Neurocomputing 275 (2018) 560–567

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Joint Bayesian guided metric learning for end-to-end face verification

Di Chen[a], Chunyan Xu[a,*], Jian Yang[a], Jianjun Qian[a], Yuhui Zheng[b], Linlin Shen[c]

[a] School of Computer Science and Engineering, Nanjing University of Science and Technology, 210094, China
[b] Jiangsu Engineering Center of Network Monitoring, School of computer and software, Nanjing University of Information Science and Technology, 210044, China
[c] School of Computer Science and Software Engineering, Shenzhen University, 518060, China

ARTICLE INFO

Article history:
Received 15 October 2016
Revised 18 July 2017
Accepted 2 September 2017
Available online 7 September 2017

Communicated by Qingshan Liu

Keywords:
Face verification
Convolutional neural network
Joint Bayesian model

ABSTRACT

In this work, we address the problem of face verification, namely determining whether a pair of face images belongs to the same or different subjects. Previous works often consider solving the problem of face verification in two steps: feature extraction and face recognition, resulting in a fragmented procedure. We argue that these techniques, although working well, fail to explicitly exploit a full end-to-end framework for face verification, which has received much attention and achieved significant improvements recently. In this paper, we propose a novel Joint Bayesian guided metric learning technique for dealing with the face verification task, which well integrates the above two steps of face verification into an end-to-end convolutional neural network (CNN) architecture. In the training stage, an initial neural network, which has the similar architecture with GoogLeNet CNN model, is firstly pre-trained by optimizing classification-based objective functions on the publicly available CASIA WebFace database. Based on constructed face pairs dataset from CASIA WebFace and LFW datasets, we then fine-tune the whole network parameters under the guide of the learned knowledge, which is obtained from the highly successful Joint Bayesian model. This guided learning procedure, which can also be seen as a metric learning technique, can further update network parameters for discriminating face pairs. In the testing process, the outputs by this unified network are discriminated with a threshold value to produce the ultimate prediction for the face verification task. Comprehensive evaluations over the LFW dataset well demonstrate the encouraging face verification performance of our proposed framework.

1. Introduction

Face verification, which aims to determine a pair of face images belongs to the same or different subjects, has drawn much research interest during the past few decades [1–6]. Although its performance has been improved substantially, face verification remains a challenge problem under complex and constrained conditions (e.g., multi-pose, illumination, expression, large age span, makeup, resolution, occlusion, etc.). Previous works often consider solving the problem of face verification in two steps: feature extraction and face recognition, resulting in a fragmented system. We argue that these techniques, although working well, fail to explicitly exploit a full end-to-end framework for face verification, which has received much attention and achieved significant improvements recently.

In this paper we aim to develop a unified framework for face verification. To this end, we reconsider the classical level technique. The most challenging point for the face verification task is how to compute the distance or similarity of two face images. Conventional metric learning methods [7–12] aim to directly learn the distance metric from the available training data. Due to the encouraging effectiveness of the nearest neighbor rule, existing works put emphasis on learning a similarity matrix of the Mahalanobis distance to further improve the performance of nearest neighbor classifiers. Liu et al. [13] made a thorough review of subspace and distance metric learning and proposed a unified framework for face verification. Among these approaches, low level features such as SIFT, LBP and HOG are usually employed, which typically suffer from representing these complicated image appearances.

Due to the increasing computing power and availability of large training data, convolutional neural network (CNN) has facilitated great advances in several vision tasks, e.g, face verification [4,5], image classification [14], and text detection [15], etc. In contrast to these carefully designed hand-crafted features utilized by conventional approaches, learned face representation with deep

* Corresponding author.
E-mail addresses: deechan1994@gmail.com (D. Chen), cyx@njust.edu.cn (C. Xu), csjyang@njust.edu.cn (J. Yang), csjqian@njust.edu.cn (J. Qian), zheng_yuhui@nuist.edu.cn (Y. Zheng), llshen@szu.edu.cn (L. Shen).

http://dx.doi.org/10.1016/j.neucom.2017.09.009
0925-2312/

neural network structures has show their great potential in the face verification task. For example, DeepFace [4] from Facebook, which can be seen as one of pioneer works devoted to face verification, derives a face representation from a multi-layer deep neural network and achieves human-level performance on the LFW dataset with an accuracy of 97.35%. A series of DeepID methods [5,16–18] follows on, increasing the performance on the LFW dataset incrementally and steadily. FaceNet [19] from Google yields an outstanding result with a massive dataset of 200 million identities and 800 million face pairs. Although these representative cases mark the success in the face verification with deep feature learning features, all the works mentioned above treat feature representation and face recognition separately. Recently, Sun et al. [20] propose a deep and wide architecture of the hybrid network and can unify feature extraction and recognition stages under a single network structure. However, the proposed hybrid deep network handles a pair of face images in two separate pipelines of deep network and not explicitly considers the fact that a single/unified pipeline can be optimized for learning face representation in the process of face verification.

In the literature of face verification, there are few works trying to unify feature learning and metric learning. Siamese architecture [2], which begins to take shape on this effort, employs a CNN as the raw-to-target space mapping function. The recent FaceNet [19] designs a triplet loss training strategy, which takes in three faces (a, p, n) at one time where (a, p) are congruous and (a, n) are incongruous. The optimization goal is to make the distance between a and p smaller than that of a and n. Both works can be viewed as training feature with a fixed metric, siamese architecture using L_1 norm while FaceNet adopting L_2 norm. However, the plain norm-based metric may not be powerful enough to reveal the discriminative information between feature vectors. DARI [21] takes a further step on, integrating metric learning and deep feature learning by factorizing Mahalanobis distance matrix as one fully connected layer upon a deep network. A hinge-loss like objective function designed in a triplet style is used to train the whole network. This work is the closest approach to ours except two significant differences. First, the top fully connected layer of DARI is interpreted as the factorization of a Mahalanobis distance matrix, while ours serve no more than a metric function approximator, not restricted to Mahalanobis distance. Second, the fully connected layers not only learn discrimination information from data like DARI does, but also distill knowledge from a guiding expert, a Joint Bayesian model in this case.

Based on the above observation, we propose to develop an end-to-end face verification framework for better integrating the above two steps of face verification. By jointly modeling two faces with an appropriate prior on the deep face representation, a Joint Bayesian model [22] is proposed to deal with the face verification problem and achieves effective performance on the LFW dataset. Inspired by the successful Joint Bayesian model, we present a novel Joint Bayesian guided metric learning technique for the end-to-end face verification, which can well unify the feature extraction and recognition stages into a CNN architecture. Specifically, a shared CNN model is firstly pre-trained on the publicly available CASIA WebFace dataset [23] by minimizing classification-based objective functions with ground-truth labels. We construct a large number of face pairs from CASIA WebFace and LFW datasets, and then further optimize the whole network parameters under the guide of learned knowledge, which is obtained from the highly successful Joint Bayesian model. From the aspect of face verification, training a shared CNN model and fine-tuning the whole network can be seen as the process of learning the face representation and recognition, respectively. In the testing stage, the outputs by the whole network are further discriminated with a threshold value to produce the ultimate prediction for the face verification problem. Our proposed framework can be indicated in Fig. 1.

The major contributions of our end-to-end face verification framework can be summarized as follows: (1) We propose a novel end-to-end face verification framework, which well integrates two steps of face verification (i.e., feature extraction and recognition) into a unified CNN architecture. (2) For better optimizing a whole CNN, we present a novel metric learning technique under the guide of learned knowledge, which is obtained from a highly successful Joint Bayesian model. (3) Our proposed face verification approach is a very general framework for discriminating image pairs, which also provides a novel technique for employing the learned knowledge of traditional methods beyond Joint Bayesian (e.g., Mahalanobis distance, SVM, etc.) in the process of optimizing a neural network. (4) The face verification results on the public LFW dataset well verify the effectiveness of the proposed neural network framework.

2. End-to-end face verification architecture

In this section, we first introduce the proposed end-to-end face verification architecture. We focus on the pre-training of an initial neural network for learning face representation, and then the entire network is fine-tuned with the Joint Bayesian guided metric learning technique. Finally, testing with the end-to-end CNN framework is conducted for face verification.

2.1. Network pre-training

In order to obtain a better face representation from a neural network, we first pre-training an initial neural network on the publicly available CASIA WebFace database [23] by optimizing multiple classification-based objective functions (e.g., cross-entropy loss, hinge loss, L_2 loss, ranking loss, etc.) under the guide of the ground-truth information.

In our implementation, our initial CNN network can be based on any general neural network structures, such as AlexNet, NIN, DSN, VGG, GoogLeNet and so on. We choose a similar network architecture to GoogLeNet [24], which has yielded exceptional performance in the 2014 ILSVRC classification challenge. For the GoogLeNet CNN structure, we learn a neural network by minimizing multiple cross-entropy loss functions $\mathcal{L}_G$ with the ground-truth label information of the whole training dataset X.

$$\begin{aligned} L_G &= L(X, l; \Theta) = \sum_{i=1}^{N} L(x_i, l_i; \Theta) \\ &= -\sum_{i=1}^{N}\sum_{c=1}^{C} \mathbf{1}_{\{l_i=c\}} \log p(l_c|x_i; \Theta), \end{aligned} \tag{1}$$

where Θ denotes the parameters of the neural network, $X = \{x_1, x_2, \ldots, x_N\}$ and $\{l_1, l_2, ..., l_N\}$ are the sets of training samples and its corresponding ground-truth label information, N and C is the total number of the training samples and classes.

For the face verification task, an effective face representation is necessary as the first step to achieve a good performance. This network pre-training process is just for learning one kind of face representation by training an initial CNN model on a huge face dataset. This deep face features learned from the CASIA WebFace dataset is employed for training a successful Joint Bayesian model of face verification, which can generate some useful knowledge to further guide the process of network optimization. Furthermore, the face representation is also the basic of learning a distance metric for face recognition in the whole end-to-end face verification framework.

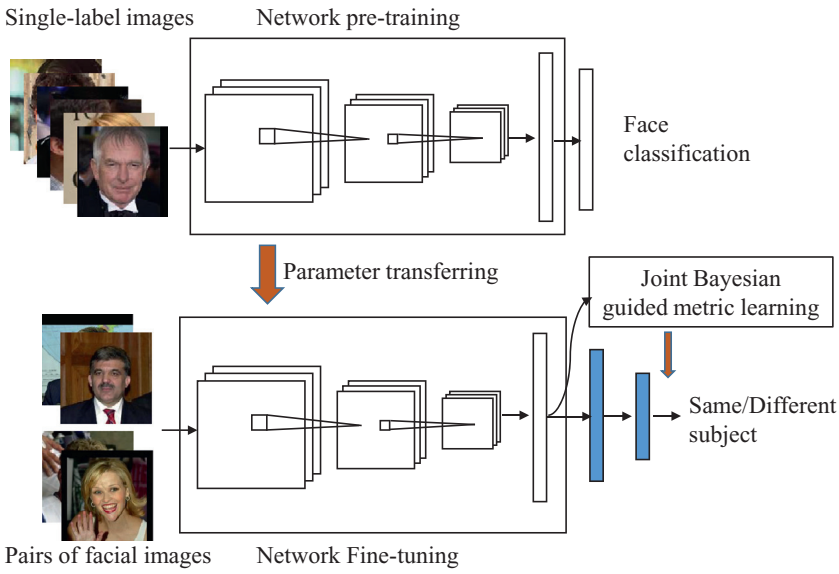

Fig. 1. Illustration of the Proposed End-to-end Framework of Face Verification. We first pre-train the network with a classification task on the whole CASIA WebFace dataset [23]. Then extra layers are added on top of the network and fine-tuned together with face pairs, under the guidance of Joint Bayesian model.

2.2. Network fine-tuning with joint Bayesian guided metric leaning technique

For constructing an end-to-end CNN structure, we would fine-tune the whole network by the proposed Joint Bayesian guided metric learning technique. Specifically, these added fully connected layers (i.e., marked as blue color in Fig. 1) of the final network are designed to produce the similarity/distance between a pair of face images, which are generated from the CASIA WebFace and LFW datasets.

A Joint Bayesian guided metric learning technique is also presented under the guide of learned knowledge, which is obtained from a successful Joint Bayesian model. This procedure can be abstracted by a function $s = f(\mathbf{x}_1, \mathbf{x}_2)$, where $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$ is the input feature pair and $s \in \mathbb{R}$ the similarity. It is reasonable to use fully connected layers as the function formulation since Multi-Layer Preceptrons(MLPs) with no less than one hidden layer are universal approximators [25]. Our network takes in two images $\mathbf{I}_1$, $\mathbf{I}_2$ and produces two features $\mathbf{x}_1$ and $\mathbf{x}_2$ at the feature fusion layer, then $|\mathbf{x}_1 - \mathbf{x}_2|$ is fed into the fully connected layers.

We devise a multi-loss training strategy aiming at approximating the Joint Bayesian model and capturing discrimination from data simultaneously. The first part of the loss is a squared difference constraint:

$$L_1(\mathbf{I}_1, \mathbf{I}_2; \Theta) = \frac{1}{2c}(\mathbf{N}(\mathbf{I}_1, \mathbf{I}_2; \Theta) - \mathbf{J}(\mathbf{x}_1, \mathbf{x}_2))^2 \tag{2}$$

where $\mathbf{N}(\mathbf{I}_1, \mathbf{I}_2; \Theta)$ is the similarity produced by the proposed network with Θ being the model parameter. $\mathbf{J}(\mathbf{x}_1, \mathbf{x}_2)$ is produced by a pre-trained Joint Bayesian model. $\mathbf{x}_1$, $\mathbf{x}_2$ are the corresponding feature vectors of image $\mathbf{I}_1$ and $\mathbf{I}_2$. c is a normalization coefficient. The proposed network imitates the act of Joint Bayesian and learns the superior performance of its mentor by minimizing the difference between the output of the two models.

The other part of the loss intends to enlarge the similarity between genuine pairs and reduce that of imposter pairs. It takes the following form:

$$L_2(\mathbf{I}_1, \mathbf{I}_2, y; \Theta) = (1-y) \times e^{\frac{1}{c}(\mathbf{N}(\mathbf{I}_1, \mathbf{I}_2; \Theta) - t)} + y \times e^{\frac{-1}{c}(\mathbf{N}(\mathbf{I}_1, \mathbf{I}_2; \Theta) - t)} \tag{3}$$

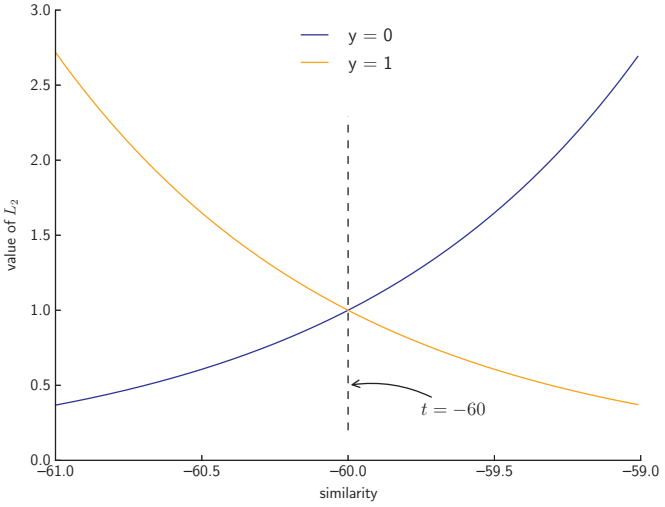

Fig. 2. Minimizing $L_2(\mathbf{I}_1, \mathbf{I}_2, y; \Theta)$ increases the similarity between genuine pairs and reduce that of imposter pairs.

where y stands for the same-or-not binary label. t is the best threshold which determines whether the output similarity is large enough to make a same-identity decision. It is computed on a validation set and updated every epoch.

Fig. 2 shows a naive demonstration of L_2. When $\mathbf{I}_1$, $\mathbf{I}_2$ belong to the same identity($y = 1$), L_2 is minimized by increasing the similarity. Similarly, when $\mathbf{I}_1$, $\mathbf{I}_2$ belong to different identities($y = 0$), L_2 minimization leads to a similarity decrease. The gradient is relatively large when the model makes a wrong decision, which can be seen as a punishment for the self-exploring on data apart from the guidance of Joint Bayesian.

The final loss function is a weighted average of L_1 and L_2.

$$L(\mathbf{I}_1, \mathbf{I}_2, y; \Theta) = w_1 L_1(\mathbf{I}_1, \mathbf{I}_2; \Theta) + w_2 L_2(\mathbf{I}_1, \mathbf{I}_2, y; \Theta) \tag{4}$$

Note that the normalization coefficient c mentioned above is to ensure that the gradient contribution of L_1 and L_2 remain roughly balanced, while w_1 and w_2 control the weight of L_1 and L_2. We found that the relatively better results are generally obtained by

using a considerably larger w_1 at first, then turn down w_1 and level up w_2 gradually as the training process goes on.

2.3. Testing for face verification

In the testing process, we feed each pair of face images into our end-to-end network framework and output a predicted similarity for the face verification task. We adopt the threshold technique to produce the ultimate prediction of the testing pairs of face images. For the ith pair of face images, the predicted result is given by

$$P_i = \begin{cases} 1, & \text{if } s_i > t \\ 0, & \text{else } s_i \le t \end{cases}, \tag{5}$$

where s_i is the similarity value outputted by the face verification network, t denotes the threshold value which can be defined empirically for determining the same (i.e., $P_i = 1$) or different (i.e., $P_i = 0$) subjects.

Our testing process with this end-to-end face verification network is different from other existing framework. Specifically, two steps of face verification in the training process have been carefully considered with the network pre-training, and metric learning, respectively. To ensure the stability of face representation, we can proceed each pair of face images with the same network architecture, and then compute its similarity for the face verification task.

2.4. Connection with knowledge distilling

To analyze how the learned knowledge guides the network learning process, we further present the connection between our proposed method and neural network (NN) knowledge distilling method. Recently, Hinton et al. [26] pointed out that a trained neural network with softmax activation usually has a lot more information contained in them than just a plain classifier. The *dark knowledge* is within the probability distribution, or correlation between categories outputted by softmax, telling a lot about how the trained model tends to generalize. To infuse the generalization ability into another model, a soft target generated by the softmax of the trained model is used as a supervisory signal during training, with the one-hot target supervising simultaneously.

Our multi-loss training strategy can be seen as a variant form of the knowledge distilling procedure. Apart from the same-or-not decision, Joint Bayesian expresses much more information by the relative value of similarity. For example, we have three faces: A, B and C. A and B are male faces, while C belongs to a female. A well trained model tends to produce a relatively large similarity between A and B, yet that of A and C should be relatively small. More properties other than gender information, such as age, skin color, race and etc., could be potentially implied by the relative similarity. L_1 is designed to distill the *dark knowledge* of Joint Bayesian by minimizing the difference of similarities produced by the two models. During the training process, the Joint Bayesian model acts as a mentor for the new model to imitate and learn generalization ability from.

However, we cannot only consider the learned knowledge from the Joint Bayesian model, but also employ the supervised information from the samples. If we train the model using only L_1, the model just tends to approximate Joint Bayesian. This means we are also gathering the mistakes made by Joint Bayesian, not surpassing its performance. Thus we need L_2 to act as a supervisor, correcting the wrong decisions according to the ground truth y. But training with L_2 alone will generally lead the model go wild in the parameter space and cause a relatively bad performance. We further prove the conclusion in Section 3.2. Therefore, it is important to balance the weight of L_1 and L_2 for optimizing an excellent neural network.

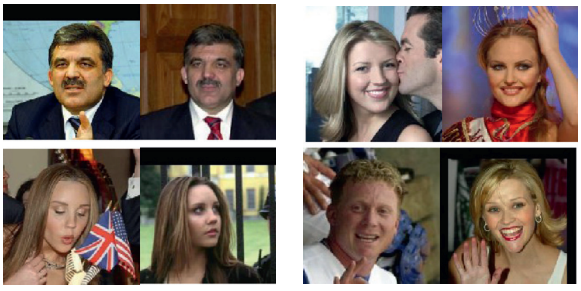

Fig. 3. Example pairs of face images from the LFW dataset.

It is worth to notice that the guiding expert with Joint Bayesian can be expanded to distance mapping algorithms or other discriminative models (including Support Vector Machine, Mahalanobis distance, etc.). Therefore, our work is a general framwork to exploit dark knowledge from any conventional models. Moreover, we can even adopt dark knowledge from multiple experts to guide the fully connected layers, exploit multi-view discriminative abilities and construct an ensemble model inexplicitly.

3. Experiments

In this section, we evaluate the effectiveness of our proposed face verification framework by comparing it with several existing state-of-the-art algorithms. We first introduce the experimental settings, and then report and analyze the experimental results, after that a further discussion about the proposed technique is given.

3.1. Experimental settings

Dataset: The LFW dataset [27] contains 13,233 uncontrolled face images of 5749 public figures with a variety of pose, lighting, expression, race, ethnicity, age, gender, clothing, hairstyles, and other parameters. All of these images are collected from the web. For fairly comparison with other existing methods, we follow the standard unconstrained verification protocol: using 6000 face pairs to tell if they are shot from the same person. Some example pairs of face images from the LFW dataset are shown in Fig. 3.

In the network pre-training stage, we use the CASIA WebFace dataset [23] which contains up to 10,575 subjects and 494,414 images. Apart from the complete dataset, we choose a subset to train the Joint Bayesian model and fully connected layers. The subset includes all of the subjects and 104,925 images with no more than 10 samples per-subject. Another validation set is formed to choose the threshold t for best verification accuracy, which consists of 2200 random image pairs from LFW training set(i.e., pairsDevTrain). All face images contained in the datasets mentioned above are gray scale. We first detect the face points using the method described in [28], then align the face portion to the center and cut out the 128×128 central patch as the ingredient image.

Implementation details: At the pre-training stage, we initialize the network weight using 'Xavier' as in [29]. We use SGD with a mini-batch size of 50. The learning rate starts from a base value of 0.01 and follows a polynomial decay policy where power is equal to 0.5 and max iteration step is 2400,000. We use a weight decay of 0.0002 and a momentum of 0.9. Features can be extracted from the latter two supervisory fully connected layer 'loss2/classifier' and 'loss3/classifier'. The feature fusion layer combines the two feature vectors by an averaging and produce the final 10,575-dim feature.

We extract the features of the CASIA WebFace subset and reduct their dimension to 3000 using PCA. A Joint Bayesian is then

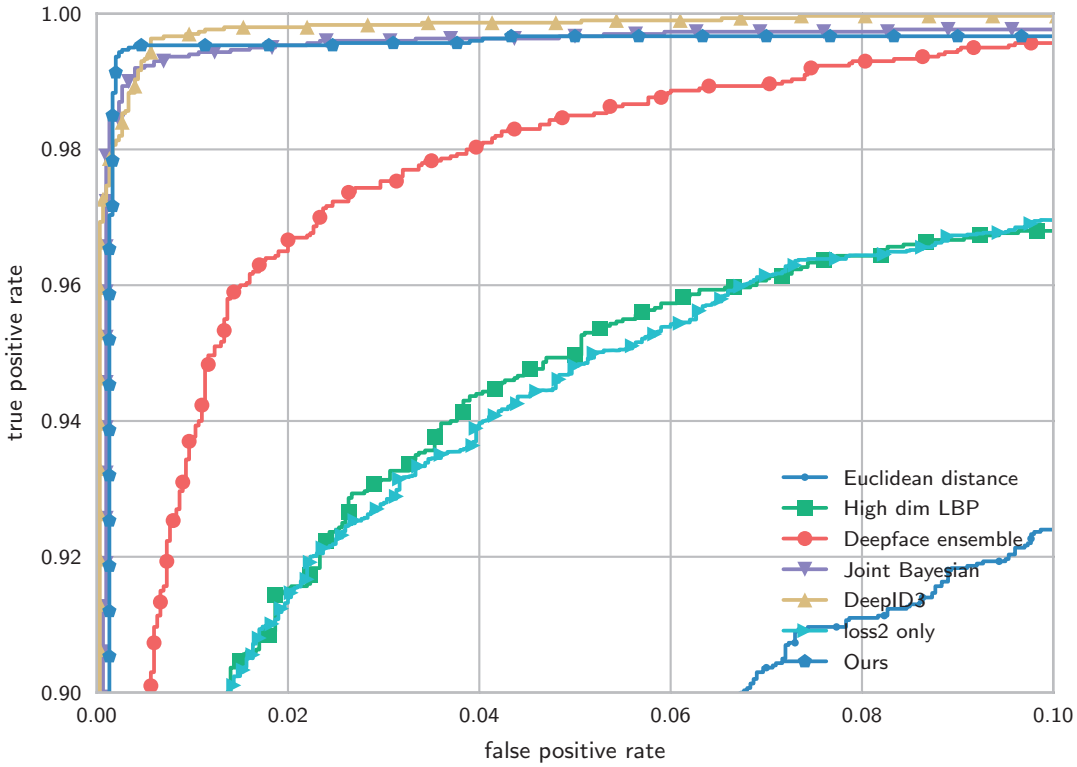

Fig. 4. ROC comparison with the previous results on LFW.

Table 1
Training process analyzing for our proposed framework.

Sub-epoch	Update method	Learning rate	Batch size	w_1	w_2
0	Adadelta	0.01	2000	0.8	0.2
11,296	Nesterov	1e−4	1000	0.5	0.5
12,257	Nesterov	5e−6	500	0.3	0.7
17,314	Nesterov	1e−6	500	0.2	0.8
18,763	Nesterov	1e−7	500	0.1	0.9

trained on the 3000-dim feature set. The layer dimension of our fully connected layers is "(3000, 1024, 1)" where the weight of the first hidden layer is initialized using the PCA projecting matrix trained above and the rest of them using 'Xavier'. P-relu [30] activation and Batch Normalization [31] are used in each layer. A dropout ratio of 0.5 is added to the last hidden layer.

As a routine, 'training an epoch' means the model has traverse the whole dataset. However, on a pair-wise training stage, the number of pairs becomes so large that it's impractical to traverse them all. Therefore, we define a sub-epoch here: in a sub-epoch, all categories are presented with two pairs, one of them being genuine and the other imposter. A batch size of 50 means there are 50 different categories with 100 pairs contained in the mini-batch. We train our network using Adadelta [32] and Nesterov momentum [33]. All the face pairs are randomly generated online. We tune the hyper-parameters manually as showed in Table 1 and keep the normalization coefficient c fixed at 600 and momentum fixed at 0.9. Note that the learning rate of the layers before the fusion layer is set to 0.0001.

3.2. Performance comparison

We compare our proposed framework with several existing face verification methods, including SIG-SML [34], high dimensional LBP [3], Deepface ensemble algorithm [4], DeepID [5], DeepID2 [16], GaussianFace [35] and DeepID3 [18]. Among these methods, SIG-SML, high dimentional LBP and GaussianFace adopt traditional metric learning approaches like Mahalanobis distance, Joint Bayesian and Gaussian process. DeepID, DeepID2 and DeepID3 also use Joint Bayesian as their discriminative metric, but with deep features instead of hand-crafted features. Deepface employs a siamese architecture to learn feature and metric simultaneously in an end-to-end style. These works cover entirely three categories of face verification frameworks: hand-crafted feature + conventional metric [3,34,35], deep feature + conventional metric [5,16,18] and deep feature + deep metric [4]. While our work can be seen as the combination of deep feature + deep metric with knowledge from conventional metric.

Table 2
Mean accuracy of face verification on LFW.

Method	Mean accuracy (%)
SIG-SML [34]	94.00
High dim. LBP [3]	95.17
Deepface Ensemble [4]	97.35
DeepID [5]	97.45
DeepID2 [16]	99.15
GaussianFace [35]	98.52
DeepID3 [18]	99.53
Euclidean distance†	91.80
L_2 loss only†	95.32
Joint Bayesian†	99.40
Ours	99.60

Table 2 shows the mean accuracy of our proposed framework on the LFW dataset and the corresponding comparison with the state-of-the-arts. Our proposed framework achieves a better mean accuracy of 99.60%, which is 4.43%, 2.25%, 0.07% higher than high dimensional LBP, Deepface ensemble algorithm and DeepID3 . Fig. 4. presents the ROC curves of our framework and other state-of-the-art methods for face verification on the LFW dataset. It can

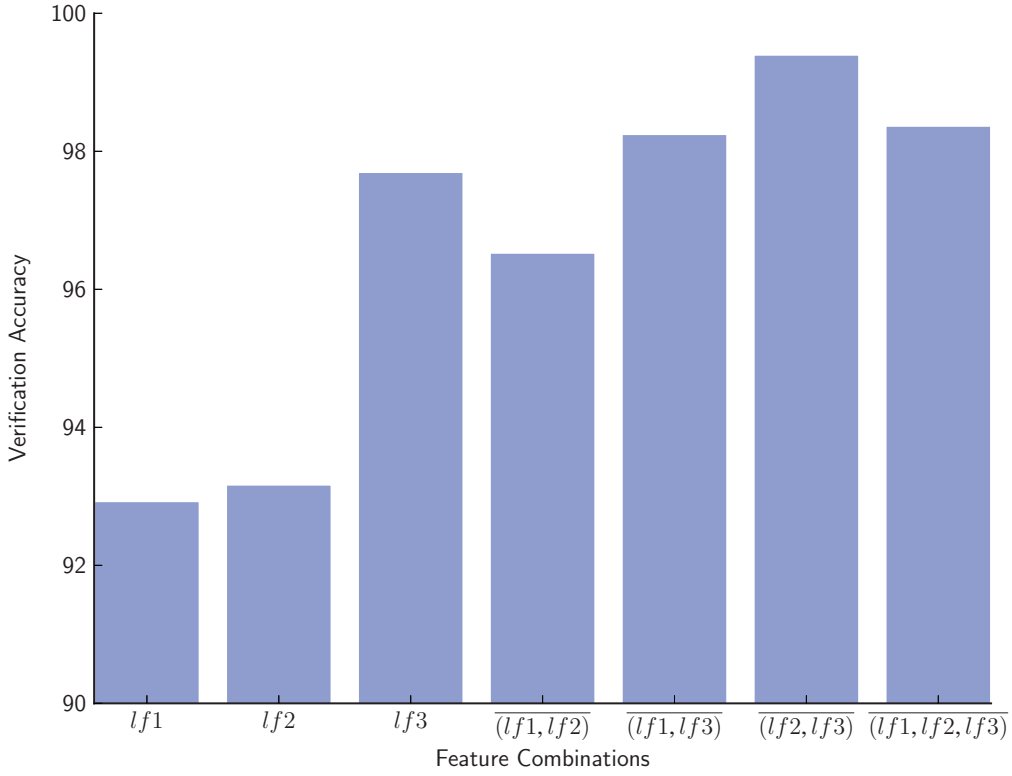

Fig. 5. Joint Bayesian Face Verification with Different Combinations of Features.

be observed that we can achieve comparable result with relatively less data, no complex multi-scale cropping and an end-to-end framework.

To better evaluate the effectiveness of the proposed Joint Bayesian guided metric learning framework, we further compare our result with some baseline models (i.e., marked with †), which are trained on the same deep representation extracted from our pre-trained network. A relatively good performance is achieved, e.g., 91.80% with Euclidean distance, 99.40% with Joint Bayesian model, respectively. We also report the result by training our metric with L_2 loss only, which is 95.32%. The large gap between this result and our final solution indicates that the guide of Joint Bayesian is a necessity for good performance. On the other side, combining L_1 and L_2 results a better accuracy than the vanilla Joint Bayesian model. In a word, the fact that a higher performance is achieved based on the same feature indicates the superiority of our proposed Joint Bayesian guided metric learning framework.

3.3. Algorithm analysis

We can see from Table 2 that a Joint Bayesian model trained on the pre-trained network feature set produces a relatively excellent result. Therefore it is reasonable to assume that the pre-trained network excels in feature extraction and we set the learning rate of layers before the final fully connected layers to a small constant. Another reason is that training a shallow part rather than the whole deep network needs much less computation and thus leads to a faster convergence.

The feature we used for verification is an average over the pre-softmax values of the second and third supervisory layer. As we know, there are three supervisory layers in the pre-trained network, denoted as 'loss1/classifier', 'loss2/classifier', and 'loss3/classifier'. Here we call the feature vectors yielded by these layers 'lf1', 'lf2', and 'lf3'. To choose the best of them, we train Joint Bayesian models on them and their liner combinations respectively. The training set is the CASIA 104,925 subset and LFW is used for testing. Fig. 5 shows the verification results. Obviously the average of 'lf2' and 'lf3', denoted as $\overline{(lf2, lf3)}$, works best with Joint Bayesian. Therefor our network connects 'loss2/classifier' and 'loss3/classifier' to the fusion layer followed by the fully connected layers for best performance.

Fig. 6. Face Images with Blocks Vary from 10×10 to 70×70.

Furthermore, we evaluate the robustness of our metric over its learning teacher Joint Bayesian on face images with occlusions. Random blocks of 10×10 to 70×70 in size are added to test images, as shown in Fig. 6.

The occluded regions in a face pair to be verified are generally different. Network in this experiment are the same as before. Fig. 7 shows the comparison results. Our fully connected layers achieve higher accuracies while the overall drop ratio of the two models, as block size increases, remains roughly the same. However, the accuracies of Joint Bayesian drops significantly faster when the blocks vary from 20×20 to 40×40 in size. Thus we could say our learned metric is more robust than its guiding expert.

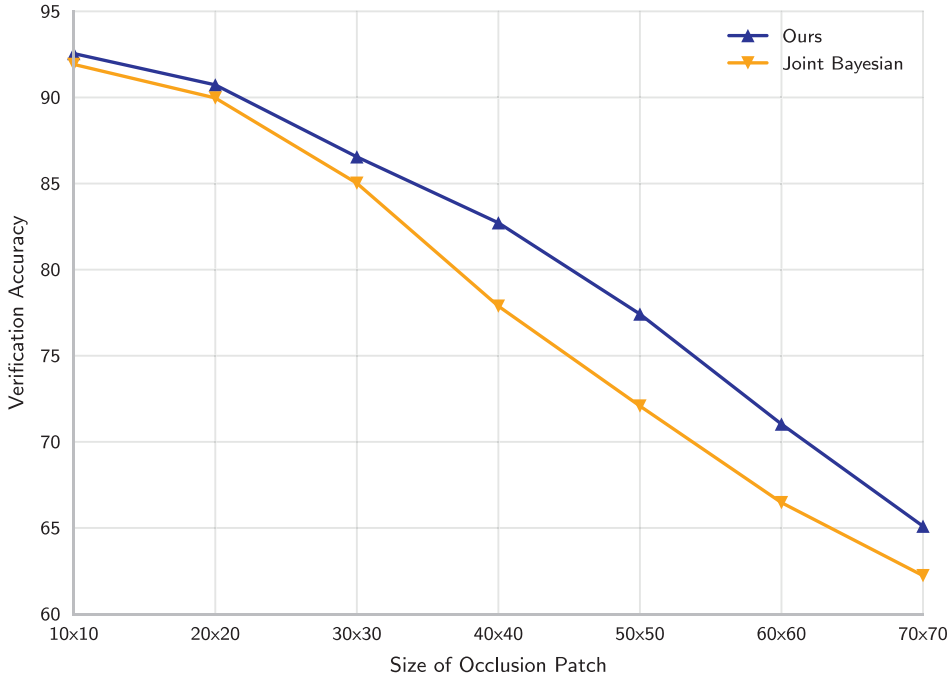

Fig. 7. Face Verification with Partial Occlusions.

4. Conclusion

In this work, we proposed an end-to-end CNN framework for the face verification task. This general scheme allows us to gracefully integrate two steps of face verification (i.e., feature extraction and face recognition) into a unified network architecture. By employing the learned knowledge from the successful Joint Bayesian model, a novel metric learning technique was built to jointly optimize the network parameters and predict the ultimate results for face verification effectively. Extensive experimental results on the LFW dataset clearly demonstrated the effectiveness of the proposed framework. In the further, we plan to further explore how to adequately utilize the learned knowledge from successfully traditional models (e.g., Mahalanobis distance, Support Vector Machine, etc.) for improving the discriminative capability of a deep neural network.

Acknowledgments

This work was supported by the National Science Fund of China [Grant Nos. 91420201, 61472187, 61233011, 61373063, 61602244, 61502235], the 973 Program No.2014CB349303, CCF-Tencent Open Research Fund, and Program for Changjiang Scholars and Innovative Research Team in University.

References

[1] J. Cheng, Q. Liu, H. Lu, Y.-W. Chen, Supervised kernel locality preserving projections for face recognition, Neurocomputing 67 (2005) 443–449.
[2] S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, in: CVPR, 2005, pp. 539–546.
[3] D. Chen, X. Cao, F. Wen, J. Sun, Blessing of dimensionality: high-dimensional feature and its efficient compression for face verification, in: CVPR, 2013, pp. 3025–3032.
[4] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: closing the gap to human-level performance in face verification, in: CVPR, 2014, pp. 1701–1708.
[5] Y. Sun, X. Wang, X. Tang, Deep learning face representation from predicting 10,000 classes, in: CVPR, 2014, pp. 1891–1898.
[6] Z. Cui, S. Shan, R. Wang, L. Zhang, X. Chen, Sparsely encoded local descriptor for face verification, Neurocomputing 147 (2015) 403–411.
[7] K.Q. Weinberger, J. Blitzer, L.K. Saul, Distance metric learning for large margin nearest neighbor classification, in: NIPS, 2005, pp. 1473–1480.
[8] J.V. Davis, B. Kulis, P. Jain, S. Sra, I.S. Dhillon, Information-theoretic metric learning, in: ICML, 2007, pp. 209–216.
[9] M. Guillaumin, J. Verbeek, C. Schmid, Is that you? metric learning approaches for face identification, in: ICCV, 2009, pp. 498–505.
[10] Y. Ying, P. Li, Distance metric learning with eigenvalue optimization, J. Mach. Learn. Res. 13 (2012) 1–26.
[11] Z. Cui, W. Li, D. Xu, S. Shan, X. Chen, Fusing robust face region descriptors via multiple metric learning for face recognition in the wild, in: CVPR, 2013, pp. 3554–3561.
[12] G. Wang, F. Zheng, C. Shi, J.-H. Xue, C. Liu, L. He, Embedding metric learning into set-based face recognition for video surveillance, Neurocomputing 151 (2015) 1500–1506.
[13] Q. Liu, D.N. Metaxas, A unified framework of subspace and distance metric learning for face recognition, in: Analysis and Modeling of Faces and Gestures: Third International Workshop, 2007, pp. 250–260.
[14] C. Xu, C. Lu, X. Liang, J. Gao, W. Zheng, T. Wang, S. Yan, Multi-loss regularized deep neural network, IEEE Trans. Circ. Syst. Video Technol. 26 (12) (2016) 2273–2283.
[15] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, X. Bai, Multi-oriented text detection with fully convolutional networks, in: CVPR, 2016, pp. 4159–4167.
[16] Y. Sun, Y. Chen, X. Wang, X. Tang, Deep learning face representation by joint identification-verification, in: NIPS, 2014, pp. 1988–1996.
[17] Y. Sun, X. Wang, X. Tang, Deeply learned face representations are sparse, selective, and robust, in: CVPR, 2015, pp. 2892–2900.
[18] Y. Sun, D. Liang, X. Wang, X. Tang, Deepid3: Face recognition with very deep neural networks, CoRR, 2015, arXiv:abs/1502.00873.
[19] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: a unified embedding for face recognition and clustering, in: CVPR, 2015, pp. 815–823.
[20] X.T. Yi Sun Xiaogang Wang, Hybrid deep learning for face verification, in: ICCV, 2013, pp. 1489–1496.
[21] G. Wang, L. Lin, S. Ding, Y. Li, Q. Wang, Dari: distance metric and representation integration for person verification., in: AAAI, 2016, pp. 3611–3617.
[22] D. Chen, X. Cao, L. Wang, F. Wen, J. Sun, Bayesian face revisited: a joint formulation, in: ECCV, 2012, pp. 566–579.
[23] D. Yi, Z. Lei, S. Liao, S.Z. Li, Learning face representation from scratch, CoRR, 2014, arXiv:abs/1411.7923.
[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: CVPR, 2015, pp. 1–9.
[25] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (5) (1989) 359–366.
[26] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, 2015, arXiv:1503.02531.
[27] G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled faces in the wild: a database for studying face recognition in unconstrained environments, Technical Report, University of Massachusetts, Amherst, 2007.
[28] Y. Sun, X. Wang, X. Tang, Deep convolutional network cascade for facial point detection, in: CVPR, 2013, pp. 3476–3483.
[29] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: AISTATS, 2010, pp. 249–256.

[30] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on imagenet classification, in: ICCV, 2015, pp. 1026–1034.
[31] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: ICML, 2015, pp. 448–456.
[32] M.D. Zeiler, Adadelta: an adaptive learning rate method, CoRR, 2012, arXiv:abs/1212.5701.
[33] Y. Nesterov, A method of solving a convex programming problem with convergence rate O (1/k2), Soviet Math. Doklady 27 (2) (1983) 372–376.
[34] X.-N. Hou, S.-H. Ding, L.-Z. Ma, C.-J. Wang, J.-L. Li, F.-Y. Huang, Similarity metric learning for face verification using sigmoid decision function, Vis. Comput. 32 (2016) 479–490.
[35] C. Lu, X. Tang, Surpassing human-level face verification performance on LFW with gaussianface, in: AAAI, 2015, pp. 3811–3819.

Di Chen received the B.Sc. degree in Computer Science and Technology from Nanjing University of Science and Technology, Nanjing, China, in 2016. Now he is a graduate student in the School of Computer Science and Engineering from Nanjing University of Science and Technology, Nanjing, 210094. His current research interests include machine learning, computer vision, and deep learning algorithms.

Chunyan Xu received the Ph.D. degree from the School of Computer Science and Technology, Huazhong University of Science and Technology, 2015. From 2013 to 2015, she was a visiting scholar in the Department of Electrical and Computer Engineering at National University of Singapore. Now she is a lecture in the School of Computer Science and Engineering from Nanjing University of Science and Technology, Nanjing, 210094, China. Her research interests include computer vision, manifold learning and deep learning.

Jian Yang received the BS degree in mathematics from the Xuzhou Normal University in 1995. He received the MS degree in applied mathematics from the Changsha Railway University in 1998 and the PhD degree from the Nanjing University of Science and Technology (NUST), on the subject of pattern recognition and intelligence systems in 2002. In 2003, he was a postdoctoral researcher at the University of Zaragoza, and in the same year, he was awarded the RyC Program Research Fellowship sponsored by the Spanish Ministry of Science and Technology. From 2004 to 2006, he was a postdoctoral fellow at Biometrics Centre of Hong Kong Polytechnic University. From 2006 to 2007, he was a postdoctoral fellow at Department of Computer Science of New Jersey Institute of Technology. Now, he is a professor in the School of Computer Science and Technology of NUST. He is the author of more than 80 scientific papers in pattern recognition and computer vision. His research interests include pattern recognition, computer vision and machine learning. Currently, he is an associate editor of Pattern Recognition Letters and IEEE Transactions on Neural Networks and Learning Systems.

Jianjun Qian received the BS and MS degrees in 2007 and 2010, respectively, and the PhD degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology (NUST) in 2014. Now he is an assistant professor in the school of computer science and engineering of NUST. His research interests include pattern recognition, computer vision and face recognition in particular.

Yuhui Zheng was born in Shanxi, China, in 1982. He received the B.S. degree in chemistry and his Ph.D. degree in computer science from Nanjing University of Science and Technology (NJUST), Nanjing, Jiangsu, China, in 2004 and 2009, respectively. From 2014 to 2015, he was a visiting scholar in the digital media laboratory of the school of Electronic and Electrical Engineering, Sungkyunkwan University, Korea. He is currently an associate professor at the School of Computer and Software in Nanjing University of Information Science and Technology. His research interests cover image processing, pattern recognition, and remote sensing information system.

Linlin Shen is currently a professor at School of Computer Science & Software Engineering, Shenzhen University. He received his Ph.D. degree from University of Nottingham, UK in 2005. Before joining Shenzhen University, he has been working as Research Fellow on MRI brain image processing at Medical school, University of Nottingham. His research interest covers pattern recognition, medical image processing and biometrics.