ATLANTIS PRESS

Advances in Computer Science Research, (ACSR), volume 52

2016 International Conference on Computer Engineering and Information Systems (CEIS-16)

Metric Learning Based Multi-Patch Ensemble for High Precision Face Verification

Dang-Dang Chen, Lan-Qing He, Zhong-Dao Wang, Sheng-Jin Wang

Tsinghua National Laboratory for Information Science and Technology,
Department of Electronic Engineering, Tsinghua University,
Beijing 10084, China
E-mail: chendd14@163.com, wgsgj@mail.tsinghua.edu.cn

***Abstract*-Face verification under video surveillance is an important issue in computer vision for decades. Several methods on automatic face verification have significantly raised the accuracy by using Gabor wavelets, high-dimensional LBP features, Fisher vectors, Joint Bayesian, etc. Especially the usage of Deep Convolution Neural Networks(CNNs), artificial intelligence beats humans for the first time on Labeled Faces in the Wild(LFW) face verification task. In order to further improve the verification accuracy, we propose an approach that combines a multi-patch deep CNN by using two step metric learning. Experiments show that our method archives an accuracy of 99.37% on LFW, and 92.80% on YouTube Faces(YTF), which is very competitive with the state-of-the-art.**

Keywords-face recognition; deep learning; CNN; LFW; metric learning

I. INTRODUCTION

Face verification is to tell whether two given faces belong to the same person or not. It is one of the core problems in computer vision and has been actively studied for decades. Many algorithms have been proven to work well on images collected in controlled settings. However, the performance of these algorithms often degrades significantly on images with large variations in pose, illumination, expression, aging, or occlusion.

In order to solve these problems, Deep Convolution Neural Networks(CNN) methods have been developed. DeepFace[5] was the first to train CNN by 4.4M facial images as a feature extractor for face verification tasks. It uses 3D alignment for data preprocessing, and a 4096-d vector for face representation. The accuracy of the method is 97.35% on Labeled Faces in the Wild(LFW)[2]. Sun et al. [7] achieved a result that surpass human performance for face verification on the LFW dataset using an ensemble of 25 simple Deep CNNs. And after that, Sun et al.[8] adopted in joint identification-verification supervision signal, leading to more discriminative features. Schroff et al.[9] trained deep CNN models with triple loss, thus the distance between the anchor and positive being minimized, while negative maximization. They achieve the state-of-the-art performance on LFW dataset. These works essentially demonstrate the effectiveness of the Deep CNN model for feature learning.

Another key component that can boost the performance of a face verification system is to learn a similarity metric from data for feature projection. There are many profitable metric learning approaches have been proposed. For instance, Hu et al.[17] learned a discriminative metric with deep neural network. Weinberger et al.[15] proposed Large Margin Nearest Neighbor(LMNN), it learns a Mahanalobis distance metric for k-nearest neighbor classification by semidefinite programming. Chen et al.[16] proposed a joint Bayesian approach for face verification which models the joint distribution of a pair of face images instead of the Euclidean distance between them.

In this paper, we will introduce our method based on deep CNNs for multi-patch feature extraction and two step metric learning for feature fusion. We trained our model with CASIA-WebFace[3] and VGGFace[6], and evaluate the performance of the proposed method on LFW dataset and YouTube Faces Database(YTF).

The main contributions of this paper are summarized as follows:

- A residual convolution neural network called RES-FaceNet is designed for extracting the high level feature of faces
- A multi-patch ensemble by using two step metric learning method is proposed.
- The proposed method obtains a very competitive result on LFW and YTF.

The rest of the paper is organized as follows. Details of our approach is given in Section II. Experimental results are presented in Section III. Finally, we conclude the paper in Section IV with a brief summary.

II. METHOD

Our approach consists of both training and testing. For training, we first perform face normalization on the CASIA-WebFace and VGGFace. Next, we train every Deep CNN on the normalized faces (or cropped patches) with CASIA-WebFace and derive the metric learning using the Deep CNN features of VGGFace dataset.

The overview of out method is shown in Figure 1.

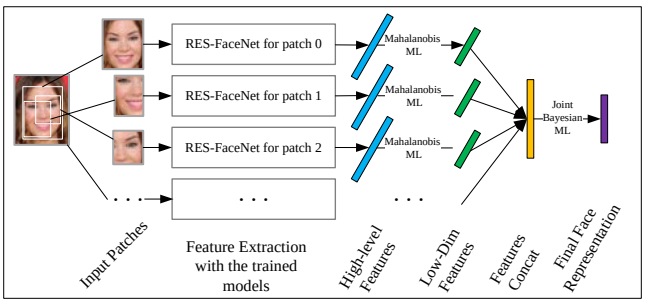

Figure 1. Overview of our method. At test stage, we first crop 8 patches from the normalized faces, and extract deep feature vector for every patch. Then Low Rank Mahalanobis Metric Learning is performed to compress the deep feature to a lower space. We concatenate the low-dimension features, and finally estimate the distance between faces using Low Rank Joint Bayesian Metric Learning.

A. Data Preprocessing

Before training the convolutional neural networks, we perform face detection and landmark detection on the CASIA-WebFace and VGGFace. Then, each face is aligned into the canonical coordinate with similarity transform using the landmark points. After alignment, face images are normalized to 125×160 pixels in RGB, shown in Figure 2.

Figure 2. Examples of aligned faces from CASIA-WebFace.

We cropped 8 patches from a normalized face in the training dataset according to the landmarks detected, and separately trained 8 CNN models with every patch. A set of examples is listed in Figure 3.

Figure 3. Patches used in our method.

B. Deep Face Feature Representation

Our Deep CNN is shown in Fig. 4. The deep network is constructed by 26 convolution layers, 5 max-pooling layers and 1 fully connected layers. Cross-layer feature transmission is added between some layers for residual learning and faster training of the network.

The face representation is a $512-d$ vector, and softmax cost function is used as supervised information.

Instead of using a commonly used activation Rectified Linear Unit(ReLU), we use Parametric Rectified Linear Unit(PReLU) instead. The formula of PReLU is as follows,

$$f(y_i) = \begin{cases} y_i & \text{if } y_i > 0 \\ a_i y_i & \text{if } y_i \le 0 \end{cases}, \qquad (1)$$

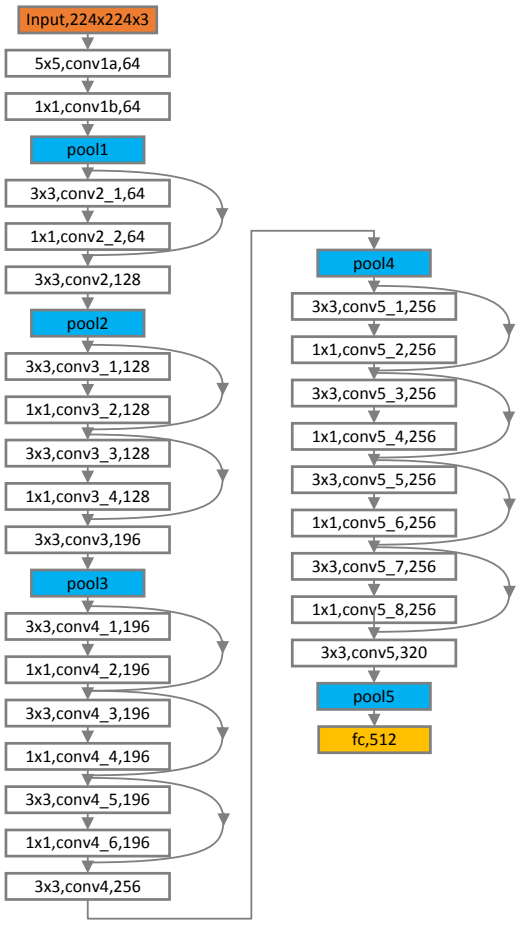

Figure 4. An illustration of the architecture of RES-FaceNet.

where y_i is the input of the nonlinear activation f on the ith channel, and a_i is a learnable coefficient controlling the slope of the negative part. It becomes ReLU when $a_i = 0$. Figure 5 shows the comparison between ReLU and PReLU.

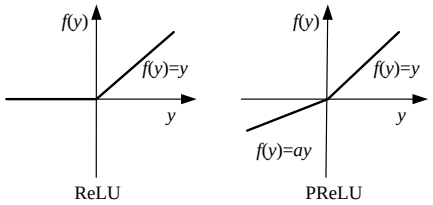

Figure 5. ReLU vs. PReLU. For PReLU, the coefficient of the negative part is not constant and is adaptively learned

As is known, the motivation of ReLU in the negative part is zero, which may loss much information. The method of PReLU adaptively learns the parameters jointly with the whole model. It introduces a very small number of extra parameters and negligible risk of overfitting. Experiments in [23]indicate that the learned coefficients of first convolution layer is significantly greater than 0, while the deeper convolution layers generally have smaller coefficients. This implies that the PReLU model tends to keep more

information in earlier stages and becomes more discriminative in deeper stages where the activations become "more nonlinear" at the meantime.

The Deep CNNs are implemented using Caffe [10].

C. Low Rank Mahalanobis Metric Learning for Feature Dimension Reduction

After extract features of every patch with CNN models, instead of PCA, we use Low Rank Mahalanobis Metric Learning(LRMML) to compress the features to a lower discriminative representation.

In detail, the aim is to learn a linear projection $W \in \mathbb{R}^{p\times d}$, $p < d$, which projects high-dimensional CNN features $x \in \mathbb{R}^d$ to low-dimensional vectors $Wx \in \mathbb{R}^p$, such that the distance between ith and jth can be written as

$$d_W^2(x_i, x_j) = \left\|Wx_i - Wx_j\right\|_2^2 = (x_i - x_j)^T W^T W(x_i - x_j) \quad ,(2)$$

where x_i is the CNN feature of the ith normalized face(or face patch), x_j is the CNN feature of the jth normalized face(or face patch), $W^T W \in \mathbb{R}^{d\times d}$ is the Mahalanobis matrix defining the metric. In order to optimize W, we optimize the distance in a large-margin framework as follows:

$$\arg\min_{W,b} \sum_{i,j} \max[1 - y_{ij}(b - (x_i - x_j)^T W^T W(x_i - x_j)), 0] \,, \quad (3)$$

where $b \in \mathbb{R}$ is the threshold, and y_{ij} is the label of a pair. If person i and j are the same $y_{ij} = 1$, otherwise $y_{ij} = -1$. To optimize the above formula, stochastic gradient method is used as follows. We equally train positive and negative pairs in turn.

$$W_{t+1} = \begin{cases} W_t, & \text{if } y_{ij}(b - d_W^2(x_i, x_j) > 1 \\ W_t - \gamma y_{ij} W_t \Gamma_{ij}, & \text{otherwise} \end{cases}$$
$$b_{t+1} = \begin{cases} b_t, & \text{if } y_{ij}(b - d_W^2(x_i, x_j) > 1 \\ b_t + \gamma_b y_{ij}, & \text{otherwise} \end{cases} \quad , \quad (4)$$

where $\Gamma_{ij} = (x_i - x_j)(x_i - x_j)^T$ and γ is the learning rate for W, and γ_b for the bias b.

After this projection, features of a patch is compressed from $512 - d$ to $128 - d$.

D. Low Rank Joint Bayesian Metric Learning for Feature Fusion

After LRMML, we concatenate features of the 8 patches to form a long vector, $1024 - d$. In order to acquire better verification accuracy, we learn Low Rank Joint Bayesian metrics which have achieved good performances on face verification problems. Instead of modeling the difference between two faces using $L2$ distance, this approach directly models the joint distribution of feature of both ith and jth face images $\{x_i, x_j\}$ as a Gaussian. Let $P(x_i, x_j \mid H_I) = N(0, \sum_I)$ when x_i and x_j belong to the same class, and $P(x_i, x_j \mid H_E) = N(0, \sum_E)$ when they are from different classes. In addition, each face vector can be represented by $x = \mu + \varepsilon$, where μ stands for the identity and ε is the face variation(e.g., lightings, pose, and expressions) within the same identity. The latent variable μ and ε are assumed to follow two Gaussian distributions $N(0, S_\mu)$ and $N(0, S_\varepsilon)$.

The log likelihood ratio of intra- and inter-classes, $r(x_i, x_j)$ can be computed as follows(more mathematical derivations are given in [16]):

$$r(x_i, x_j) = \log \frac{P(x_i, x_j \mid H_I)}{P(x_i, x_j \mid H_E)} = x_i^T A x_i + x_j^T A x_j - 2x_i^T G x_j \,. \quad (5)$$

Equation (5) can be written as

$$(x_i - x_j)^T A(x_i - x_j) - 2x_i^T B x_j, \quad \text{where } B = G - A \,. \quad (6)$$

Instead of directory estimate A and B, we define $A = WW^T$ and $B = VV^T$ where both W and $V \in \mathbb{R}^{p\times d}$, thus (6) can be written as

$$(x_i - x_j)^T W^T W(x_i - x_j) - 2x_i^T V^T V x_j \,. \quad (7)$$

We optimize the distance in a large-margin framework as follows:

$$\arg\min_{W,V,b} \sum_{i,j} \max[1 - y_{ij}(b - (x_i - x_j)^T W^T W(x_i - x_j) + 2x_i^T V^T V x_j), 0] \quad , (8)$$

where $b \in \mathbb{R}$ is the threshold, and y_{ij} is the label of a pair. If person i and j are the same, $y_{ij} = 1$, otherwise $y_{ij} = -1$. For simplicity, we denote $(x_i - x_j)^T W^T W(x_i - x_j) - 2x_i^T V^T V x_j$ by $d_{W,V}(x_i, x_j)$, W and V are updated using stochastic gradient descent as follows and are equally trained on positive and negative pairs in turn:

$$W_{t+1} = \begin{cases} W_t, & \text{if } y_{ij}(b_t - d_{A,B}(x_i, x_j)) > 1 \\ W_t - \gamma y_{ij} W_t \Gamma_{ij}, & \text{otherwise} \end{cases}$$
$$V_{t+1} = \begin{cases} V_t, & \text{if } y_{ij}(b_t - d_{A,B}(x_i, x_j)) > 1 \\ V_t + \gamma y_{ij} V_t (x_i x_j^T + x_j x_i^T), & \text{otherwise} \end{cases} \quad ,(9)$$
$$b_{t+1} = \begin{cases} b_t, & \text{if } y_{ij}(b_t - d_{A,B}(x_i, x_j)) > 1 \\ b_t + \gamma_b y_{ij}, & \text{otherwise} \end{cases}$$

where $\Gamma_{ij} = W_t(x_i - x_j)(x_i - x_j)^T$, γ is the learning rate for W and V , and γ_b for the bias b .

Compared with LRMML, we can find that LRMML is a special case of Low Rank Joint Bayesian Metric Learning, when $V = [0]$.

III. EXPERIMENTS AND RESULTS

In order to allow for a direct comparison to previous work, evaluation is performed on existing benchmark datasets.

1) LFW dataset: contains 13,233 images with 5,749 identities, and is the standard benchmark for automatic face verification. We follow the standard evaluation protocol defined for the "unrestricted with labeled outside data" using data external to LFW for training. We test on 6,000 face pairs and report the experiment results in TABLE I and TABLE II.

TABLE I. VERIFICATION RESULT OF EVERY MODEL ON LFW

Method	***Patch ID***	***Accuracy(avg)***
RES-FaceNet	0	98.55%
	1	98.35%
	2	98.25%
	3	97.57%
	4	97.75%
	5	97.08%
	6	97.55%
	7	97.13%
Final result	Ensemble of 8 patches	99.37%

As described in TABLE I, due to our excellent network design and the proper usage of Metric Learning, among the 8 patches, the best one can get an accuracy of 98.55%, and finally eight-patch embedding model achieves 99.37% pair-wise classification accuracy, which is already comparable to the best published results under this protocol, shown in TABLE II. Six of the misclassified pairs are listed in Fig. 6.

a) False negative samples

b) False positive samples

Figure 6. Failed cases in the LFW pair-wise verification task:(a)False negative samples. (b)False positive samples

From Fig. 6 we can conclude that expression change, occlusion, illumination change is still an important factor affecting the accuracy of face recognition.

2) YTF dataset: consists of 3,425 videos of 1,595 different people, with an average of 2.15 videos per person. The clip durations vary from 48 frames to 6,070 frames, with an average length of 181.3 frames. Also, we follow the "unrestricted with labeled outside data" protocol and report the result on 5,000 video pairs in TABLE III.

TABLE II. COMPARISON WITH STATE-OF-ART METHODS ON LFW

Method	***#Net***	***Protocol***	***Accuracy(avg)***
DeepFace[5]	7	unrestricted	97.35%
DeepID2[8]	25	unrestricted	98.97%
WebFace[3]	1	unsupervised	97.73%
FaceNet(NN1)[9]	-	unsupervised	99.63%
Face++[20]	-	unsupervised	99.50%
VGGFace[6]	1	unsupervised	97.27%
Ours	8	unsupervised	99.37%

Due to low resolution and motion blur, the quality of images in the YTF dataset is worse than LFW. We randomly select 50 samples from each video and compute the average distances. As shown in TABLE III, we obtain an accuracy of 92.80% on YTF.

TABLE III. COMPARISON WITH STATE-OF-ART METHODS ON YTF

Method	***#Net***	***Protocol***	***Accuracy(avg)***
DeepFace[5]	1	supervised	91.40%
WebFace[3]	1	unsupervised	90.60%
VGGFace[6]	1	unsupervised	91.60%
Ours	8	unsupervised	92.80%

IV. CONCLUSION

In this paper, we have developed a multi-patch ensemble by two step metric learning method to perform high accurate face verification. We first extract features of 8 patches cropped form a normalized face, and apply LRMML to compress the deep feature from $512-d$ to a lower 128 dimension. Then we concatenate the 8 low dimension feature vector to form a $1024-d$ feature vector as the face representation, and low rank joint Bayesian metric is used to estimate the distance between faces. Our method achieves a very competitive accuracy of 99.37% on LFW, and 92.80% on YTF.

Despite the high accuracy on LFW, result on YTF indicates that the face recognition in low resolution videos needs to be further studied, and how to effectively utilize multi-frame face images in video also need to be further

explored. Perhaps using LSTM instead of a simple averaging is an alternative to getting better results.

The use of multiple patches makes it 50ms for our method to extract feature of a single image on a NVIDIA GeForce 1080, acceleration work is a must. Faster RCNN shows us a way to speed up computations by sharing feature maps, which perhaps can be used in our method.

Future work will focus on better performance under expression change, occlusion, illumination change, and develop more efficient verification framework.

REFERENCES

[1] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 1-9.

[2] LFW, http://vis-www.cs.umass.edu/lfw/

[3] Yi D, Lei Z, Liao S, et al. Learning face representation from scratch[J]. arXiv preprint arXiv:1411.7923, 2014.

[4] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher vector faces in the wild. In British Machine Vision Conference, volume 1, page 7, 2013.

[5] Taigman Y, Yang M, Ranzato M A, et al. Deepface: Closing the gap to human-level performance in face verification[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014: 1701-1708.

[6] Parkhi O M, Vedaldi A, Zisserman A. Deep face recognition[C]. British Machine Vision Conference. 2015, 1(3): 6.

[7] Sun Y, Wang X, Tang X. Deep learning face representation from predicting 10,000 classes[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014: 1891-1898.

[8] Sun Y, Chen Y, Wang X, et al. Deep learning face representation by joint identification-verification[C]. Advances in Neural Information Processing Systems. 2014: 1988-1996.

[9] Schroff F, Kalenichenko D, Philbin J. Facenet: A unified embedding for face recognition and clustering[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 815-823.

[10] Jia Y, Shelhamer E, Donahue J, et al. Caffe: Convolutional architecture for fast feature embedding[C]. Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014: 675-678.

[11] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Web-scale training for face identification. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, pages 2746–2754, 2015.

[12] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 1701–1708, 2014.

[13] Patel V M, Wu T, Biswas S, et al. Dictionary-based face recognition under variable lighting and pose[J]. IEEE Transactions on Information Forensics and Security, 2012, 7(3): 954-965.

[14] Simonyan K, Parkhi O M, Vedaldi A, et al. Fisher Vector Faces in the Wild[C]. BMVC. 2013, 2(3): 4.

[15] Weinberger K Q, Blitzer J, Saul L K. Distance metric learning for large margin nearest neighbor classification[C]. Advances in neural information processing systems. 2005: 1473-1480.

[16] Chen D, Cao X, Wang L, et al. Bayesian face revisited: A joint formulation[C]. European Conference on Computer Vision. Springer Berlin Heidelberg, 2012: 566-579.

[17] Hu J, Lu J, Tan Y P. Discriminative deep metric learning for face verification in the wild[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014: 1875-1882.

[18] Huang Z, Wang R, Shan S, et al. Projection metric learning on Grassmann manifold with application to video based face recognition[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 140-149.

[19] Wu X, He R, Sun Z. A Lightened CNN for Deep Face Representation[J]. arXiv preprint arXiv:1511.02683, 2015.

[20] Zhou E, Cao Z, Yin Q. Naive-deep face recognition: Touching the limit of LFW benchmark or not?[J]. arXiv preprint arXiv:1501.04690, 2015.

[21] Sun Y, Chen Y, Wang X, et al. Deep learning face representation by joint identification-verification[C]. Advances in Neural Information Processing Systems. 2014: 1988-1996.

[22] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[J]. arXiv preprint arXiv:1512.03385, 2015.

[23] He K, Zhang X, Ren S, et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification[C]. Proceedings of the IEEE International Conference on Computer Vision. 2015: 1026-1034.