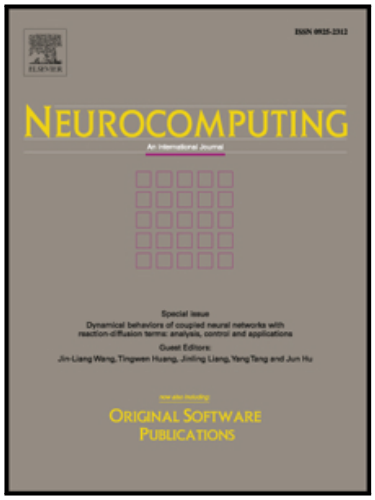
Communicated by Dr. Qingshan Liu

Accepted Manuscript



Improving Face Recognition with Domain Adaptation

Ge Wen, Huaguan Chen, Deng Cai, Xiaofei He

PII: S0925-2312(18)30112-7
DOI: 10.1016/j.neucom.2018.01.079
Reference: NEUCOM 19281

To appear in: *Neurocomputing*

Received date: 19 October 2017
Revised date: 8 January 2018
Accepted date: 28 January 2018

Please cite this article as: Ge Wen, Huaguan Chen, Deng Cai, Xiaofei He, Improving Face Recognition with Domain Adaptation, *Neurocomputing* (2018), doi: 10.1016/j.neucom.2018.01.079

Improving Face Recognition with Domain Adaptation

Ge Wen, Huaguan Chen, Deng Cai*, Xiaofei He

The State Key Lab of CAD&CG, Zhejiang University, No.388 Yu Hang Tang Road, Hangzhou 310058, China

Abstract

Nearly all recent face recognition algorithms have been evaluated on the Labeled Faces in the Wild (LFW) dataset and many of them achieved over 99% accuracy. However, the performance is still not enough for real-world applications. One problem is the data bias. The faces in LFW and other web-collected datasets come from celebrities. They are quite different from the faces of a normal person captured in the daily life. In other words, they are different in the face distribution. Replacing the training data with the right distribution is a simple solution. However, the photos of common people are much harder to collect because of the privacy concerns. So it is useful to develop a method that transfers the knowledge in the data of different face distribution to help improving the final performance. In this paper, we crawl a large face dataset whose distribution is different from LFW and show the improvement of LFW accuracy with a simple domain adaptation technique. To the best of our knowledge, it is the first time that domain adaptation is applied in the unconstrained face recognition problem with million scale dataset. Besides, we incorporate face verification threshold into FaceNet triplet loss function explicitly. Finally, we achieve 99.33% on the LFW benchmark with only single CNN model and similar performance even without face alignment.

Keywords: Face recognition, domain adaptation, face verification loss

Introduction

Face recognition is the problem of identifying a specific individual, rather than merely detecting the presence of a human face. It is widely used in public security, finance security, commercial domain and so on. Due to its wide applications, face recognition has become a core problem and one of the most popular research topics in computer vision. It includes two different but related tasks, face verification (are these two pictures the same person) and face identification (who is this person). Face verification can be extended to solve face identification task by repeating one-vs-one comparison. Nearly all recent methods have been evaluated on the Labeled Faces in the Wild (LFW) dataset [8]. In this paper, we focus on the face verification task and report the performance on the LFW benchmark as well.

The recent face recognition methods are based on convolutional neural network [13], and have made a great progress, even beating human beings on the LFW benchmark. But its performance is still not enough for real-world applications. One problem is the data bias[32]. The faces in LFW and other web-collected datasets come from celebrities. They are quite different from the faces of a normal person captured in the daily life. In other words, they are different in the face distribution. Replacing the training data with the right distribution is a simple solution. But the photos of common people are much harder to collect because of the privacy concerns. Besides, a generic recognition system is required to be transferred to a domain-specific application for performance. Both can be formulated as Domain Adaptation[19], which transfers the knowledge in the source domain to the target domain. In this paper, we crawl a large face dataset called TaoMM whose distribution is different from LFW and show the improvement of LFW accuracy with a simple domain adaptation technique.

In the testing phase of face verification, the distance between face pair is compared with a pre-computed threshold θ. If $dis < \theta$, the face pair is regarded as from the same person, otherwise from different person. There is a similar threshold θ in open-set face identification. Most face recognition methods don't consider the threshold in their optimization process explicitly. So there exists an optimization gap in their methods. By incorporating the threshold into FaceNet [23] triplet loss function explicitly, we reduce the LFW error rate by 26.9%. DDML[7] use a similar idea, but our final formulation is a triplet loss in an end-to-end framework.

Data augmentation is a very common preprocessing step for CNN based method[11], as a CNN model contains millions of parameters and is prone to overfitting. Most face recognition methods [29, 26, 25, 27, 4, 30] align face in both training and testing phase. It seems contradictory to apply data augmentation after face alignment. In this paper, we replace face alignment in training phase with aggressive data augmentation. Surprisingly, similar accuracy is achieved on LFW benchmark with or without face alignment during testing, which is different from prior results[18, 23].

*Corresponding author

Email addresses: zjuwenge@gmail.com (Ge Wen), chenhuaguanzju@gmail.com (Huaguan Chen), dengcai@cad.zju.edu.cn (Deng Cai), xiaofeihe@cad.zju.edu.cn (Xiaofei He)

Our contributions can be summarized as follows:

- We crawl a million scale face dataset called TaoMM, whose distribution is different from LFW, and we show the improvement of LFW accuracy with a simple domain adaptation technique even with a million scale target domain dataset.
- Face verification threshold θ is incorporated into FaceNet triplet loss function explicitly, with which the error rate on LFW is reduced by 26.9%.
- We achieve 99.33% on the LFW benchmark with only single CNN model and similar performance even without face alignment by applying aggressive data augmentation. We achieve 99.28% without face alignment which is better than FaceNet [23] 98.87% under the same circumstance.

Related Work

Our method is related to numerous works on face recognition and domain adaptation, which we briefly discuss below.

Face Recognition

Owing to deep learning, lots of breakthroughs have been made in recent years in face recognition [29, 26, 25, 27, 4, 23, 30].

[29, 26, 18, 14] train a face feature extractor by employing classification loss. Then [29] uses weighted χ^2 distance as face verification metric which is trained using a linear SVM. [26] reduces the feature dimension to 150 by PCA and learns a Joint Bayesian model [2] with the features. [18, 14] tune the extracted feature for verification in Euclidean space by using a metric learning method with a triplet loss training scheme. In order to develop more effective feature representations, [25, 27] train the feature extractor with joint classification and verification loss. [30] proposes a new supervision signal called center loss. By combining classification loss and center loss, they train a robust CNN to obtain discriminative features. In addition to the preceding two stages methods, [23] employs an end-to-end learning process which is the same as ours. It directly learns an embedding into an Euclidean space for face verification by triplet loss.

Most face recognition methods [29, 26, 25, 27, 4, 30] align face in both training and testing phase. Several complex face alignment methods have been developed, *e.g*. 2D similarity transformation [29, 26, 25, 27, 4, 30], 3D alignment [29, 4], frontalization [29]. [18, 23] find that using 2D alignment on training data only provides slightly or no performance improvement but performing 2D alignment on testing images does improve some performance. [18] augments data by random cropping and flipping, but most other methods don't as they have already aligned the face images. In this paper, we employ aggressive data augmentations including random cropping, flipping, rotation, scaling, color channel augmentation *etc*.

Domain Adaptation

Domain adaptation aims to transfer knowledge between related source and target domains whose distributions are different [19]. Many domain adaptation (or transfer learning) approaches have been proposed for computer vision applications [19, 5, 17, 3]. [5, 17] learn the features on the large-scale ImageNet [22] dataset in a supervised setting first, and then transfer them to different tasks with different labels. The key idea is that the internal layers of CNN can act as a generic extractor of image representation, which can be pre-trained on one dataset *e.g*. ImageNet. [3] proposes a novel double-path deep domain adaptation network to model the data of clothes images from constrained and unconstrained conditions jointly.

There are several prior works which apply domain adaptation to face recognition[20, 24, 16, 10, 1]. CMU Multi-PIE face dataset [6] contains 337 subjects with 15 poses, 20 illuminations, 6 expressions and 4 different sessions, which is the most popular dataset for applying domain adaptation method to face recognition. In most experiments [20, 24, 16], frontal faces were taken as the source domain and different poses were taken as the target domain. [10] proposes an unsupervised domain adaptation method via targetizing the source domain images bridged by the common subspace learning and applies it to domain adaptation across view angle, ethnicity and imaging condition. The datasets used in the preceding methods are captured in the lab with lesser variance. But in this paper, data of both source and target domain are captured in the wild and contain million scale images. [1] presents a generative Bayesian transfer learning algorithm and tests on challenging datasets.

Method

Similarly to most recent face recognition methods, we employ a deep convolutional neural network which learns its weights directly from the pixels of the face. By using large dataset of labelled faces, CNN model can learn the invariance to pose, illumination and other variational conditions.

The framework of training contains pre-training and training phase as shown in Figure 1. The following sections will describe the details of both phases as well as the network structure.

Domain Adaptation

In this paper, the datasets of both source and target domain are large enough to train a model from scratch. Here we want to employ all data to improve the face recognition performance in the target domain. But simply combining the data of source and target domain and throwing into the model achieves worse performance than the model only trained by the data of target domain.

In order to transfer the knowledge in source domain to target domain, we employ a simple domain adaptation (or transfer learning) technique similar to [5, 17], which is achieved by pre-training.

In the pre-training phase, the deep CNN model is bootstrapped by considering the problem of classifying N unique

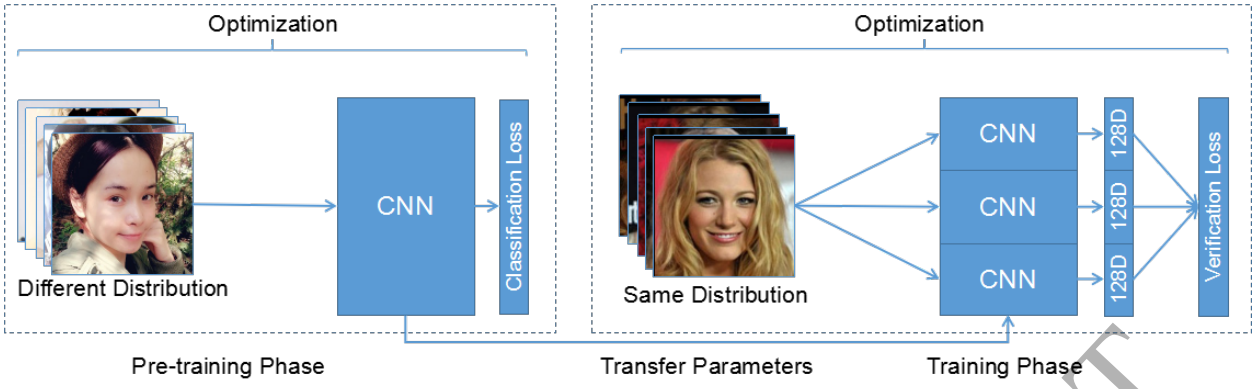


Figure 1: The framework of training.

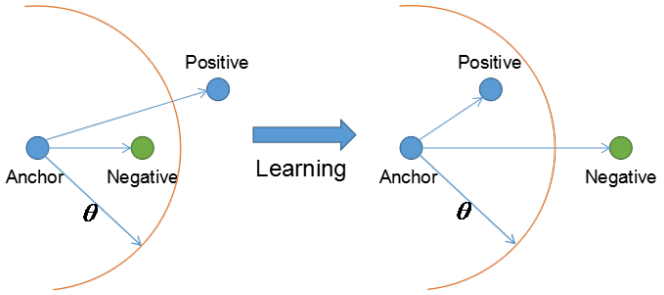


Figure 2: Face verification loss. The anchor(blue) and the positive(blue) have the same identity. The anchor(blue) and the negative(green) have different identity. The face verification loss try to minimize the distance between same identity and maximize the distance between different identity, and consider the face verification threshold θ in the same time.

identities, and we use the *softmax log-loss* function as optimization objective which is common for classification problem. [26, 18] employ the preceding method to obtain a discriminative feature extractor for face recognition. But here the feature extractor is not our target. After learning from labelled data with different distribution, we transfer the weights of the pre-trained model to the model in training phase and initialize the model with these weights instead of random sampling. Later, instead of refining last layer, we refine all weights by an end-to-end framework based on face verification loss.

Face Verification Loss

We employ an end-to-end learning in the training phase, which follows the approach of FaceNet[23]. The face verification loss directly reflects what we want to achieve in face verification and open-set face identification problems.

The CNN model $f_\theta(x)$ maps an image x into a feature space of d-dimensional hypersphere. In the testing phase of face verification, the distance between face pair is compared with a precomputed threshold θ. If $dis < \theta$, the face pair is regarded as from the same person, otherwise, from different person. Here we want to ensure that the distance from an image x_i^a (anchor) of a specific person to all other images x_i^p (positive) of the same person is less than θ. Meanwhile, the distances from an image x_i^a (anchor) to all other images x_i^n (negative) of different persons are larger than θ. This is visualized in Figure 2. So for a perfect face verification model, the following inequality holds,

$$\begin{aligned} \left\|f\left(x_i^a\right) - f\left(x_i^p\right)\right\|_2^2 &< \theta - \alpha/2, \\ \left\|f\left(x_i^a\right) - f\left(x_i^n\right)\right\|_2^2 &> \theta + \alpha/2, \\ \forall\left(x_i^a, x_i^p, x_i^n\right) &\in T \end{aligned} \tag{1}$$

where α is the margin. T is all possible triplets $\left(x_i^a, x_i^p, x_i^n\right)$ that can be generated from the datasets with x_i^a and x_i^p from the same person, x_i^a and x_i^n from different persons. By using constraints in Eq. (1), we can derive Eq. (1) in the paper of FaceNet [23] simply, but not opposite. It means our constraints are stricter than that of FaceNet by considering the face verification threshold θ.

By combining the constraints in Eq. (1), our loss function is

$$\begin{aligned} L = \sum_{i=1}^{N} &\left|\left\|f\left(x_i^a\right) - f\left(x_i^p\right)\right\|_2^2 - (\theta - \alpha/2)\right|_+ \\ &+\lambda\left|(\theta + \alpha/2) - \left\|f\left(x_i^a\right) - f\left(x_i^n\right)\right\|_2^2\right|_+ \end{aligned} \tag{2}$$

where N is the number of triplets in T. α is set to 0.2 in all our experiments. λ is the weight of two different errors. In practical applications of face verification, we want high True Accept Rate at extreme low False Accept Rate. So we should use large λ to punish the error of the second term. But in LFW, both errors are equal. $\lambda = 1$ is used in this paper. θ is a hyper-parameter, which is set to 0.8.

It is easy for randomly generated triplets to satisfy the constraints of Eq. (1), especially the second constraint as randomly picked two persons are less likely to be similar in appearance. So it is crucial to select the triplets that violate the constraints in Eq. (1). We use a similar strategy in FaceNet.

Every mini-batch contains 30 persons and 10 images for each person. We add additional 60 images of other persons in each mini-batch. For every image x_i^a in the mini-batch, we choose all possible images of the same person x_i^p. Meanwhile, negative image x_i^n is selected randomly which fulfils

$$\gamma\theta < \left\|f\left(x_i^a\right) - f\left(x_i^n\right)\right\|_2^2 < \theta + \alpha/2 \tag{3}$$

type	output size	depth	#1 × 1	#3 × 3 reduce	#3 × 3	double #3 × 3 reduce	double #3 × 3	Pool +proj
conv1(7 × 7/2)	112 × 112 × 64	1						
max pool(3 × 3/2)	56 × 56 × 64	0						
conv2(3 × 3/1)	56 × 56 × 192	1		64	192			
max pool(3 × 3/2)	28 × 28 × 192	0						
inception (3a)	28 × 28 × 256	3	64	64	64	64	96	avg + 32
inception (3b)	28 × 28 × 320	3	64	64	96	64	96	avg + 64
inception (3c)	28 × 28 × 576	3	0	128	160	64	96	max + pass
inception (4a)	14 × 14 × 576	3	224	64	96	96	128	avg + 128
inception (4b)	14 × 14 × 832	3	0	96	128	96	128	max + pass
inception (5a)	7 × 7 × 576	3	176	96	160	80	112	avg + 128
inception (5b)	7 × 7 × 576	3	176	96	160	96	112	max + 128
avg pool	1 × 1 × 576	0						
fully connected	1 × 1 × 128	0						
L2 normalization	1 × 1 × 128	0						

Table 1: CNN model used in this paper, which is simplified from inception-v2

where γ controls the difficulty of negative image. Too hard or too easy is not beneficial for convergence. We choose 0.8 for γ and the generated negative exemplars are called *semi-hard*.

Deep Convolutional Neural Network

In this paper, we use a CNN model called inception-v2 [9] which appends batch normalization layer after convolutional layer in the GoogleNet [28] for accelerating training. In order to achieve faster prediction speed, we simplify inception-v2 by removing inception (4b), (4c), (4d) and reducing the number of feature maps in inception (4e), (5a), (5b). Please see Table 1 for more details.

For the embedding model, every face image is mapped to a L_2 normalized feature vector in 128D space. For the pre-training model, we append fully connected layer and softmax loss layer after the average pooling layer. Layers before the average pooling layer are exactly the same between embedding model and pre-training model, so weights can be transferred between these two models.

In all our experiments, we train the CNN using Stochastic Gradient Descent(SGD) with standard back propagation[21]. In pre-training phase, we start with a learning rate of 0.2 and decay half for every 5 epochs. The pre-training models are trained on one GPU (GTX TITAN X) for 25 epochs and spend 89.4 hours. In training phase, we start with a learning rate of 0.04 and decay half for every 10 epochs. The models are trained on twos GPUs (GTX TITAN X) for 20 epochs and spend 18.6 hours.

Datasets and Evaluation

Public Datasets

Labeled Faces in the Wild(LFW)[8] is the most widely used benchmark for face verification. It contains 5,749 celebrities and 13,233 images which are collected from the web. We follow the Unrestricted, Labeled Outside Data Protocol and report the mean classification accuracy as well as the standard error of the mean.

CASIA-WebFace[31] is one of the largest public datasets for face recognition. It contains 10575 actors/actress and 494,414 images which are crawled from the IMDB movie website.

VGGFace[18] is another large public dataset which contains 2,622 celebrities and 2.6M images. The label of VGGFace is quite noisy because of the semi-automatic dataset collection method. We only use the good part of the images in the final model which contains 982,803 images. However, only 845,878 ones (VGGFace-Good) can be downloaded because of the broken link.

The preceding three datasets are all celebrities' faces and collected from the web. In another word, they are nearly the same in face distribution. We combine CASIA-WebFace and VGGFace-Good together, which contains nearly 1.3M images in total, and use it for the training phase.

TaoMM Dataset

TaoMM Dataset is crawled from a website of fashion model platform [1]. There are a lot of photos in the album of fashion models. Most photos contain only one face of the fashion model. So there is no labelling effort at all.

TaoMM Dataset is a large dataset for face recognition, and we will make it freely available to the research community. It contains 37,511 fashion models and 3.2M images which include many kinds of variations in face, *e.g.* pose, occlusion, hair style, make-up and expression. But it contains only young, beautiful Chinese women, so it is quite different from LFW in the face distribution. Figure 3 shows some example images. Simply ignoring the difference and throwing the TaoMM dataset into training data will lead to a reduction in face recognition performance.

[1] https://mm.taobao.com

(a) TaoMM

(b) LFW

Figure 3: Example images from TaoMM and LFW dataset. The face distributions of two datasets are quite different.

Experiments

Implementation Details

Our implementation is based on the popular MXNet[2] framework. All our experiments are carried on NVIDIA GTX TITAN X GPU with 12GB on-board memory, one for the pre-training phase and two for the training phase. Two GPUs are required because of the large mini-batch size 360 and the complexity of the CNN model. The code will be released public on GitHub later.

Faces are detected using the method described in [15]. If face alignment is required, face is aligned to the canonical position by three points, left eye, right eye and the center of mouth. We don't align face in the training phase (similar to [18, 23]), instead we employ an aggressive data augmentation process. We use the data augmentation method provided by the ImageRecordIter of MXNet. Detailed parameters are listed as follows. The specific definition for each parameter can be found in the official document of MXNet[3].

- $rand\_crop = True$
- $rand\_mirror = True$
- $max\_rotate\_angle = 20$
- $max\_random\_scale = 1.1$
- $min\_random\_scale = 0.9$
- $random\_h, random\_l, random\_s = 20$
- $max\_aspect\_ratio = 0.1$
- $max\_shear\_ratio = 0.1$

[2]https://github.com/dmlc/mxnet

[3]https://mxnet.apache.org/api/python/io/io.html?highlight=imagereco#mxnet.io.ImageRecordIter

The $f_\theta(x)$ outputs $D = 128$ feature vectors. Given a face image x, ten $224 * 224$ pixel patches are cropped from the four corners and the center with horizontal flip (similar to [18]), and feature vectors from these patches are L2 normalised after concatenation. We can crop one, two or five patches, which is a trade off between accuracy and prediction speed.

We measure the similarity between two images through the simple cosine similarity without any feature reduction method *e.g.* PCA.

Component analysis

This section evaluates the effect of different options of the system on the LFW benchmark. Table 2 shows the results.

Face Verification Loss: Face verification threshold θ is incorporated into FaceNet triplet loss function explicitly. By employing more appropriate loss function, we reduce the LFW error rate by 26.9% which can be seen from Table 2 rows 1 and 2.

Domain Adaptation: As we can see from Table 2 rows 2 and 3, simply combining the data of source and target domain and throwing into the model achieves worse performance than the model only trained by the data of target domain because of the different distributions between source and target domain. In this paper, we apply a simple domain adaptation technique to transfer the knowledge in the source domain to the target domain. Here both source domain and target domain contains millions of images. It is intuitive to believe that domain adaptation won't make any difference as we have millions labelled images of the target domain which is large enough to train a CNN model from scratch. As can be seen from Table 2 rows 2 and 4, we can still reduce the LFW error rate by 27.4% with domain adaptation. Please note that the models in Table 2 rows 1, 2 and 3 are all pre-trained with VGGFace dataset, so the improvement is not due to pre-training but domain adaptation.

Test Alignment: As can be seen from Table 2 rows 4 and 6, rows 5 and 7, using alignment on test images does improve the performance. But the improvement is so small that can be omitted. This result is quite different from FaceNet, 99.63% with test alignment and 98.87% without test alignment. When comparing performance without test alignment, we achieve better than FaceNet even if FaceNet is trained with a much larger dataset (200M images). We believe that it is the contribution of aggressive data augmentation. It brings three advantages. Firstly, it saves the time of face alignment in the prediction phase. Secondly, face alignment leads to a loss of identity information. Lastly, it is hard to align face in some extreme cases, *e.g.* profile face.

Prediction Speed

In the practical application of face recognition, prediction speed as well as accuracy will be taken into consideration especially for online (real time) application. As we can see from Table 3, there is a trade off between prediction speed and accuracy, higher accuracy lower prediction speed. But we achieve 99.33% in the online mode with 55.6 samples/sec.

No.	Face Verification Loss	Domain Adaptation	TaoMM in Training Data	Patches	Test Alignment	LFW Accuracy(%)
1	No	No	No	1	Yes	98.40
2	Yes	No	No	1	Yes	98.83
3	Yes	No	Yes	1	Yes	98.48
4	Yes	Yes	No	1	Yes	99.15
5	Yes	Yes	No	10	Yes	99.33
6	Yes	Yes	No	1	No	99.08
7	Yes	Yes	No	10	No	99.28

Table 2: Performance evaluation on LFW with different options.

Patches	Offline Mode (samples/sec)	Online Mode (samples/sec)	LFW Accuracy(%)
1	839.7	129.5	99.15
2	419.8	119.3	99.22
10	84.0	55.6	99.33

Table 3: Prediction speed of our single model with different patches. Time is measured on the server with GTX TITAN X and Intel i7-5930K. We only includes feature extraction phase after face alignment. Here offline mode (or batch mode) means all face images are already prepared in the prediction phase and processed by GPU in a large mini-batch size *e.g*. 128. While online mode (or real time mode) means face image comes one by one in the prediction phase and processed by GPU in a small mini-batch size which contains 1 to 10 patches of the face image. Due to the characteristics of GPU parallelism, prediction is faster in offline mode than online mode.

#Models #Patches	Prediction Speed (samples/sec)
1/1	129.5
1/2	119.3
1/10	55.6
2/1	64.75
10/1	12.95

Table 4: Prediction speed of our CNN model with different models and patches. Time is measured on the server with GTX TITAN X and Intel i7-5930K. We only includes feature extraction phase after face alignment. Here prediction speed is measured in the online mode which is more close to practical application of face recognition. Due to the characteristics of GPU parallelism, 1 model and 10 patches is 4.3 times as fast as 10 models and 1 patch.

Lots of methods achieve higher accuracy by an ensemble of tens of models[14, 27, 25, 26], at the cost of slowing down prediction speed ten times. In this paper, we use multi-patches with single model instead of multi-models with single patch. As can be seen from Table 4, multi-patches with single model is much faster than multi-models with single patch in prediction speed.

Comparison with the state-of-the-art

Table 5 shows the performance of our model and the state-of-the-arts on LFW. Both accuracy and standard error of the mean are reported. The standard errors are omitted if not provided. In consideration of both training and prediction efficiency, we use multi-patches with single model instead of multi-models with single patch. It can be observed that we achieve comparable results to the state of the art with only single model. When compared with single model and single patch, we achieve similar results as BaiduFace which is the best in the Table 5. We achieve 99.28% without face alignment which is better than FaceNet [23] 98.87% under the same circumstance.

Conclusion

In this paper, we crawl a million scale face dataset called TaoMM whose distribution is different from LFW. By employing a simple domain adaptation technique, we improve the LFW accuracy even with a million scale target domain dataset. By incorporating face verification threshold θ into FaceNet triplet loss explicitly, we reduce the LFW error rate by 26.9%. Finally, We achieve 99.33% on the LFW benchmark with only single CNN model and similar performance even without face alignment by applying aggressive data augmentation. When compared without face alignment, we achieve 99.28% which is better than FaceNet 98.87%, even if FaceNet uses a much larger dataset with 200M images, about 44 times of ours.

Further work will focus on applying more complex domain adaptation technique to fully exploit the knowledge in the source domain to help improving the performance of target domain. We will also look into the effect of large λ in Eq. (2) when we pursue high True Accept Rate at extreme low False Accept Rate in face verification.

References

[1] X. Cao, D. Wipf, F. Wen, G. Duan, and J. Sun. A practical transfer learning algorithm for face verification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3208–3215, 2013.

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

[3] Q. Chen, J. Huang, R. Feris, L. M. Brown, J. Dong, and S. Yan. Deep domain adaptation for describing people based on fine-grained clothing attributes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5315–5324, 2015.

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

[5] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014.

Method	LFW Accuracy(%)	Training Data	#Models #Patches
BaiduFace[14]	99.77 ± 0.06	1.2M	70/1
	99.13	1.2M	1/1
FaceNet[23]	99.63 ± 0.09	200M	1/1
FaceNet[23] (No Alignment)	98.87 ± 0.15	200M	1/1
DeepID2+[27]	99.47 ± 0.12	0.29M	25/1
Ours	99.33 ± 0.10	1.3+3.2M	1/10
	99.15 ± 0.12	1.3+3.2M	1/1
Ours (No Alignment)	99.28 ± 0.10	1.3+3.2M	1/10
	99.08 ± 0.12	1.3+3.2M	1/1
CenterLoss[30]	99.28	0.7M	1/2
Human[12]	99.20	N/A	N/A
DeepID2[25]	99.15 ± 0.13	0.20M	175/1
MMDFR[4]	99.02 ± 0.19	0.5M	8/1
VGGFace[18]	98.95	2.6M	1/30
DeepID[26]	97.45 ± 0.26	0.20M	100/1
DeepFace[29]	97.35 ± 0.25	4.4M	8/1

Table 5: Results on LFW compared with other state-of-the-arts. In this paper, we use 1.3M images from the same face distribution and 3.2M images from different face distribution.

[6] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010.

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

[8] 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.

[9] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[10] M. Kan, J. Wu, S. Shan, and X. Chen. Domain adaptation for face recognition: Targetize source domain bridged by common subspace. *International journal of computer vision*, 109(1-2):94–109, 2014.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[12] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 365–372. IEEE, 2009.

[13] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[14] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang. Targeting ultimate accuracy: Face recognition via deep embedding. *arXiv preprint arXiv:1506.07310*, 2015.

[15] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *European Conference on Computer Vision*, pages 720–735. Springer, 2014.

[16] J. Ni, Q. Qiu, and R. Chellappa. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 692–699, 2013.

[17] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014.

[18] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, volume 1, page 6, 2015.

[19] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine*, 32(3):53–69, 2015.

[20] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa. Domain adaptive dictionary learning. In *European Conference on Computer Vision*, pages 631–645. Springer, 2012.

[21] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

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

[24] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa. Generalized domain-adaptive dictionaries. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 361–368, 2013.

[25] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, pages 1988–1996, 2014.

[26] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1891–1898, 2014.

[27] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2892–2900, 2015.

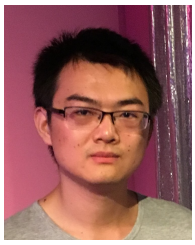
[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[29] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[30] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, pages 499–515. Springer, 2016.

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

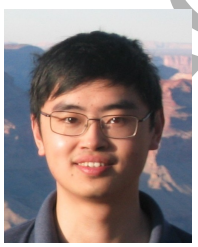
[32] E. Zhou, Z. Cao, and Q. Yin. Naive-deep face recognition: Touching the limit of lfw benchmark or not? *arXiv preprint arXiv:1501.04690*, 2015.



Ge Wen received the B.S. degree in Computer Science from Zhejiang University of China in 2014. He is currently pursuing the Ph.D. degree at the State Key Laboratory of CAD&CG, Zhejiang University. His research interests include computer vision and machine learning.

Huaguan Chen is a Software Engineer in the YITU-tech at Shanghai, China. He received the Master degree in computer science from Zhejiang University in 2017. His research interests include computer vision and natural language processing.

Deng Cai is a Professor in the State Key Lab of CAD&CG, College of Computer Science at Zhejiang University, China. He received the Ph.D. degree in computer science from University of Illinois at Urbana Champaign in 2009. His research interests include machine learning, data mining, computer vision and information retrieval.



Xiaofei He received the BS degree in Computer Science from Zhejiang University, China, in 2000 and the Ph.D. degree in Computer Science from the University of Chicago, in 2005. He is a Professor in the State Key Lab of CAD&CG at Zhejiang University, China. Prior to joining Zhejiang University, he was a Research Scientist at Yahoo! Research Labs, Burbank, CA. His research interests include machine learning, information retrieval, and computer vision.