEE ICCV-W*, pages 118–126, 2015.

[7] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proc. of IEEE CVPR*, pages 539–546, 2005.

[8] N. Crosswhite, J. Byrne, O. M. Parkhi, C. Stauffer, Q. Cao, and A. Zisserman. Template adaptation for face verification and identification. *arXiv:1603.03958*, 2016.

[9] C. Ding and D. Tao. Robust face recognition via multimodal deep face representation. *IEEE Trans. on Multimedia*, 17(11):2049–2058, 2015.

[10] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, and G. Wang. Recent advances in convolutional neural networks. *arXiv:1512.07108*, 2015.

[11] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. *CoRR*, abs/1607.08221, 2016.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. of IEEE CVPR*, pages 1026–1034, 2015.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. of IEEE CVPR*, 2016.

[14] 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. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[15] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. of ICML*, pages 448–456, 2015.

[16] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *Proc. of IEEE CVPR*, pages 1931–1939, 2015.

[17] E. Learned-Miller, G. Huang, A. RoyChowdhury, H. Li, G. Hua, and G. B. Huang. Labeled faces in the wild: A survey. 2015.

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324, 1998.

[19] S. Liao, Z. Lei, D. Yi, and S. Z. Li. A benchmark study of large-scale unconstrained face recognition. In *Proc. of IEEE IJCB*, pages 1–8, 2014.

[20] J. Liu, Y. Deng, and C. Huang. Targeting ultimate accuracy: Face recognition via deep embedding. *arXiv:1506.07310*, 2015.

[21] X. Liu, M. Kan, W. Wu, S. Shan, and X. Chen. VIPLFaceNet: An open source deep face recognition sdk. *arXiv:1609.03892*, 2016.

[22] I. Masi, S. Rawls, G. Medioni, and P. Natarajan. Pose-aware face recognition in the wild. In *Proc. of IEEE CVPR*, pages 4838–4846, 2016.

[23] I. Masi, A. Tran, T. Hassner, J. T. Leksut, and G. Medioni. Do We Really Need to Collect Millions of Faces for Effective Face Recognition? In *ECCV*, 2016.

[24] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. *Proc. of BMVC*, 1(3):6, 2015.

[25] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa. An all-in-one convolutional neural network for face analysis. *arXiv:1611.00851*, 2016.

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

[27] S. Sankaranarayanan, A. Alavi, C. Castillo, and R. Chellappa. Triplet probabilistic embedding for face verification and clustering. *arXiv:1604.05417*, 2016.

[28] S. Sankaranarayanan, A. Alavi, and R. Chellappa. Triplet similarity embedding for face verification. *arXiv:1602.03418*, 2016.

[29] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proc. of IEEE CVPR*, 2015.

[30] A. Steger, R. Timofte, and L. Van Gool. Failure detection for facial landmark detectors. *arXiv:1608.06451*, 2016.

[31] Y. Sun, D. Liang, X. Wang, and X. Tang. DeepID3: Face recognition with very deep neural networks. *arXiv:1502.00873*, 2015.

[32] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *Proc. of IEEE CVPR*, pages 2892–2900, 2015.

[33] Y. Sun, X. Wang, and X. Tang. Sparsifying neural network connections for face recognition. In *Proc. of IEEE CVPR*, 2016.

[34] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proc. of IEEE CVPR*, pages 1701–1708, 2014.

[35] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Web-scale training for face identification. In *Proc. of IEEE CVPR*, pages 2746–2754, 2015.

[36] S. Theodoridis and K. Koutroumbas. *Pattern Recognition, Fourth Edition*. Academic Press, 4th edition, 2008.

[37] D. Wang, C. Otto, and A. K. Jain. Face search at scale. *IEEE TPAMI*, 2016.

[38] Y. Wen, Z. Li, and Y. Qiao. Latent factor guided convolutional neural networks for age-invariant face recognition. In *Proc. of IEEE CVPR*, pages 4893–4901, 2016.

[39] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *Proc. of ECCV*, pages 499–515. Springer, 2016.

[40] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *Proc. of IEEE CVPR*, pages 529–534, 2011.

[41] X. Wu, R. He, Z. Sun, and T. Tan. A light cnn for deep face representation with noisy labels. *arXiv:1511.02683*, 2015.

[42] J. Yang, P. Ren, D. Chen, F. Wen, H. Li, and G. Hua. Neural aggregation network for video face recognition. *arXiv:1603.05474*, 2016.

[43] H. Ye, W. Shao, H. Wang, J. Ma, L. Wang, Y. Zheng, and X. Xue. Face recognition via active annotation and learning. In *Proc. of ACM Multimedia*, pages 1058–1062. ACM, 2016.

[44] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv:1411.7923*, 2014.

[45] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, Oct 2016.

[46] E. Zhou, Z. Cao, and Q. Yin. Naive-deep face recognition: Touching the limit of lfw benchmark or not? *arXiv:1501.04690*, 2015.