ATLANTIS PRESS

Advances in Intelligent Systems Research, volume 134

2nd International Conference on Control, Automation, and Artificial Intelligence (CAAI 2017)

An Efficient and Accurate Face Verification Method Based on CNN Cascade Architecture

Dangdang Chen[1,2,3], Lanqing He[1,2,3], Shengming Yu[1,2,3] and Shengjin Wang[1,2,3*]

[1]State Key Laboratory of Intelligent Technology and Systems;
[2]Tsinghua National Laboratory for Information Science and Technology;
[3]Department of Electronic Engineering, Tsinghua University, Beijing 10084, China
[*]Corresponding author

***Abstract*—Unconstrained face verification has been actively studied for decades in computer vision. Recent algorithms rely on Convolution Neural Network to further improve the accuracy. However, such algorithms tend to be time-consuming and computationally complex, which cannot meet the real-time requirements. In this paper, we propose an efficient and accurate face verification method based on Convolution Neural Network Cascade architecture. First, we use a compact network to handle most of the simple samples. Then, we use a complex network to handle a small number of hard samples. Finally, we use an ensemble of multi-patch networks with metric learning. Our method achieves an accuracy of 99.72% on LFW, which performs favorably against the state-of-the-arts. Furthermore, we significantly reduce time cost from 485ms to only 20ms on a single core i7-4790, which has strong practical value for real-time face verification systems.**

Keywords-face verification; deep learning; convolution neural network; metric learning

I. INTRODUCTION

In recent years, Convolution neural networks (CNNs) have achieved great success in computer vision, especially on face verification problem. The goal of face verification is to verify whether two given faces belong to the same person, which is very important for public security nowadays. Results on public datasets (e.g., LFW) keep climbing as more deep CNN based methods are introduced.

Learning invariant and discriminative feature representation is the first step for a face verification system using CNNs. Advances in deep learning methods have shown that compact and discriminative representation can be learned by using deep CNNs from very large datasets [14, 17, 18]. Taigman *et al.* [1] train CNN by 4.4M face images as a feature extractor for face verification tasks for the first time. It employs a 3D alignment method for data pre-processing and obtains an accuracy of 97.35% on LFW with 4096D feature vectors. Sun *et al.* [2] achieve an accuracy that surpass human performance for face verification on the LFW dataset using an ensemble of 25 simple Deep CNNs trained on weakly aligned face images. Sun *et al.* [19] adopt in joint identification-verification supervision signal which leads to more discriminative features. Schroff *et al.* [5] adapt the state-of-the-art deep architecture from object recognition to face recognition and train it on a large-scale unaligned private face dataset with a triplet loss. These work essentially demonstrates the effectiveness of the deep CNN model for feature learning.

Learning a similarity measure from data is another key component that can boost the performance of a face verification system. Many approaches have been proposed in the literature that essentially exploit the label information from face images or face pairs. For instance, Hu *et al.* [11] learn a discriminative metric within the deep neural network framework. Weinberger *et al.* [9] propose Large Margin Nearest Neighbor (LMNN) metric which enforces the large margin constraint among all triplets of labeled training data. Huang *et al.* [12] learn a projection metric over a set of labeled images which preserves the underlying manifold structure. Chen *et al.* [10] propose a joint Bayesian approach for face verification which models the joint distribution of a pair of face images instead of the difference between them, and the ratio of between class and within-class probabilities is used as the similarity measure. In our approach Joint Bayesian Metric Learning method is used.

As the network goes deeper and wider for addressing variations in pose, illumination, expression, age, cosmetics, and occlusion, the algorithms tend to be time-consuming and computationally complex. However, in most cases, faces can easily be distinguished by a shallower network.

In this paper, we propose a CNN cascade architecture, which contains three stages. The first stage is a simple network with relatively less powerful discriminating ability. The second stage comes with a network which is deeper and wider, and verification ability is relatively stronger. At the last stage, a multi-patch ensemble method is used, it has the best verification ability, but at the same time, needs more than 12 times the computational complexity compared to the network used in the first stage.

In our cascade architecture, feature extracted in the former stage will also be used in the later stages. With the advantage of using multiple patches and CNNs, we achieve competitive performance with the state-of-the-art methods. We traine our model with our own dataset called CvFaceDb, and evaluate the performance of the proposed method on the LFW dataset and YouTube Faces dataset (YTF). The CvFaceDb dataset contains 6,725,683 images with 110,438 identities and has no intersection with LFW or YTF.

The contributions of this paper are summarized as follows:

- A CNN cascade architecture is proposed for face verification, which works both accurately and efficiently.

- A lightened convolutional neural network named MFM-FaceNet and a residual convolution network called RES-FaceNet are designed for extracting the features of faces. The first network is simple with high speed of feature extraction, about 5ms on a single core i7-4790, while the second one archives an accuracy of 99.22% with a single model on the LFW dataset.

- The proposed framework obtains an accuracy of 99.72% on LFW and 93.10% on YTF, at the same time, costs only 20ms on a single core i7-4790 CPU per face image.

The rest of the paper is organized as follows. Details of the CNN cascade architecture is given in Section 2. Experimental results are presented in Section 3. Finally, we conclude the paper in Section 4 with a brief summary.

II. THE PROPOSED APPROACH

A common pipeline of CNN based face verification methods consists of two steps. First a deep CNN is used to extract a feature vector with relatively high dimension and the network can be supervised by multiclass and verification losses. Then, PCA, Joint Bayesian or metric learning methods are used to learn a more efficient low dimensional representation. In order to further improve face verification performance, multi-patch ensemble or fusion of multiple CNN models are often used.

In contrast, there are three stages in our cascade approach, and every stage can be viewed as a common face verification method. In the training phase, we first perform face detection and alignment on the CvFaceDb dataset to localize and align each face. We randomly choose 90% (99,394) people from CvFaceDb, train every deep CNN on the aligned faces (or cropped patches), and derive the Joint Bayesian metrics with the remaining 10% people, one by one. The overall pipeline of our method is shown in Figure 1.

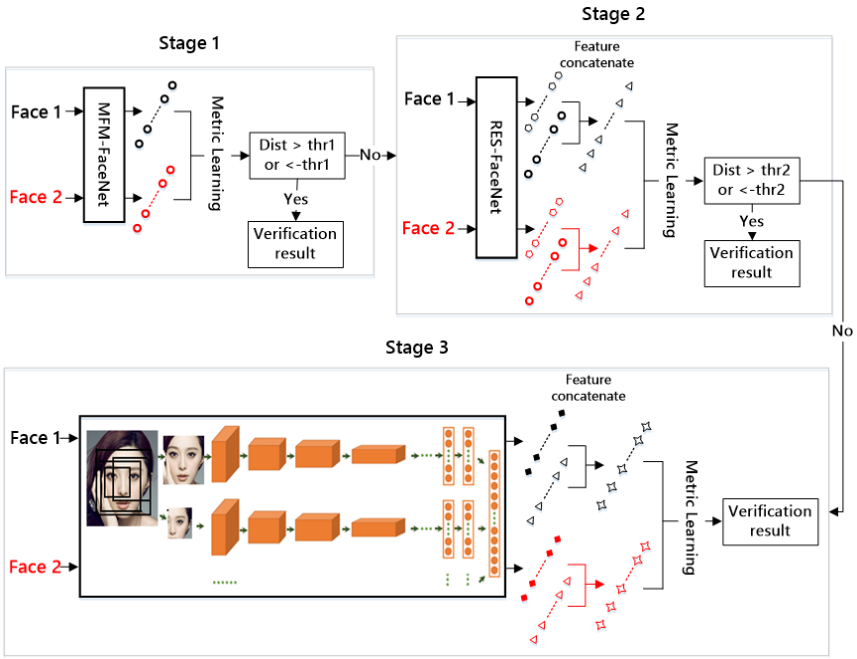

FIGURE I. OVERVIEW OF CNN CASCADE STRUCTURE FOR FACE VERIFICATION. JOINT BAYESIAN METRIC LEARNING USE FEATURES EXTRACTED FROM BOTH CURRENT STAGE AND THE FORMER STAGES. *THR*1 AND *THR*2 ARE PARAMETERS LEARNED BY EXPERIMENTS.

In testing phase, given a pair of test faces, we first extract their features with MFM-FaceNet (network used in the first stage), *fea*1 and *fea*2. Then Joint Bayesian Metric Learning is applied to evaluate the distance between the two given faces, *ML*(*fea*1, *fea*2). A threshold *thr*1 decides whether the face pair needs to be verified further. The sketch is shown in Figure 2.

$$\begin{cases} ML(fea1, fea2) > thr1, & \text{different person} \\ ML(fea1, fea2) < -thr1, & \text{same person} \\ others, & \text{need further evaluate} \end{cases} \quad (1)$$

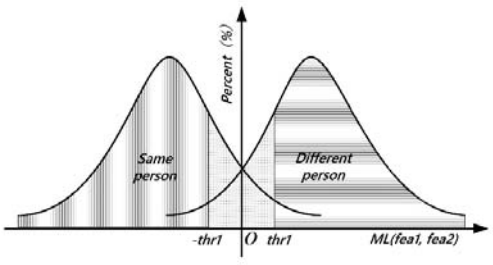

FIGURE II. A THRESHOLD NEED TO BE SET TO DECIDE WHERE THE FACE PAIR NEEDS TO BE EVALUATED FURTHER.

*thr*1 affects both accuracy and time cost of the whole cascade structure. While small *thr*1 will lead to high speed, it will result in more incorrect labels at the same time. On the contrary, large *thr*1 will lead to more further evaluations of test pairs, and more time will be cost. *thr*1 needs to be learned by experiments.

At the second stage, a network relatively more complicated with more powerful verification ability, called RES-FaceNet, is used. After extracting features of the given face pair, we concatenate them with the corresponding features extracted from the first stage, and use Joint Bayesian Metric Learning to estimate the distance between the test pair with the assembled features. Another threshold *thr*2 is set to decide whether the pair needs to be verified by the next stage.

At the last stage, a multi-patch ensemble method is used. An input image is cropped to 8 patches according to the detected landmarks, and we extract features of every patch. The final representation of a face is a concatenation of the 8 patches' features as well as the features extracted from the former stages. We also perform Joint Bayesian Metric Learning to make the final decision.

The details of each component of our approach are presented in the following subsections.

A. *Data Preprocessing*

Before training the CNNs, we perform face detection and landmark detection on the CvFaceDb dataset. Then, we align each face into the canonical coordinate with similarity transformation using the landmark points. After alignment, face images are normalized to 125×160 pixels in RGB, shown in Figure 3.

FIGURE III. EXAMPLES OF ALIGNED FACES IN CVFACEDB.

B. *Deep Face Feature Representation*

As mentioned above, our cascade approach contains three stages. Three sets of CNN models are trained with aligned faces.

At the first stage, a carefully designed network called MFM-FaceNet is used. The network contains 9 convolution layers, 4 max-pooling layers and 2 fully connected layers (including softmax). Max-Feature-Map (MFM) [13] activation function is used. MFM in convolution layers is a variation of the maxout operator, which is proved to be helpful in feature learning.

A Residual Network [20] is used in the second stage, called RES-FaceNet. The deep network is constructed by 26 convolution layers, 5 max-pooling layers, and 2 fully connected layers (including softmax). Cross-layer info transmission is added between some layers for better feature learning and faster training of the network. Instead of using a commonly used activation ReLU in RES-FaceNet, we use PReLU [21] instead. 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 [21] indicate that the learned coefficients of first convolution layer are 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. Figure 4 shows the comparison between ReLU and PReLU.

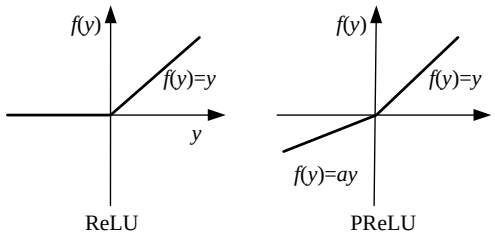

FIGURE IV. RELU VS. PRELU. FOR PRELU, THE COEFFICIENT OF THE NEGATIVE PART IS NOT CONSTANT AND IS ADAPTIVELY LEARNED.

The last stage uses the same network structure as in the second stage, but with multi-patch ensemble, we cropped 20 patches from a face according to the detected landmarks, and trained 20 CNN models separately. Finally, 8 patches are selected according to their performance. A set of examples is listed in Figure 5.

FIGURE V. PATCHES USED IN OUR METHOD.

The architectures of MFM-FaceNet and RES-FaceNet are shown in Figure 6.

C. *Joint Bayesian Metric Learning*

In order to acquire better verification accuracy, we learn joint Bayesian metrics which have achieved good performance on face verification problems. Instead of modeling the difference between two faces using $L2$ distance, this approach directly models the joint distribution of features of both *i*th and *j*th images $\{x_i, x_j\}$ as a Gaussian distribution. Let $P(x_i, x_j \mid H_I) = N(0, \Sigma_I)$ when x_i and x_j belong to the same class, and $P(x_i, x_j \mid H_E) = N(0, \Sigma_E)$ 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 $N(0, S_\varepsilon)$.

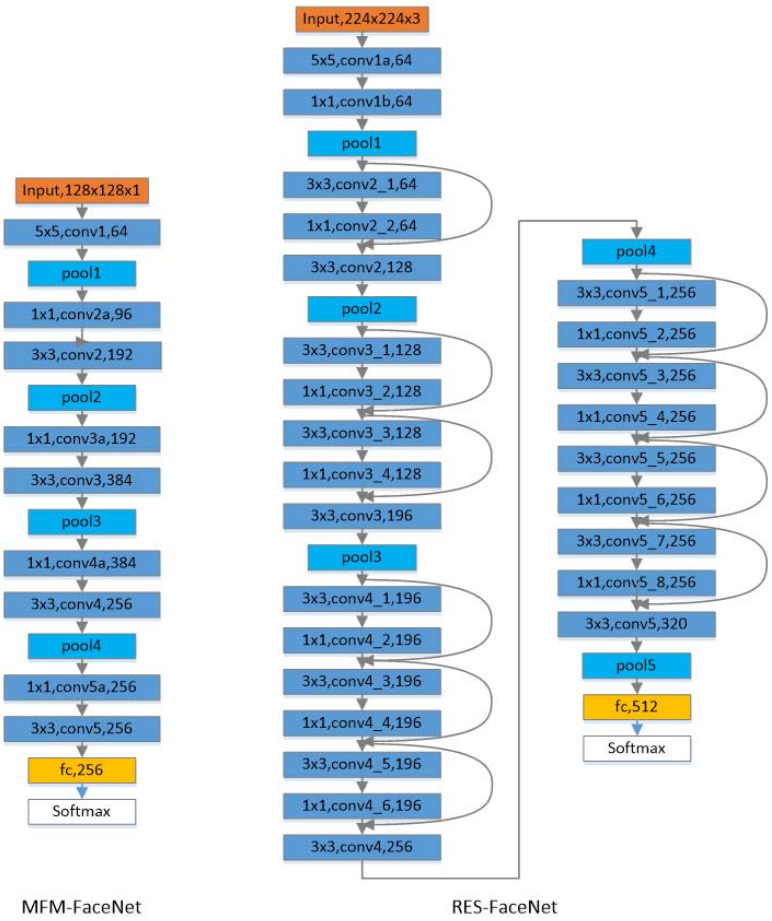

FIGURE VI. ILLUSTRATION OF THE ARCHITECTURE OF MFM-FACENET AND RES-FACENET.

The log likelihood ratio of intra- and inter-classes, $r(x_i, x_j)$ can be computed as follows,

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

where A and G are both negative semi-definite matrices.

(2) can be written as

$$(x_i - x_j)^T A(x_i - x_j) - 2x_i^T B x_j, \quad \text{where } B = G - A. \quad (3)$$

Instead of using the EM algorithm to estimate S_μ and S_ε, we optimize the distance in a large-margin framework as follows,

$$\arg\min_{A,B,b} \sum_{i,j} \max[1 - y_{ij}(b - (x_i - x_j)^T A(x_i - x_j) + 2x_i^T B x_j), 0], \quad (4)$$

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

$$A_{t+1} = \begin{cases} A_t, & \text{if } y_{ij}(b_t - d_{A,B}(x_i, x_j)) > 1 \\ A_t - \gamma y_{ij} \Gamma_{ij}, & \text{otherwise} \end{cases}$$
$$B_{t+1} = \begin{cases} B_t, & \text{if } y_{ij}(b_t - d_{A,B}(x_i, x_j)) > 1 \\ B_t + 2\gamma y_{ij} x_i x_j^T, & \text{otherwise} \end{cases}, \quad (5)$$
$$b_{t+1} = \begin{cases} b_t, & \text{if } y_{ij}(b_t - d_{A,B}(x_i, x_j)) > 1 \\ b_t + \gamma_b y_{ij}, & \text{otherwise} \end{cases}$$

where $\Gamma_{ij} = (x_i - x_j)(x_i - x_j)^T$, and γ is the learning rate for A and B, and γ_b for the bias b. We use random semi-definite matrices to initialize A and B, with $A = WW^T$ and $B = VV^T$, where W, $V \in \mathbb{R}^{d \times d}$, $w_{ij}, v_{ij} \sim N(0,1)$.

III. EXPERIMENTS AND RESULTS

Evaluations are performed on existing benchmark datasets for a direct comparison to previous work.

- **LFW dataset** contains 13,233 images with 5,749 identities, and it is a 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, test on 6,000 face pairs and report the experiment results in Table 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 3.

A. *Accuracy on LFW of every CNN model*

A total of 9 CNNs are trained in our cascade framework. We test every model on LFW, and the results are shown in Table 1.

TABLE I. VERIFICATION RESULTS ON LFW, AND CPU TIME COST BY EVERY METHOD.

Exp. id	Methods	*Patches*	*Accuracy (avg)*	*Time Cost(ms)*
1	MFM-FaceNet	0	97.28%	5
2	RES-FaceNet	0	99.22%	60
3		1	99.10%	60
4		2	99.05%	60
5		3	98.57%	60
6		4	98.75%	60
7		5	98.08%	60
8		6	98.26%	60
9		7	98.55%	60
10	RES-FaceNet patch0+ML	0	99.34%	60
11	RES-FaceNet multi-patch+ML	0-7	99.71%	480
12	Stage 1: MFM-FaceNet patch0+ML	0	98.28%	5
13	Stage 2: Ensemble of MFM-FaceNet patch0 and RES-FaceNet patch0 with ML	-	99.32%	65
14	Stage 3: Ensemble of MFM-FaceNet patch0 and RES-FaceNet multi-patch with ML	-	99.72%	485

From the table, we can conclude that:

Effect of different network structures, MFM-FaceNet and RES-FaceNet are trained with the same data, but the later gets an accuracy of 99.22%, reducing the error rate significantly by 71.32% when compared with the former. It shows the strong learning ability of the CNNs, and deeper networks can learn more discriminative features. Meanwhile, the later takes 12 times CPU time than the former network.

Effect of Joint Bayesian Metric Learning, by comparing Exp.1 with Exp.12 or Exp.2 with Exp.10, we can find that Joint Bayesian Metric Learning can further improve the accuracy in compare with Euclidean distance, as mentioned above.

Effect of multi-patch and multi-CNN ensemble, also multi-patches ensemble and multi-CNNs ensemble improves the final accuracy, when comparing Exp.11 with Exp.10 and Exp.14 with Exp.11.

B. *Experiments on the Parameter thr1 and thr2*

Parameters *thr*1 and *thr*2 affect not only verification accuracy of our approach, but also affect time cost of a given test pair. The following experiments are performed to determine the parameters.

The face verification accuracy and percent of pairs handled by the first and the second stage along with *thr*1 and *thr*2 changes from 0 to 0.35 are shown in Figure 7.

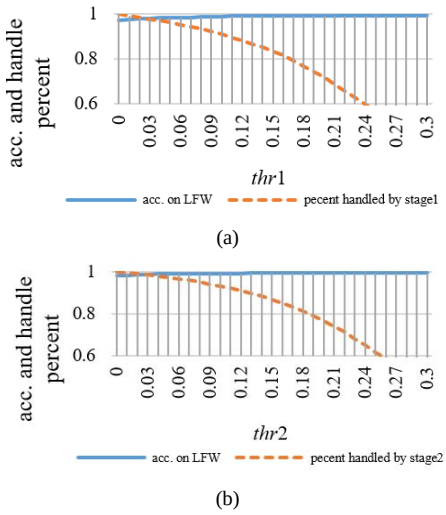

FIGURE VII. FACE VERIFICATION ACCURACY AND PERCENT OF PAIRS HANDLED BY SPECIFIC STAGE WITH DIFFERENT THRESHOLD.

As shown in Figure 7, the proper *thr*1 is around 0.15, when only 25% of the pairs beyond the verification ability of the first stage, and have no effect on the final verification accuracy. And the best value of *thr*2 is around 0.1, when 93% of the test pairs can be handled by the second stage.

Thus, the average time consuming of a given face is about 20.3ms.

C. *Final Result on LFW and YTF*

TABLE II. COMPARISONS WITH STATE-OF-THE-ART METHODS ON LFW

Methods	#Net	Accuracy (avg)	Protocol	Time Cost (ms)
DeepFace[8]	7	97.35%	unrestricted	-
DeepID2[19]	25	98.97%	unrestricted	-
WebFace	1	97.73%	unsupervised	29
FaceNet(NN1)[5]		99.63%	unsupervised	49
Face++[16]		99.50%	unsupervised	-
VGG[6]	1	97.27%	unsupervised	414
Ours	9	99.72%	unsupervised	20.3

As described in Table 2, our cascade framework achieves 99.72% verification accuracy on the LFW dataset, which is among the best published results under this protocol. At the same time, out approach costs only 20.3ms on a single core i7-4790, which is much less time cost than others.

The ROC curves on LFW are listed in Figure 8.

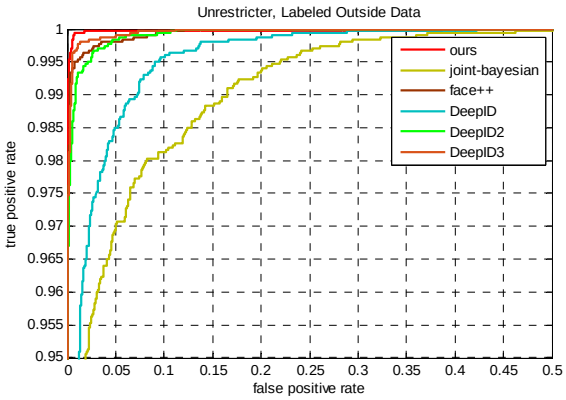

FIGURE VIII. COMPARISONS WITH THE STATE-OF-THE-ART METHODS ON LFW IN TERMS OF ROC CURVES

Figures 9 shows some failed cases on LFW pairwise verification task. From these examples, we can conclude that expression change, occlusion, illumination change are still important factors affecting the accuracy of face recognition.

(a) False negative samples

(b) False accept samples

FIGURE IX. FAILED CASES IN THE TASK ON LFW PAIR-WISE VERIFICATION TASK.

We also test our method on the YTF dataset and the result is shown in Table 3. 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 3, we obtain an accuracy of 93.10% on YTF.

TABLE III. COMPARISONS WITH STATE-OF-THE-ART METHODS ON YTF

Methods	#Net	Accuracy (avg)	Protocol	Time Cost (ms)
DeepFace[8]	1	91.40%	supervised	-
WebFace	1	90.60%	unsupervised	29
VGG[6]	1	91.60%	unsupervised	414
Ours	1	93.10%	unsupervised	20.3

IV. CONCLUSION

In this paper, we develop a cascade framework to perform efficient and accurate face verification, which contains three stages. The first stage of our framework is a simple network, which extracts features of faces very fast. In the second stage, a deeper and wider network named RES-FaceNet is used, it is more computing complex and time consuming, but with relatively higher verification accuracy. In the last stage, a multi-patch ensemble model is used, which has the best verification ability. Network in the first stage handle most of the easy examples and hard ones pass through to the following stages, the third stage comes with the final verification result. Our method achieves an accuracy of 99.72% on LFW while using a short time of 20.3ms on a single core i7-4790 on average, which has practical value for real-time face recognition systems.

In our current implementation, deep CNNs are trained separately for every patch in the 3rd stage, making the time-consuming linear growth with the patch number. If we train a multi-branch CNN, every patch shares the same feature-maps in lower stages and ROI-Pooling layers are used to pool out the correspondent feature-map, followed by patch-independent layers and Softmax layers. This can significantly reduce time costs and could be an interesting direction to be explored in the future.

REFERENCES

[1] 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.

[2] 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.

[3] Ahonen T, Hadid A, Pietikainen M. Face description with local binary patterns: Application to face recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2006, 28(12): 2037-2041.

[4] 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.

[5] 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.

[6] Parkhi, Omkar M., Andrea Vedaldi, and Andrew Zisserman. "Deep face recognition." British Machine Vision Conference. Vol. 1. No. 3. 2015.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

[15] Sun, Yi, et al. Deepid3: Face recognition with very deep neural networks. CoRR, abs/1502.00873, 2015.

[16] 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.

[17] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. 2012.

[18] He, Kaiming, et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Proceedings of the IEEE International Conference on Computer Vision. 2015.

[19] Sun, Yi, et al. Deep learning face representation by joint identification-verification. Advances in Neural Information Processing Systems. 2014.

[20] He, Kaiming, et al. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition.

[21] 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: 1026-103