2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Learning Face Recognition from Limited Training Data using Deep Neural Networks

Xi Peng[1]
Department of Computer Science
Rutgers University
Piscataway, New Jersey 08854

Nalini Ratha
IBM Thomas J. Watson
Research Center
Yorktown Heights, New York 10598

Sharathchandra Pankanti
IBM Thomas J. Watson
Research Center
Yorktown Heights, New York 10598

***Abstract*—Often deep learning methods are associated with huge amounts of training data. The deeper the network gets, the larger is the need for training data. A large amount of labeled data helps the network learn about the variations it needs to handle in the prediction stage. It is not easy for everyone to get access to huge amounts of labeled data leaving a few to have the luxury to design very deep networks. In this paper, we propose to flatten the disparity by using the modeling methods to minimize the need for huge amounts of data for training a deep network. Using face recognition as an example, we demonstrate how limited labeled data can be leveraged to obtain near state of the art performance with generalization capability across multiple databases. In addition, we show that the normalization in the overall network can improve the speed and resource requirement for the prediction/inferencing stage.**

I. Introduction

Face recognition, in essential, targets to learn a mapping from face image to a feature space where distance directly correspond to a measure of face similarity. Because of many advantages with faces being available readily, and how humans can recognize faces without any effort has attracted many researchers to build a successful automatic face recognition over several decades. Overall recognition accuracy involved in face recognition with many novel underlying technologies has improved many folds over last decade [5]. A traditional face recognition involves face detection from a given image followed by landmark detection and/or image characterization through complex filters in a high dimensional space such as Gabor, LBP, SHIFT, HOG etc. The representation of a face image can then be compared with previously stored image representation using well-known distance metrics such as L1-norm, L2-norm or cosine.

It has been seen that face recognition accuracy can suffer when there is a significant difference in pose, occlusion, (low) resolution, illumination, aging, expression and decorations [33]. In order to handle these variations, traditional face recognition methods have been based on either global (appearance based) or local (feature based) techniques. In the global method, the extracted features are based on the information embedded in the whole face [27]. Belhumeur et al. [2] extended this PCA based technique was extended and improved by using linear discriminant analysis. Global methods are mostly limited to handling face images taken in highly controlled environments with extremely cooperative subjects. In contrast, local-based algorithms are based on encoding details of subregions of the face including eyes, nose, mouth, and lips [26] and adopt a fusion strategy to combine the outputs from different face subregions. Pose and illumination changes are handled using the local orientations and related image features such as SIFT [14], the phase of Gabor jets [29], and gradient pyramids [13] and 3-D alignment. Sparse representation [30] and local binary patterns [26] have also been proposed to address face recognition problem.

In last couple of years, several research groups [16], [20], [25] have shown great success on face recognition using deep convolutional neural networks (CNN). Training these networks with limited dataset is expected result in suboptimal (e.g., overfitting) solution due to extensive variations in appearances of individuals due to occlusion, illumination, resolution, expression, decoration, and head pose. As a result, training these deep neural networks usually requires a large number of labeled face images. Besides, many approaches rely on external pre-processing systems for face detection and alignment before feeding images into the recognition network. This fragmentation not only limits the testing efficiency, but also impairs the recognition accuracy due to easy failures of external systems.

To reuse available labeled data, often the deep learning community used data augmentation methods [21] by adding noise or rotating the image or looking at subwindows from the image to enhance the training performance. However, for face recognition these data augmentation methods have minimal impact as often the preprocessing steps normalize these variations. While public face datasets are useful in training and testing algorithms, data sets such as LFW are often based on celebrity face images or dataset like NIST MEDS has sampling bias as it has been obtained from a smaller population. Thus currently, an unbiased training data for face recognition with large numbers of persons and large numbers of samples/person are not available to researchers.

With these constraints in mind, we propose to use a model based deep learning network for face recognition. By using a face model, we can use known normalization methods before the recognition stage. Additionally, having the model normalization and recognition trained simultaneously through a deep learning framework helps in faster response for prediction over

[1]This work was completed while Xi Peng was at IBM Research

978-1-5090-4847-2/16/$31.00 ©2016 IEEE

using a normalization outside the deep learning framework.

The paper is organized as follows. We summarize recent results of deep learning based face recognition in Section 2. Our proposed approach is described in Section 3. Results from our proposed approach are presented in Section 4.

The prediction accuracy of our approach is comparable with state-of-the-art, *e.g., 96.6% vs. 97.3%* on public benchmarks, while it needs significantly fewer training images than previous approaches, *e.g., 0.4 million vs. 200 million*. Moreover, our end-to-end framework guarantees highly efficient performance in testing phase, *e.g., 0.027s vs. 2.4s per frame*, compared to the methods that rely on external pre-processing systems.

II. Related Works

In recent years, Convolutional Neural Networks (CNNs) have shown great success on the task of face recognition. They outperform other approaches that use hand-crafted features (such as SIFT, HOG, LBP) with a substantial margin.

One such representative work is presented by Taigman *et al.* in DeepFace [25]. They use 4.4 million examples of 4,000 identities to train a Siamese architecture [3], where two identical CNNs with shared weights are applied to a pair of faces to obtain a discriminative similarity metric. There are 11 layers in each CNN which perform convolutional, max pooling and fully connected operations. The goal of the training is to maximize the distance between impostor pairs (different identity) and minimize the distance between genuine pairs (same identity). In addition to using a very large number of training images, they employ an external module to apply 3D face alignment in the pre-processing phase, which is proved to be crucial to increase the training speed and improve the generalization ability.

Similar to DeepFace that relies on external module for 3D face alignment, multiple other CNN based methods [16], [23], [31] have been proposed recently to apply 2D face alignment in the pre-processing phase. They usually first locate the face as well as facial landmarks and then apply scale and in-plane rotation to align all face images. The aligned faces have the same geometric measurement such as the same inter-ocular or eye-month distance. Although they show extremely good performance on face recognition accuracy, their approach suffers from obvious limitations. The reliance on external pre-processing modules which is not only less efficient compared to an end-to-end framework but also has inevitable performance degradation when external modules fail in challenging conditions.

To avoid reliance on the complex external face alignment module, Schroff *et al.* use a massive dataset to train very deep CNNs for face recognition which has a similar 22-layer architecture to [24]. This dataset contains 200 million examples of 8 million identities. Different from the pair-wise metric learning used in Siamese architecture, they propose a triplet loss for the network training. More specifically, they pick out a pair of genuine pair of examples $(face-a, face-b)$ as well as a third impostor example $(face-c)$ that is closed to the genuine pair from the training batch. The goal is to make $face-a$ closer to $face-b$ than $face-c$. This approach achieves impressive recognition accuracy on in-the-wild benchmarks. However, like other large-scale face image sets with annotations, their training dataset is private. It is infeasible to reproduce their network using much smaller public datasets.

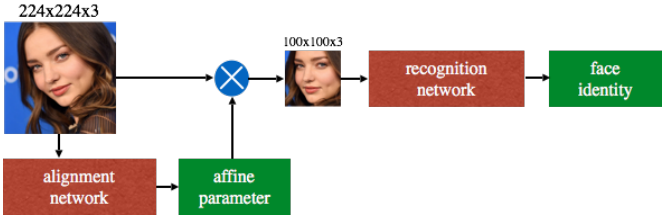

Fig. 1. Overview of the proposed network for face recognition. The alignment network takes the image as input to predict affine parameters for automatic face alignment such as translation, scaling, rotation, and shear. Then the aligned face is fed into the recognition network for identity prediction.

III. Our Approach

In this work we present an end-to-end CNN based approach for face recognition from either a single photograph or a set of tracked faces in a video. We will introduce details of our approach in this section: Firstly, we propose to apply spatial transformer learning for automatic face alignment. Secondly, we carefully design a recognition network for efficient and robust face recognition. Thirdly, we present the details of our specially designed training procedure which are crucial for efficient convergence using limited training data. An overview of our approach is shown in Figure 1.

A. Alignment Network

Face images on individuals in wild conditions suffer extensive variations due to expression, decoration, resolution, pose, and illumination [6], [12], [17]. The previous deep networks-based recognition approaches either attempt to learn the invariance from large amount labeled samples [20], [23] or employ external face normalization modules to reduce the variation space [25], [28], [31]. The former approaches usually require millions of training images, which is typically infeasible for most researchers since only a small number of training images are available to them. The latter approaches are prone to failure on outliers as the performance of external alignment modules degrades significantly in challenging conditions [18], [19]. Besides, they also suffer from slower testing speed since a sequence of pre-processing (such as face detection, landmark localization and face alignment) undertaken in the external module could potentially be inefficient.

To address the aforementioned limitations, we propose to apply spatial transformer learning [10] for end-to-end face recognition. The goal of the alignment network is to learn a pixel-wise transformation:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (1)$$

where (x, y) are the coordinates of the input image and (x', y') are the coordinates of the aligned image. $\{\theta_{11}, \cdots, \theta_{23}\}$ are 6 parameters of the affine transformation which model translation, scaling, in-plane rotation, and generic shear. By using the sub-differentiable sampling mechanism [10], the loss gradients can flow back through the alignment network and the parameters of the network can be efficiently learned using the standard backpropagation. Since the spatial transformer learning is semi-supervised, no additional annotation of affine transformation is required. Instead, the alignment network parameters can be learned simultaneously along with the recognition network.

Fig. 2. The architecture of our alignment network.

The architecture of the alignment network is shown in Figure 2. It takes $224 \times 224 \times 3$ image as input and outputs 6 parameters of the affine transformation. The filter sizes in convolutional layers are varying from 7×7, 5×5 to 3×3 to capture coarse-to-fine features in cascade. We apply 2×2 max pooling after each convolutional layer and a fixed 2-pixel stride in all layers to reduce the number of parameters. We empirically concluded that 256 neurons followed by 64 neurons in the fully connected layers can provide acceptable results to predict 6 affine parameters.

There are multiple merits of the proposed spatial transformer learning approach. First, it is practical and convenient to train the alignment network in a semi-supervised manner together with the recognition network, which only requires face identity labels. Second, the alignment network can significantly reduce the extensive variations in wild conditions, which facilitates the recognition task with respect to both training efficiency and testing generalization. Third, the alignment network and recognition network are jointly optimized in the same task, which alleviates the performance degradation issue of external alignment module in challenging conditions. Most importantly, the face alignment and recognition are accomplished in an end-to-end framework, which is not only convenient to train but also very efficient in test. Our approach takes only 0.027s to process a single image, while former methods that rely on external alignment modules usually take more than 2s to process the same image.

B. Recognition Network

The goal of the recognition network is to seek a non-linear mapping from the aligned face image to a sparse feature vector for identity representation. On the one hand, as the training data is limited, the training is prone to overfitting with local optimum if the CNN architecture goes too deep. On the other hand, to guarantee satisfying prediction accuracy, the network need to be complicated enough to encode all variations in representation subspace. With these considerations, we incorporate the inception module [20] in our design to make the network go wider instead of deeper.

The architecture of the recognition network is shown in Figure 3. In summary, it contains 5 functional modules: 2 convolutional modules and 3 inception modules. Each convolutional module consists of 2 convolutional layers in cascade to extract low-level features. 3×3 filters and stride 2 are used in all convolutional layers. We apply ReLU [15] and Local Response Normalization (LRN) after each convolutional layer to improve the training efficiency. Similarly, each inception modules consists of 2 inception layers in cascade to generate high-level combinations of features extracted in earlier layers. As we illustrate in Figure 3, the inception module applies a series of convolutional filters in parallel, *e.g.* 1×1, 3×3 and 5×5, to fully explore the feature combinations for discriminative representation. Additionally, 1×1 convolutional filters are used in parallel forks to reduce the total number of network parameters.

The recognition network takes $100 \times 100 \times 3$ aligned image as the input and outputs a $256d$ feature vector as identity representation. We set the aligned image to $100 \times 100 \times 3$ instead of the original $256 \times 256 \times 3$ as it achieves a good balance between the recognition accuracy and efficiency. Both softmax loss and contrastive loss [11] are employed in network training as shown in Figure 4. To calculate softmax loss, we concatenate a fully connected layer of 10575 neurons after the $256d$ identity feature, which corresponds to the 10575 identities in the training dataset. We compute the softmax loss in a multi-class classification framework:

$$\ell_s = -\frac{1}{N}\sum_{n=1}^{N} log(\hat{f}_n, l_n), \tag{2}$$

where n counts images in a mini batch, $\hat{f}_n$ is the identity prediction, and l_n is the one-hot label vector which has a single 1 for the selected identity and all 0s for others.

To calculate the contrastive loss, we generate both genuine (same identity) and impostor (different identities) pairs in a mini batch. Generally speaking, it is more difficult for a Siamese network [3] to correctly classify a genuine pair than an impostor pair due to the extensive intra-class variations especially in unconstrained settings. Therefore, for effective training, we use all the genuine pairs but pick out only the top-k closest impostor pairs in a mini batch to compute the contrastive loss in a binary classification framework:

$$\ell_c = \frac{1}{2M}\sum_{m=1}^{M} (y_m)d^2 + (1 - y_m)max(D - d, 0)^2, \tag{3}$$

where m counts image pairs in a mini batch. $y_m = \{0, 1\}$ is the binary ground truth: 0 and 1 denote genuine and impostor pair respectively. $d = \|\hat{f}_m^l - \hat{f}_m^r\|_2$, where $\hat{f}_m^l$ and $\hat{f}_m^r$ are the $256d$ representation vectors of the image pair. D is the margin between the largest genuine distance and the smallest impostor distance.

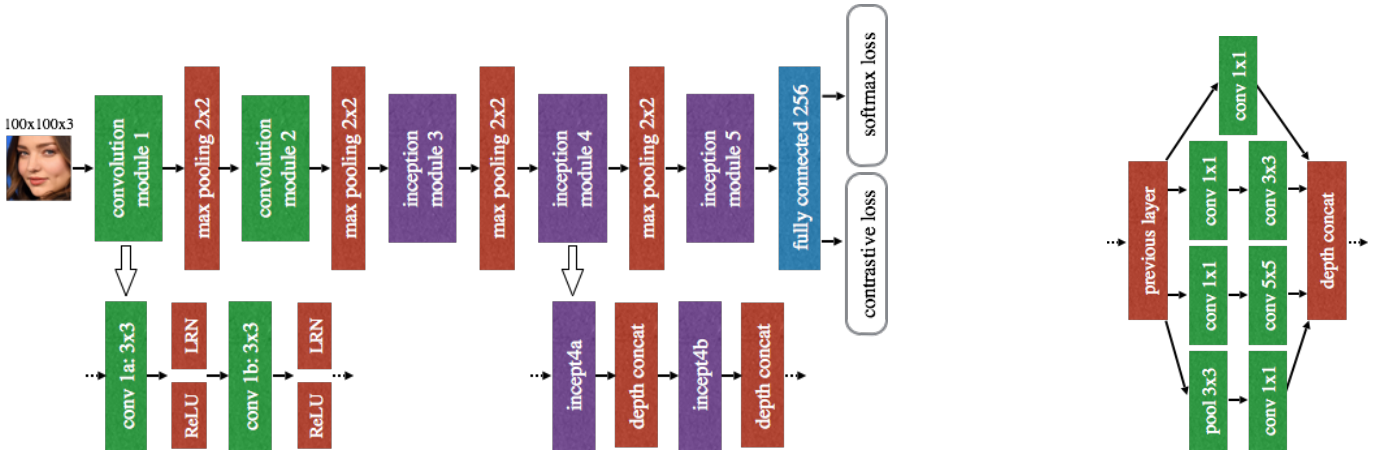

Fig. 3. (a) The recognition network; and (b) the inception module. The 1×1 convolutional layers are used to cut down the network complexity.

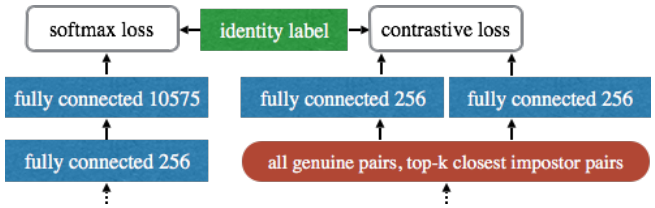

Fig. 4. Both softmax and contrastive loss are used in the network training.

C. Training Strategy

It is difficult to train the entire network from scratch when only a small number of training images are available. To address this challenge, we design a two-step procedure to guarantee the efficiency of the network training. The first step is to pre-train the recognition network using aligned images. For every training image, we perform a standard 2D face alignment to alleviate extensive variations in training set. The process is illustrated in Figure 5, which includes face detection, landmark localization, face alignment, and horizontal flipping transformations for data augmentation. The aligned images are $100 \times 100 \times 3$. The pre-training is performed for 100 epochs.

Fig. 5. An illustration of the 2D face alignment process. It includes face detection, landmark localization, face alignment and horizontal mirroring. All training images are aligned to have the same distance between the center of the eyes and the center of the mouth.

The second step is to add the alignment network and fine-tune the entire network using unaligned images. The unaligned images are $224 \times 224 \times 3$. For every training image, we only randomly perform scale, rotation, and horizontal flipping for data augmentation. The fine-tuning is performed for another 50 epochs to get network parameters to converge.

It is worth mentioning that although we perform the two-step training on a small training set, our approach is capable of end-to-end training when sufficient training images are available. Besides, deploying the network is computationally efficient as no additional external face alignment module is required.

IV. Experiments

In this section, we first introduce datasets used in our experiments and present the implementation details of our approach. Then we compare different distance metrics and different data pre-processing methods. Finally, we compare the proposed approach with state-of-the-art and present discussions.

A. Datasets and Implementation Details

In order to best evaluate the performance of our approach, we used CASIA-WebFace dataset [31] for training, and used LFW [8] and MEDS [9] datasets for testing. Note that there is no identity overlap between the training and testing datasets.

The CASIA-WebFace dataset is currently the largest publicly available face dataset with identity annotation. We used this dataset to train both the alignment and the recognition networks in our approach. It contains 494,414 face images of 10,575 different celebrities collected from IMDb website. These images were collected under unconstrained conditions.

The first testing dataset is Labeled Faces in the Wild Dataset (LFW), which is the standard benchmark for face recognition and verification. This dataset contains 13,233 face images of 5,749 identities. These images exhibit multiple challenges related to face decoration, pose, expression, illumination, and partial occlusion. The second testing dataset is Multiple Encounter Dataset (MEDS), which is public dataset used to evaluate real-world face recognition systems. This dataset contains 1,309 face images of 518 identities.

Our implementation is based on Theano [1]. We used the Stochastic Gradient Descent with a fixed momentum

coefficient of 0.9 for network optimization. The learning rate was initialized to 0.1 and decayed 0.02 after every 20 training epochs. The weights of both networks were randomly initialized using a Gaussian distribution with 0 mean and 0.01 standard deviation. To regularize the trained model, we applied both weight decay with a rate of 0.001 and dropout after fully connected layers with a rate of 0.4.

In the pre-train step, we employed [22] for face detection and 5-point landmark detection. The distance between the center of the eyes and the center of the mouth was normalized to25 pixels by scaling and in-plane rotation for 2D face alignment. We flipped the image left to right to double the number of training images for data augment. In the fine-tuning step, we also performed the following data augmentation to facilitate the network training. We first randomly scaled, rotated, and flipped the images. The scale ratio was set to 0.8 to 1.0 and the rotation range was set to $-15°$ to $15°$. We subsequently re-scaled the transformed images to 256×256 pixels. Then we randomly cropped out 224×224 patches from the re-scaled images.

B. Comparison of Different Distance Metric

We used the face verification protocol for the performance evaluation [8]. The network was used to extract two feature vectors from a pair of testing images. The distance between the two faces was then computed based on the pair of feature vectors. We trained a linear SVM classifier which takes the distance as input and outputs 0 or 1 to indicate genuine or impostor pair. We found in experiments that different distance metric can significantly affect the verification accuracy.

The testing accuracy with respect to different training epochs is demonstrated in Figure 6. We investigate four widely used distance metrics: L_1, L_2, K^2 and $Cosine$. Our experimental results demonstrate that $Cosine$ distance outperforms other metrics by a substantial margin. A possible explanation is the feature vector output by the network is highly sparse [32], while $Cosine$ distance is more robust to scale and rotation than the other metrics on sparse representation. We used $Cosine$ distance metric in the following experiments.

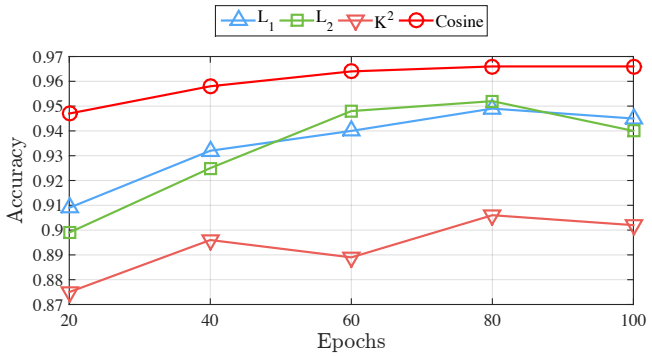

Fig. 6. Accuracy comparison of different distance metrics for face verification.

TABLE I
COMPARISON BETWEEN EXTERNAL FACE ALIGNMENT AND THE PROPOSET ALIGNMENT NETWORK.

	Training	Testing		Test speed
	CASIA	LFW	MEDS	per image
External alignment	99.3%	90.3%	92.5%	2400ms
Alignment network	98.6%	96.6%	98.4%	27ms

Fig. 7. Examples of frontalized faces using 3D face alignment [7]. The second row shows fake frontalized faces caused by incorrect facial landmark localization in challenging cases. These failures significantly deteriorated the testing accuracy.

C. Comparison of Different Data Pre-processing

To validate the proposed alignment network, we implemented a variant of our approach, which employed an external face alignment module instead of the spatial transformer network for data pre-processing. To train the variant, the face frontalization method proposed in [7] was employed to generate frontalized face images. The 3D face alignment procedure is similar to the data pre-processing used in DeepFace [25]. Figure 7 shows examples of frontalized faces. The variant network was also trained on CASIA-WebFace for the same 100 epochs.

The comparison of external face alignment and the proposed alignment network is shown in Table I. We can see that using external face alignment achieved higher accuracy on training data compared to that using the proposed alignment network: *e.g.* 99.3% *v.s.* 98.6%. However, the external alignment method suffered significant performance degradation on the testing data. The reason is the external 3D face alignment is prone to generate fake results once the 68-landmark localization fails in challenging cases as shown in Figure 7. Generally speaking, the proposed alignment network has better generalization ability as the alignment and recognition tasks are jointly optimized. The proposed method achieved 96.6% and 98.4% accuracy on LFW and MEDS datasets, respectively. Besides, the proposed approach is highly efficient in testing since the alignment and recognition are performed in the end-to-end framework obviating the need for an external processing module. It took only 27ms to process one image, which is near $100\times$ faster than using external alignment module which involves very time-consuming face detection, landmark localization, and face frontalization.

TABLE II
COMPARISON OF THE TRAINING DATASET AND TESTING ACCURACY ON LFW BETWEEN OUR APPROACH THE STATE-OF-ARTS METHODS.

	DeepFace [25]	FaceNet [20]	VggFace [16]	Ours
Training data	SFC	WebFace	VggFace	CASIA
# image	4.4M	200M	2.6M	0.4M
# subject	4K	8M	2.6K	10K
img/sub	1100	25	1000	40
Testing acc	97.3%	98.8%	98.9%	96.6%

D. Comparison with State-of-the-art

The comparisons of the training datasets and testing accuracy on LFW between our approach and state-of-the-art methods are presented in Table II. Since neither the code nor the training datasets is public for DeepFace [25], FaceNet [20] and VggFace [16], we report their performance on LFW according to their papers. We can see that our training dataset is significantly smaller than the others in terms of both number of images used or subjects involved, *e.g.* 0.4 million *vs.* 200 million images and 10,575 *v.s.* 8 million subjects. Despite this significant handicap of training database size and the number of identities involved, our system achieves comparable testing accuracy on the challenging LFW dataset, *e.g.* 96.6% *vs.* 97.3%. Besides, since both DeepFace and VggFace employ external modules for either 3D or 2D face alignment for data pre-processing, it is reasonable to expect that our approach is more efficient than theirs since we perform the alignment and recognition in an end-to-end framework.

V. CONCLUSION

In this paper, we proposed a novel deep neural network based approach that uses relatively small size training dataset for learning face recognition model that is effective under wild conditions. Our approach jointly learns the alignment and recognition task in an end-to-end framework. The experimental results demonstrate that our method overcomes the limitation of external pre-processing modules used in the previous approaches, yielding robust alignment in challenging conditions as well as extremely efficient testing.

REFERENCES

[1] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Berferon, N. Bouchard and Y. Bengio, "Theano: New Features and Speed Improvements", *NIPS*, 2012.

[2] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 711-720, Jul 1997.

[3] S. Chopra, R. Hadsell and Y. LeCun, "Learning a Similarity Metric Discriminatively, with Application to Face Verification", , IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," IEEE Computer Society Conference on Computer Vision and Pattern Recognition,2005, pp. 886-893.

[5] P. Grother and M. Ngan, "Face Recognition Vendor Test (FRVT) Performance of Face Identification Algorithms", *NIST* Interagency 2013.

[6] G. Guo and X. Wang, " study on human age estimation under facial expression changes," IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2547-2553.

[7] T. Hassner, S. Harel, E. Paz and R. Enbar, "Effective Face Frontalization in Unconstrained Images", IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[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, University of Massachusetts*, 2007.

[9] A. P. Founds, N. Orlans, W. Genevieve and C. I. Watson, "NIST Special Databse 32 - Multiple Encounter Dataset II (MEDS-II)", 2011.

[10] M. Jaderberg, K. Simonyan, A. Zisserman and K. Kavukcuoglu, "Spatial Transformer Networks", *CoRR*, 2015.

[11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding", *CoRR*, 2014.

[12] C. Li, Q. Liu, J. Liu and H. Lu, "Learning ordinal discriminative features for age estimation," IEEE Conference on Computer Vision and Pattern Recognition, pp2570–2577, 2012.

[13] H. Ling, S. Soatto, N. Ramanathan and D. W. Jacobs, "A Study of Face Recognition as People Age," IEEE International Conference on Computer Vision, 2007.

[14] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, vol. 60, pp.91-110, 2004.

[15] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines", IEEE International Conference on Machine Learning, 2010.

[16] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep Face Recognition", British Machine Vision Conference, 2015.

[17] X. Peng, J. Huang, Q. Hu, S. Zhang, A. Elgammal and D. Metaxas, "From Circle to 3-sphere: Head Pose Estimation by Instance Parameterization", Computer Vision and Image Understanding, 2015.

[18] X. Peng, S. Zhang, Y. Yu and D. Metaxas, "PIEFA: Personalized Incremental and Ensemble Face Alignment", IEEE International Conference on Computer Vision, 2015.

[19] X. Peng, Q. Hu, J. Huang and D. Metaxas, "Track Faces in Unconstrained Videos", British Machine Vision Conference, 2016.

[20] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering", IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[21] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", IEEE International Conference on Learning Representations, 2015.

[22] Y. Sun, X. Wang and X. Tang, "Deep Convolutional Network Cascade for Facial Point Detection", IEEE Conference on Computer Vision and Pattern Recognition, 2013.

[23] Y. Sun, D. Liang, X. Wang, and X. Tang, "DeepID3: Face Recognition with Very Deep Neural Networks", *CoRR*, 2015.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions", *CoRR*, abs/1409.4842, 2014.

[25] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification", IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[26] A. Timo, H. Abdenour, P. Matt, "Face description with local binary patterns: application to face recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (12) (2006) 2037-2041.

[27] M. Turk, A. Pentland, "Eigenfaces for recognition", Journal of Cognitive Neuroscience, 3 (1) (1991) 71-86.

[28] X. Wang, R. Guo, and C. Kambhamettu, "Deeply-learned feature for age estimation", IEEE Winter Conference on Applications of Computer Vision, 2015, pp. 534-541.

[29] L. Wiskott, J.-M. Fellous, N. Krger, C. von der Malsburg, "Face Recognition by Elastic Bunch Graph Matching", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no.7, pp.775-779, July, 1997.

[30] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, "Robust face recognition via sparse representation", IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 210-227.

[31] D. Yi, Z. Lei, S. Liao, and S.Z. Li, "Learning Face Representation from Scratch", *CoRR*, 2014.

[32] Y. Sun, X. Wang and X. Tang, "Deeply Learned Face Representations are Sparse, Selective, and Robust", IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2892-2900.

[33] W. Zhao, R. Chellappa, P.J. Phillips, A. Rosenfeld, "Face Recognition: A Literature Survey", ACM Computing Surveys, no.4, pp.399-458, 2003