Improved Face Verification with Simple Weighted Feature Combination

Xinyu Zhang(✉), Jiang Zhu, and Mingyu You

College of Electronics and Information Engineering, Tongji University,
4800 Cao'an Highway, Shanghai 201804, People's Republic of China
{1510464,zhujiang,myyou}@tongji.edu.cn

Abstract. Since the appearance of deep learning, face verification (FV) has made great progress with large scale datasets, well-designed networks, new loss functions, fusion of models and metric learning methods. However, incorporating all these methods obviously takes a lot of time both at training and testing stages. In this paper, we just select training images randomly without any clean and alignment procedure. Then we propose a simple weighted average method which combines features of the last two layers with different weights on the modified VGGNet, named as *CB-VGG*. It is significantly reducing the complexity of time that one model can be treated as two models. LMNN is used as a post-processing procedure to improve the discrimination of the combined features. Our experiments show relatively competitive results on LFW, CFP, and CACD datasets.

Keywords: Face verification · Deep learning
Weighted average method · LMNN metric learning

1 Introduction

Face verification (FV), whose protocol is to classify whether a pair of faces belongs to the same person or not, is one of the most challenging face tasks with difficulty of variations on pose, age, occlusion and so on [1].

In recent years, the state-of-the-art methods based on deep learning achieved the great success on FV [2–5]. One of the most important ingredients for this success is large scale facial public datasets [6] with identity labels. However, there are often noise contaminated since images are crawled from the Internet. Some methods use various measures [4,7] to clean images, while others produce multiple patches [8–12] and synthesized images [13]. It requires a lot of extra time and human resources. Since excessive noise data usually depress the classification performance, in this paper, we only select partial training data of MsCeleb [6] randomly without any cleaning and alignment methods in order to simplify pre-processing process.

J. Yang et al. (Eds.): CCCV 2017, Part II, CCIS 772, pp. 16–28, 2017.
https://doi.org/10.1007/978-981-10-7302-1_2

Some methods obtain better performance [8–11] along with concatenating features from multiple models. It needs additional time to acquire multiple models and evaluate models with high-dimensional features. Here, we propose to use both of the last two layers of a single model. Unlike concatenation, we use weighted average of the two layers without dimension increase. It can be seen that we only train one model but use it as two models by fusion of two layers, which effectively reduces the time at train and test stages.

Besides, many other strategies are effective for better performance, such as carefully-designed models [4,7] and center loss [14]. While joint Bayesian [15] has been highly successful as metric learning in many methods [9–11], Large Margin Nearest Neighbor (LMNN) [16] is used to improve accuracy. LMNN [16] is first introduced to train a matrix that maintains a large distance between imposters and constrains the k-nearest neighbors belonging to the same class. In this research, we use LMNN to further improve the discrimination of features.

Despite its simple process, our method achieves competitive results on Labeled Faces in the Wild(LFW) [17], Celebrities in Frontal-Profile in the Wild(CFP) [18] and Cross-Age Celebrity Dataset(CACD) [19]. We summarize the contributions of this paper as follows:

1. We simplify the pre-processing procedure only by selecting partial training data randomly. No any other clean measures are used to carefully select the label-corrected images.
2. We propose a new method that only requires to train one model but uses the last two layers of features via computing their weighted average, which improves performance without much time consumption.
3. We use LMNN as metric learning to strengthen the discriminative power of deeply features.
4. The proposed method performs excellent and significant results on all the three datasets of LFW, CFP and CACD.

2 Related Work

Various of methods exploited deep networks to achieve remarkable results in FV. We analyse the most critical aspects.

Data Preparation. Large scale public facial images are introduced to encourage the development of face recognition [2–5], such as CASIA-Webface [20] and MsCeleb [6]. In addition, data augmentation like multiple patches [8–12] and synthesized images [13] can make the system be more robust to various of face variation. Others like 2D/3D alignment [2,14], RGB/Gray channels [8,9] and rotated poses [13] can also contribute to performance.

Elaborately Designed Networks. Inspired by "very deep" networks, VGGNet is widely used to FV [2,13]. After that, many blocks are added to networks [4,7,11], like inception [21] and residual [22]. Besides, local connected layer(LC) [8,9,14], L2 norm [4], DeepVisage(feature normalization) [7] are designed.

Fusion of Models. Many methods show that model fusion provides additional boosts to feature presentation. DeepID series [8–11] train at least 25 models on different facial patches which are complementary with each other.

Metric Learning. Several methods use metric learning after extracting features from CNN. Principal component analysis(PCA) is mainly used [8–11] which learns a mapping matrix to reduce dimensions. Joint Bayesian is proved to be an effective way [8–11,20]. Baidu [12] uses triplets which shortens the distance of intra-class and enlarges that of inter-class.

Various Loss Functions. Softmax loss [2,12] is largely used for classification and has been used for feature embedding by removing the last classification layer. Contrastive loss [3,9–11] and triplet loss [2,5,12] are introduced to enhance intra-class binding. Then, center loss [14] is proposed to learn a center for each class.

3 The Proposed Method

We use VGGNet [2] as our baseline and modify it with a convolution-branch block(CB), called *CB-VGG*. What's more, it brilliantly exploits weighted average of the last two layers, followed by LMNN [16] before computing the cosine similarity as the verification score of a pair images. Next, we describe all the details.

3.1 Data Pre-processing

Our pre-processing process is extremely simple which saves a lot of time.

Training Data: We use MsCeleb [6] as our training data. Different from other approaches [4,7,14] cleaning the training data, we only select 29,731 identifies randomly except target datasets. For simplification, we only use Normalized Pixel Difference(NPD) [23] detector without any 2D [2,7] or 3D alignment [3]. It is inherited by [2] which found that alignment on test images instead of training data brought the best benefits. The work in [2] uses large amounts of images and the alignment step on test stage spends plenty of time. In contrast, we crop 4 corner with 1 center patches and randomly flip these images for data augmentation to promote the performance of model.

Test Data: The process is similar as training stage. Differently, LFW [17] and CACD [19] are all cropped 10 times consisting of 4 corners and 1 center with their flips. For CFP [18], we only use 1 crop without flip due to Frontal-Profile pairs.

3.2 CB-VGG Architecture

Baseline: Our baseline, VGGNet [2] comprises 11 sections with more than one linear and non-linear blocks like ReLU and max pooling. The first 8 sections

use convolution as linear operator while the following 2 use fully connected layers(FC). The last FC layer is only used for classification.

CB-VGG Model: We make a small modification on the baseline model. Similar to [14], the 4th and 5th pooling units are concatenated together as the input of the 1st FC layer. In order to match the size of the two layers, we apply another convolutional operation after the 4th pooling layer, which is named as convolution-branch block(CB). It is critical for accurate face representation, because some information would be lost after successive down sampling. In this case, we can make full use of both global-abstract semantic information and high-resolution feature. Moreover, we simply use softmax loss instead of combining with other loss [14] or normalization [4,7]. The details of the model are given in Fig. 1.

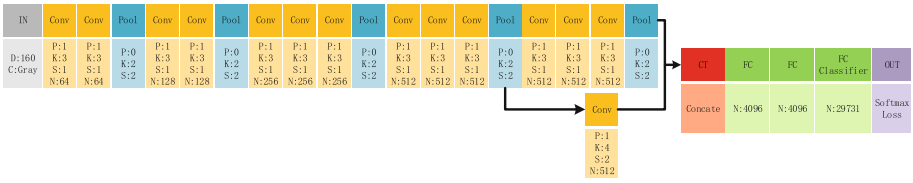

Fig. 1. Structure of CB-VGG model. **IN** is the input images. **Conv** is convolution layer. **P** represents padding. **K** is kernel size. **N** is the output number. **Pool** indicates max pooling unit. **CT** is concatenation layer which concatenates the 4th pooling(after one extra convolution) and the 5th pooling layers. **FC** and **FC Classifier** indicate fully connected layer and classification layer respectively. **OUT** is the softmax loss function.

3.3 Weighted Average Features

The deep features are taken from the last FC layer almost in all other methods. What's different in our method is that we extract features on both the last two FC layers. For an image x, we represent its feature descriptors using the 1st and the 2nd FC layer as f^x_{1st} and f^x_{2nd}. Then, the final feature descriptor f^x_{final} is expressed by taking weighted average of these two features:

$$\mathbf{f}^x_{final} = \alpha \times \frac{1}{C}\sum_{i=1}^{C}\mathbf{f}^{x_i}_{1st} + \beta \times \frac{1}{C}\sum_{i=1}^{C}\mathbf{f}^{x_i}_{2nd}. \tag{1}$$

α and β are the weights of the 1st and the 2nd FC layer's descriptors respectively which act on every elements. In order to discriminate the role of the two layers, we impose restrictions on the weights with $\beta = 1 - \alpha$. C is the number of crops which is 1 in CFP and 10 in LFW and CACD.

Model fusion is always effective. Sun et al. [9–11] concatenate 25 model's features to improve performance. Baidu [12] uses 10 embedding models to obtain high performance. However, our method only trains one model but uses two layers to extract features, which is equal to training two models but saves a lot of time and boosts the performance at the same time.

3.4 LMNN Metric Learning

Large Margin Nearest Neighbor(LMNN) [16] learns a Mahanalobis distance matrix which is good for decreasing the distance of k target neighbors, while enlarging the distance of different k nearest points. In FV task, some distances of positive pairs are larger than negative pairs, causing a bad effect on thresholds selection. Based on LMNN, we can pull the same faces closer and repel different faces further. The problem can be described as follows. i and j are same pairs, yet l is different from them. ξ_{ijl} is a slack variable. $\mathbf{M}$ is the transformable matrix.

$$min \sum_{i,j\to C_i} (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) + \sum_{i,j\to C_i, l\to C_l} \xi_{ijl}. \tag{2}$$

$$s.t. \forall_{y_i=y_j\neq y_l} (\mathbf{x}_i - \mathbf{x}_l)^T \mathbf{M}(\mathbf{x}_i - \mathbf{x}_l) - (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) \geq 1 - \xi_{ijl}, \quad \xi_{ijl} \geq 0, \mathbf{M} \succeq 0. \tag{3}$$

FaceNet [5] uses triplet loss similar to LMNN based on triplets consisting of a positive pair and a negative identity. However, it is not easy to generate triplets and [5] uses 200 M images for training. Instead, we use LMNN for employing idea of triplet but avoiding triplets selection.

4 Experiments

In this section, we first introduce the details of the training stage. Then we extract features of the last two FC layers(FC6, FC7) and perform LMNN to calculate the cosine similarity in FV. In order to verify the effectiveness of our method, we evaluate it on three different datasets. All experiments we carry out are on Caffe [24].

4.1 Details of Training Stage

In this subsection, we describe the details of model training. We randomly select 2.29 M images of 29,731 identities from MsCeleb dataset [6] without including the test identities. For simplification, **no other cleaning and alignment strategies** are used to pick up images. We train our model using 95% images (2.17 M images) and other 5% images (120 K) are used for monitoring and validating the loss. Our CNN model **only uses identity information** to optimize softmax loss. **No other normalization methods** [4,7] are used.

The baseline model and our CB-VGG model are trained from scratch with the same hyper parameters. We use stochastic gradient descent(SGD) method. Momentum and weight decay are set to 0.9 and $5e^{-4}$. We begin the training with a learning rate of 0.01 and decrease it by 10 times every 100 K iterations. The training batch size is 120. Images are resized into 170×170, and then cropped into 160×160 of 4 corners and 1 center. We also apply randomly horizontal flipping to the images for data augmentation.

4.2 Results of Test Datasets

Evaluation is performed on the commonly used LFW dataset [17], frontal-profile CFP dataset [18] and cross-age CACD dataset [19]. Like training stage, we also only use NPD to detect images without any other extra alignment process. The parameters of α and β for FC6/FC7 are 0.15/0.85 respectively. The number of target neighbors k is set to 3 in LMNN.

LFW Dataset: LFW dataset [17] is one of the most popular datasets and the performance on it is close to saturation. It consists of 13,233 face images of 5,749 different identities. We evaluate our model following the most permissive protocol: unrestricted with labeled outside data. The FV task requires evaluating on 6,000 pairs in 10 folds. Each fold contains half of the genuine pairs and half of the impostor pairs.

We compare the network trained on Baseline and CB-VGG. For both FC6 and FC7, we extract features of 10 patches separately. Table 1 shows that CB-VGG is lightly better than Baseline. More over, with the weighted average method and LMNN metric learning, accuracy is further improved. The result verifies the effectiveness of weighted average of the last two layers which can be thought as fusion of two models but only trained once.

Table 1. Accuracy on LFW (%). Y is Yes (Use), N is Not (Not use)

No	Method	FC6	FC7	LMNN	Acc%
1	Baseline	Y	N	N	98.78
2	Baseline	N	Y	N	98.85
3	Baseline	Y	Y	N	98.95
4	CB-VGG	Y	N	N	98.78
5	CB-VGG	N	Y	N	98.90
6	CB-VGG	Y	Y	N	**99.02**
7	CB-VGG LMNN	Y	Y	Y	**99.07**

We also compare our method with other state-of-the-art methods. The results are given in Table 2. We observe that our method achieves relatively significant accuracy (99.07%) with relative less images and simple methods.

CFP Dataset: CFP [18] is one of the large-pose face datasets which is composed of 10 frontal and 4 profile images of each 500 individuals. Unlike LFW, CFP has two FV experiments: frontal-frontal (FF) and frontal-profile (FP). Both contains 10 folds, each with 350 same pairs and 350 different pairs.

We also compare the Baseline and CB-VGG model. However, features are extracted from both FC6 and FC7 with only 1 center crop. PCA is applied for reducing dimensions of fusional features from $4096d$ to $300d$. Table 3 shows that CB-VGG with fusional features and LMNN is more robust to pose transformation.

Following the protocal in CFP [18], we also report the EER(Equal Error Rate) and AUC(Area under the curve) values on averages of 10 splits. Table 4 and Fig. 2 provide the results and the ROC curve[1] of ours along with the state-of-the-art methods.

Table 2. Performance comparison of state-of-the-art on LFW (%). Y is Yes (Use), N is Not (Not use)

Method	Train-Images	Clean	Align	Norm	Models	Other-Loss	Metric-Learning	Acc%
DeepFace [3]	4.4 M, 4 K	N	Y	–	3	Y	Kafang	97.35
VGG Face [2]	2.6 M, 2.6 K	N	Y	Y	1	Y	Triplet	98.95
Baidu [12]	1.2 M, 1.8 K	Y	Y	–	1	Y	Triplet	99.13
Center Loss [14]	0.7 M, 17.2 K	–	Y	–	1	Y	–	99.28
DeepID2+ [9]	0.29 M, 12 K	–	Y	–	25	Y	Joint Bayesian	99.47
DeepID3 [11]	0.29 M, 12 K	–	Y	–	25	Y	Joint Bayesian	99.53
L2-softmax [4]	3.7 M, 58.2 K	Y	Y	Y	1	N	–	99.60
DeepVisage [7]	4.48 M, 62 K	Y	Y	Y	1	N	N	99.62
FaceNet [5]	200 M, 8000 K	–	Y	Y	1	Y	–	99.63
CB-VGG LMNN	2.29 M, 29.7 K	N	N	N	1	N	LMNN	**99.07**

Table 3. Accuracy on CFP (%). Y is Yes (Use), N is Not (Not use)

No.	Method	FC6	FC7	LMNN	Acc%	
					FF	FP
1	Baseline	Y	N	N	98.21	90.36
2	Baseline	N	Y	N	98.34	91.73
3	Baseline	Y	Y	N	98.43	91.99
4	CB-VGG	Y	N	N	98.11	90.41
5	CB-VGG	N	Y	N	98.39	91.94
6	CB-VGG	Y	Y	N	**98.46**	**92.06**
7	CB-VGG LMNN	Y	Y	Y	**98.59**	**92.43**

We observe that our method achieves the competitive results to the best accuracy on both FF and FP. For FF ours result is slightly lower than FV-DCNN+pool5 [25] but enjoys being simple. FV-DCNN+pool5 [25] learns the Gaussian mixture model and performs Fisher vector encoding after extracting features from models. Besides, they also use PCA, Joint Bayesian and scores fusion with pool5. For FP, we achieve the first place if not competing p-CNN [1]. p-CNN(pose-directed multi-task CNN) specially tackles pose variation by separating all poses into several groups and then jointly learn identity and PIE(pose,

[1] Because of lacking of data, we couldn't report the work of Sankarana *et al.* [26].

Table 4. Performance comparison of state-of-the-art on CFP (%). **ACC** is Accuracy. **EER** is Equal Error Rate. **AUC** is Area Under the Curve

Method	FF (Frontal-Frontal)			FP (Frontal-Profile)		
	ACC	EER	AUC	ACC	EER	AUC
Human [18]	96.24 ± 0.67	5.34 ± 1.79	98.19 ± 1.13	94.57 ± 1.10	5.02 ± 1.07	98.92 ± 0.46
Deep features [18]	96.40 ± 0.69	3.48 ± 0.67	99.43 ± 0.31	84.91 ± 1.82	14.97 ± 1.98	93.00 ± 1.55
Sankarana *et al.* [26]	96.93 ± 0.61	2.51 ± 0.81	99.68 ± 0.16	89.17 ± 2.35	8.85 ± 0.99	97.00 ± 0.53
FV-DCNN [25]	98.41 ± 0.45	1.54 ± 0.43	99.89 ± 0.06	91.97 ± 1.70	8.00 ± 1.68	97.70 ± 0.82
FV-DCNN+pool5 [25]	$\mathbf{98.67 \pm 0.36}$	$\mathbf{1.40 \pm 0.37}$	$\mathbf{99.90 \pm 0.09}$	89.83 ± 1.88	10.40 ± 1.85	96.37 ± 0.97
p-CNN [1]	97.79 ± 0.40	2.48 ± 0.07	99.71 ± 0.02	$\mathbf{94.39 \pm 1.17}$	$\mathbf{5.94 \pm 0.11}$	$\mathbf{98.36 \pm 0.05}$
CB-VGG LMNN	$\mathbf{98.59 \pm 0.53}$	$\mathbf{1.47 \pm 0.55}$	$\mathbf{99.90 \pm 0.07}$	$\mathbf{92.43 \pm 0.75}$	$\mathbf{8.18 \pm 0.82}$	$\mathbf{97.03 \pm 0.75}$

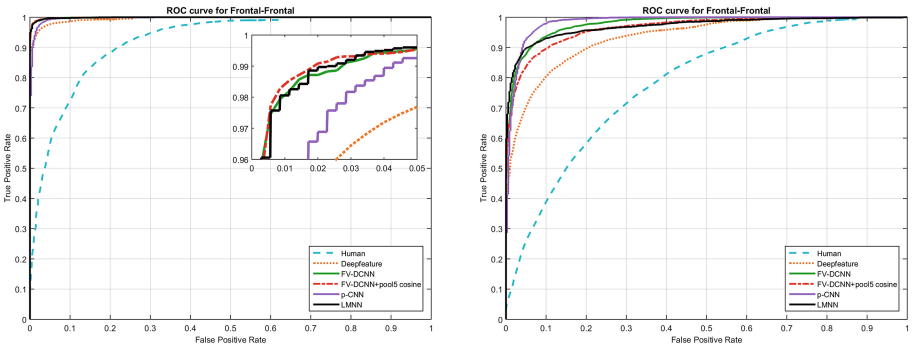

Fig. 2. ROC curve of CFP dataset. (a) is the protocol of Frontal-Frontal, while (b) is the protocol of Frontal-Profile. We only compare deep learning methods without conventional methods.

illumination and expression) for each group. Although p-CNN has great advantage on pose, it performs relatively worse than us on FF.

CACD Dataset: CACD [19] is a large cross-age celebrity dataset which consists of 163,446 images of 2,000 celebrities with age ranging from 16 to 62. For verification task, 2,000 positive image pairs and 2,000 negative pairs are selected to form CACD-VS and divided by 10 folds.

Similar to the previous process, we report comparison of baseline with ours and provide the results of other methods on CACD-VS. The results are shown in Tables 5 and 6.

MFM-CNN [28] and DeepVisage [7] are well designed that the former uses maxout activation to separate noisy signals with informative signals and the latter uses feature normalization to restrict features keeping equal contribution to cost function. LF-CNNs [27] aim at learning age-invariant features via combining latent factor layer. Note that we don't use any complex blocks or finetune measures to fit age variation.

In summary, we only use VGGNet with modification of combining FC6 with FC7 and LMNN metric learning. All the training images are randomly selected

Table 5. Accuracy on CACD (%). Y is Yes (Use), N is Not (Not use)

No	Method	FC6	FC7	LMNN	Acc%
1	Baseline	Y	N	N	97.68
2	Baseline	N	Y	N	97.38
3	Baseline	Y	Y	N	97.70
4	CB-VGG	Y	N	N	97.38
5	CB-VGG	N	Y	N	97.58
6	CB-VGG	Y	Y	N	**97.78**
7	CB-VGG LMNN	Y	Y	Y	**97.88**

Table 6. Performance comparison of state-of-the-art on CACD (%). Y is Yes (Use).

Method	Well-designed block	Aiming at age	Acc%
Human, Avg [19]	–	–	85.70
Human, Voting [19]	–	–	94.20
VGG Face [2]	–	–	96.00
LF-CNNs [27]	–	Y	98.50
MFM-CNN [28]	maxout activation	–	98.55
DeepVisage [7]	feature normalization	–	99.13
CB-VGG LMNN	–	–	**97.88**

without any clean measures. Experiments show that our method is definitely robust to pose and age variation on FV task. It means our method is robust to both frontal and pose-variable faces and weighted features of the last two layers really learns mutually complementary information. LMNN is a fast metric learning which is really able to improve discrimination.

4.3 Analysis and Discussion

For highlighting the influences of methods, we perform further analysis on (i) number and data augmentation of training datasets; (ii) different settings of weights; (iii) some examples which are corrected by our methods. All the investigations are conducted on LFW dataset.

First, we study the influence from the quantity of training data. Table 7 presents the results[2] which we observe that: (i) the larger number of images creates the better CNN performance, although the $100K$ identities decrease the results due to the large dirty images; (ii) Multi-crop augmentation is really helpful for performance, since it is equal to add more pure images for per identities

[2] Due to the restrictions of memory and time, we don't conduct an experiment on 100 K dataset with multiple crop. The number of images is less than original images which is due to failing detection.

to increase number of training images. In this paper, we don't use any selection or alignment for training dataset. Instead, multiple crop for data augmentation helps us to extract features with location and pose invariance.

Table 7. Analysis of the influences from number and data augmentation of training dataset. ∼ is the approximate number which is obtained after detection.

Number	Single-crop (%)	Multi-crop (%)
1.62 M, ∼20 K	97.00	98.75
2.29 M, ∼30 K	98.83	99.02
7.66 M, ∼100 K	98.63	–

Next, we analyse the various weights combination of the last two layers. Figure 3 shows that the weights selection of the features is important. Since FC7 layer extracts more information than FC6, the weights of the former is must be larger than the latter. Otherwise, the performance will be pull down by the weak feature.

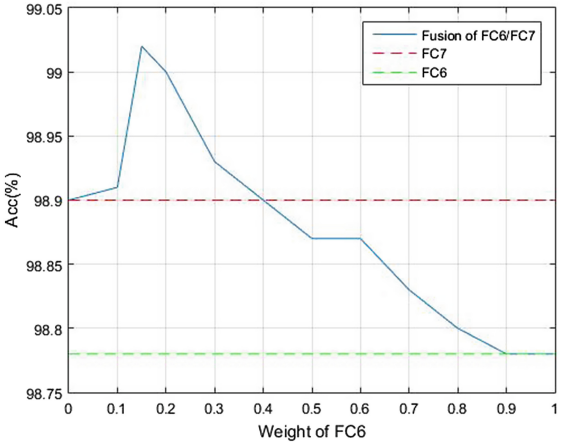

Fig. 3. Analysis of the influences from the weights of the last two layers. (Color figure online)

Finally, we provide some examples which are corrected by our methods. In Fig. 4, four misclassified pairs are presented that (a) two pairs are false rejected images and (b) another two are false accepted images. These pairs are erroneously classified by CB-CNN model with only FC7 layer. When we use weighted average features of last two layers, pairs marked with green colored rectangles are corrected. It can be seen that more discriminative features are extracted

using simple weighted combination which are more robust to pose, illumination, occlusion and so on. Then, when we further use LMNN, more difficult pairs can be corrected, such as the pair with red rectangle. That's to say that our method really have ability to improve the representation of features.

(a) Examples of false rejected images.

(b) Examples of false accepted images.

Fig. 4. Some misclassified pair images of LFW dataset by CB-CNN model with only FC7 layer. Pair images with green rectangle is corrected via weighted average features of FC6 and FC7 layers. Then the pair with red rectangle is further corrected using LMNN. (a) and (b) is false rejected images and false accepted images respectively.

5 Conclusions

In this paper, we not only use extremely simple pre-processing procedure without clean or alignment, but also propose a single model named as CB-VGG. Taking care of time and accuracy, we present a simple weighted average on the last two FC layers instead of fusing several models. This can be seen as fusion of two models but only training once. After that, we use LMNN metric learning as post-processing process. Combining all these methods, we achieve competitive results and perform relatively robust to pose and age variations. These results successfully show that it may not be necessary to use complex process of selecting images or training multiple models to boost performance. Single models can be fully used to develop a good result and the time can be saved enormously. In the future, we will explore whether the cleaned images are really important for the performance.

Acknowledgements. This work was supported by Natural Science Foundation of Shanghai (No. 17ZR1431500).

References

1. Yin, X., Liu, X.: Multi-Task Convolutional Neural Network for Face Recognition. arXiv preprint arXiv:1702.04710 (2017)
2. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: Proceedings of the British Machine Vision Conference, pp 41.1-41.12 (2015)
3. Taigman, Y., Yang, M., Ranzato, M.A., et al.: Deepface: closing the gap to human-level performance in face verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1701–1708 (2014)
4. Ranjan, R., Castillo, C.D., Chellappa, R.: L2-constrained Softmax Loss for Discriminative Face Verification. arXiv preprint arXiv:1703.09507 (2017)
5. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 815–823 (2015)
6. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: MS-Celeb-1M: a dataset and benchmark for large-scale face recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016, Part III. LNCS, vol. 9907, pp. 87–102. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_6
7. Hasnat, A., Bohné, J., Gentric, S., et al.: DeepVisage: making face recognition simple yet with powerful generalization skills. arXiv preprint arXiv:1703.08388 (2017)
8. Sun, Y., Wang, X., Tang, X.: Deep learning face representation from predicting 10,000 classes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1891–1898 (2014)
9. Sun, Y., Chen, Y., Wang, X., et al.: Deep learning face representation by joint identification-verification. In: Advances in neural information processing systems, pp 1988–1996 (2014)
10. Sun, Y., Wang, X., Tang, X.: Deeply learned face representations are sparse, selective, and robust. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2892–2900 (2015)
11. Sun, Y., Liang, D., Wang, X., et al.: Deepid3: Face recognition with very deep neural networks. arXiv preprint arXiv:1502.00873 (2015)
12. Liu, J., Deng, Y., Bai, T., et al.: Targeting ultimate accuracy: Face recognition via deep embedding. arXiv preprint arXiv:1506.07310 (2015)
13. Masi, I., Trần, A.T., Hassner, T., Leksut, J.T., Medioni, G.: Do we really need to collect millions of faces for effective face recognition? In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016, Part V. LNCS, vol. 9909, pp. 579–596. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_35
14. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016, Part VII. LNCS, vol. 9911, pp. 499–515. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_31
15. Chen, D., Cao, X., Wang, L., Wen, F., Sun, J.: Bayesian face revisited: a joint formulation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 566–579. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33712-3_41

16. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. J. Mach. Learn. Res. **10**(1), 207–244 (2009)
17. Huang, G.B., Ramesh, M., Berg, T., et al.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, University of Massachusetts (2007)
18. Sengupta S, Chen J C, Castillo C, et al.: Frontal to profile face verification in the wild. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), pp 1–9. IEEE (2016)
19. Chen, B.C., Chen, C.S., Hsu, W.H.: Face recognition and retrieval using cross-age reference coding with cross-age celebrity dataset. IEEE Trans. Multimedia **17**(6), 804–815 (2015)
20. Yi, D., Lei, Z., Liao, S., et al.: Learning face representation from scratch. arXiv preprint arXiv:1411.7923 (2014)
21. Szegedy, C., Liu, W., Jia, Y., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1–9 (2015)
22. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 770–778 (2016)
23. Liao, S., Jain, A.K., Li, S.Z.: A fast and accurate unconstrained face detector. IEEE Trans. Pattern Anal. Mach. Intell. **38**(2), 211–223 (2016)
24. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)
25. Chen, J.C., Zheng, J., Patel, V.M., et al.: Fisher vector encoded deep convolutional features for unconstrained face verification. In: 2016 IEEE International Conference on Image Processing (ICIP), pp 2981–2985. IEEE (2016)
26. Sankaranarayanan, S., Alavi, A., Castillo, C.D., et al.: Triplet probabilistic embedding for face verification and clustering. In: 2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS), pp 1–8. IEEE (2016)
27. Wen, Y., Li, Z., Qiao, Y.: Latent factor guided convolutional neural networks for age-invariant face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4893–4901 (2016)
28. Wu, X., He, R., Sun, Z.: A lightened CNN for deep face representation. In: 2015 IEEE Conference on IEEE Computer Vision and Pattern Recognition (CVPR) (2015)