Learning Feature Representation for Face Verification

Sangwoo Park[1], Jongmin Yu[2], and Moongu Jeon[3]
Gwangju Institute of Science and Technology
123, Cheomdangwagi-ro, Buk-gu, Gwangju, South Korea
{[1]parkswoo, [3]mgjeon}@gist.ac.kr, [2]jm.andrew.yu@gmail.com

Abstract

Previous models based on Deep Convolutional Neural Networks (DCNN) for face verification focused on learning face representations. The face features extracted from the models are applied to additional metric learning to improve a verification accuracy. The models extract high-dimensional face features to solve a multi-class classification. This results in a dependency of a model on specific training sets since a dimension of the feature should be equal to the number of subjects in a training set. In this paper, we propose a method for learning feature representations which directly determine whether two input images are identical using a single model based on DCNN and residual learning. It is possible to remove the dependency since the model doesn't learn face representations based on multi-class classification. We show that the proposed method achieves the competitive performance for face verification. We demonstrate the face verification performance of the proposed method using the test dataset of Labeled Face in the Wild dataset.

1. Introduction

For the past two decades, face verification has been actively studied and considered as one of challenging issues in the field of computer vision. Face verification is to determine whether two face images are the same or not via one-to-one matching. There are a lot of issues to overcome, such as illuminations, aging, occlusion, and facial expression, for developing superb face verification methods. These issues can significantly cause performance degradation. To address these issues, many researchers tried to find face features which were invariant to these issues by hand-crafted methods. Ojala *et al*. [18] proposed the Local Binary Pattern (LBP) which was useful for face recognition. Chen *et al*. [5] proposed the high-dimensional multi-scale LBP, which had shown a remarkable efficiency for face recognition. Gabor wavelets [17] can also encode a variety of information from face images.

In recent, Deep Convolutional Neural Networks (DCNN) [16] have achieved the dramatic performance in diverse areas such as object detection [20], object recognition [15], drowsiness detection [28] and face verification [21, 24]. It shows that the models based on DCNN can train robust and discriminative feature representations even if the models need a lot of training data. Face verification methods based on DCNN commonly can be divided into three approaches.

The first approaches are based on multi-class classification to learn face representations [6, 27]. The face features extracted from the models are applied to metric learning, such as Joint Bayesian Metric Learning [4] and Support Vector Machine [24] to estimate a similarity between an input pair. Since these approaches methodologically handle the face verification as the multi-class classification, these approaches derive high-dimensional face features from the models which a dimension of the features is equal to the number of subjects in the training set. Thus, these models have a dependency which trains the models using only a specific training set despite achieving the state-of-the-art verification accuracy. The second approaches focus on a combination of identification and verification model to obtain a more optimized model for face images [19, 22]. Although combined learning provides a more general face representation, achieving an optimized model is a still challenge task. The third approaches are to directly train a verification model using the same pairs (if two face images are identical) and different pairs (if two face images are not identical). The approaches aim to learn feature representations which determine whether the face images are identical, and the model of these approaches is independent on a training set because it doesn't need to classify a training set to multi-class.

In this paper, we propose a method based on the third approaches and use a novel model to achieve the competitive performance. Our method doesn't require additional metric

978-1-5386-2939-0/17/$31.00 ©2017 IEEE

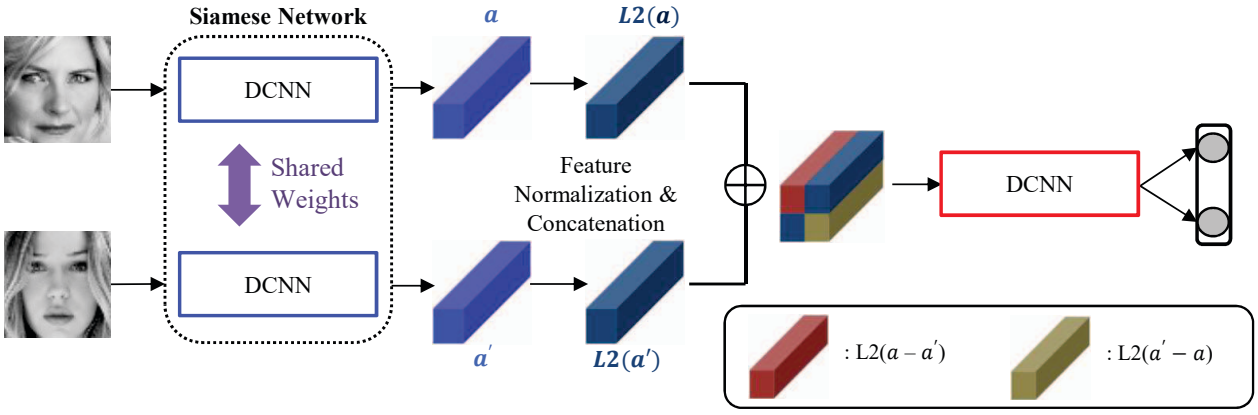

Figure 1. An overall model structure of the proposed method. The ***a*** and ***a'*** denote two extracted features from the siamese network. The L2(***a***) denotes that the vector ***a*** is L2-normalized.

learning and is independent on training datasets since the model doesn't learn face representations based on multi-class classification. We train our networks using CASIA-WebFace dataset [27] and compare the performance using the test dataset of Labeled Face in the Wild (LFW) dataset [13].

2. Related Work

In this section, we briefly introduce some methods using CASIA-WebFace dataset as a training set for face verification.

These methods are based on learning robust and discriminative face representations using DCNN models. Chen *et al*. [6] proposed a method for unconstrained face verification. Their model extracts face features based on multi-class classification, and then they apply the features to Joint Bayesian Metric Learning. Also, Chen *et al*. [7] additionally included the Fisher Vector (FV) to encode DCNN face features. Yi *et al*. [27] proposed a method which combines identification and verification model using two cost functions. Wang *et al*. [26] proposed a face search system which consists of Template Generation, Face Filtering, and Re-ranking.

The above models have something in common in that these models are trained using CASIA-WebFace dataset as the training set and extract a facial feature of high-dimensions. As a result, these methods provide comparable performance to other methods that use a private dataset. However, a limitation is that the models have a lot of parameters due to multi-class classification and the number of dimensions for the facial feature is exactly the same as the number of classes in the training set (*i.e*. 10,575). In other words, the models are extremely dependent on the training set and also are inflexible because it requires only CASIA-WebFace dataset as the training set.

3. Method

We propose a method for learning feature representations which directly determine whether two input images are identical using a single model based on DCNN and residual learning. The structure of the proposed model is inspired by the siamese network [8] and composed of a siamese network (two DCNN sharing the weights), feature normalization & concatenation layer, and a DCNN. Figure 1 shows an overall model structure of our method. First, the siamese network extracts two features for an input pair. Second, the feature normalization & concatenation layer computes differences between two features using L2-normalization and element-wise subtraction, and then the layer generates a concatenated feature. It helps to optimize the model by producing the concatenated feature, rather than a single value, to the output while maintaining properties of cos-similarity. In last, the DCNN extracts a two-dimensional vector to yield the estimation of whether the face images are identical using the concatenated feature in the early layer.

In the following subsections, we describe a preprocessing procedure of the training set, all components of the proposed model, and training methodology in details.

3.1. Preprocessing

We use CASIA-WebFace dataset for training our model. Before training the model, we convert the training dataset from RGB to grayscale and perform face detection using the OpenCV implementation of robust face detector by Viola and Jones [25]. Then, five facial landmarks in the detected image are extracted by [23] and consist of left and right eye centers, left and right mouse corners and nose tip. After extracting the points, we rotate the image by aligning two points of eyes horizontally and change the location of nose tip to center of the image. We resize resolution of the images to 100×100 pixels. We also horizontally flip each

Name	#Filters / Filter Size / Stride	Output Size	#Params
Conv1	1 / 5×5×1 / 1	96×96×32	1.6K
Pool1	1 / 2×2 / 2	48×48×32	-
Residual Block1	5 / 3×3×32 / 1 1 / 3×3×64 / 1	48×48×32 48×48×64	92K 36K
Pool2	1 / 2×2 / 2	24×24×64	-
Residual Block2	5 / 3×3×64 / 1 1 / 3×3×128 / 1	24×24×64 24×24×128	368K 148K
Pool3	1 / 2×2 / 2	12×12×128	-
Feature Concatenation	-	24×24×128	-
Residual Block3	3 / 3×3×128 / 1 1 / 3×3×256 / 1	24×24×128 24×24×256	442K 295K
Pool4	1 / 2×2 / 2	12×12×256	-
Residual Block4	3 / 3×3×256 / 1 1 / 3×3×384 / 1	12×12×256 12×12×384	1769K 885K
Pool5	1 / 2×2 / 2	6×6×384	-
Global Avg Pool	1 / 6×6 / 1	1×1×384	-
FC1	-	2	0.768K
Loss	-	2	-
Total	-	-	4037K

Table 1. The detailed architecture of the proposed model. The bold indicates layers in the siamese network. Therefore, we compute the number of parameters in the bold layers twice.

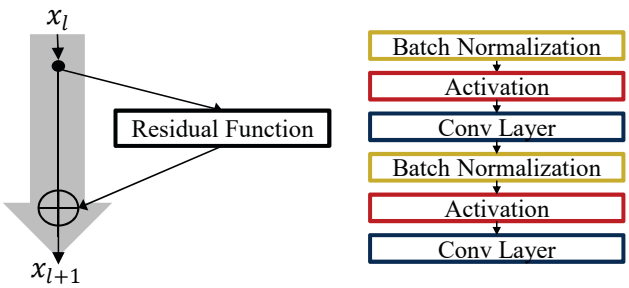

Figure 2. (Left) Residual Learning; (Right) Structure of Residual Function. The output is sum of a results of an identity mapping and residual function for the input.

image to augment the dataset. Every pixel value of the training image is scaled from 0 to 1 and then it is subtracted from the mean of scaled pixel values.

3.2. Network Architectures

Deep residual networks (Resnets) [10] with residual learning have shown to achieve remarkable performance on many datasets such as ImageNet, CIFAR, MS COCO, and PASCAL. The residual learning can optimize easily networks and improve the performance of a deeper network. Namely, a feature representation using residual learning is more robust and discriminative. Therefore, we apply the residual learning to our networks for face verification. Figure 2 shows a residual unit as suggested in [11].

We express the residual unit in a general form:

$$\mathbf{x}_{l+1} = h(\mathbf{x}_l) + \mathcal{F}(\mathbf{x}_l, W_l) \quad (1)$$

Here $\mathbf{x}_l$, $\mathbf{x}_{l+1}$, and W_l are an input, output, and set of weights of l-th layer. The function $h(\mathbf{x}_l)$ performs an identity mapping with $h(\mathbf{x}_l) = \mathbf{x}_l$ and $\mathcal{F}(\mathbf{x}_l, W_l)$ denotes a residual function, which is composed of Batch Normalization layers [14], activation functions, and convolution layers.

The proposed model includes a convolution layer, residual blocks, pooling layers, and a fully connected layer. A residual block is composed of three residual units in a DCNN of a siamese network and two residual units in a DCNN. The filter sizes of the convolution layer, residual blocks, and avg pooling layer are 5×5, 3×3, and 6×6, respectively. We use Parametric Rectified Linear Unit (PReLU) [9] as activation function, which alleviates a potential issue of Rectified Linear Unit (ReLU) which deactivates all the negative responses. Table 1 shows the detailed architecture of the proposed model.

3.3. Feature Normalization & Concatenation

The siamese network provides two features for an input pair to feature normalization & concatenation layer. We use L2-normalization and element-wise subtraction to generate a new feature for the relationship between two features. Let $\boldsymbol{a}$ represents one of the features for the pair as a vector in n-dimensional space (*i.e.* $\boldsymbol{a} \in \mathbb{R}^n$). In that case, the L2-normalized vector has a distance to 1. It means that

Method	#Net	Mean Accuracy ± SE	Protocol	Training Set
DeepFace [24]	1	95.92% ± 0.29%	unsupervised	private
DeepFace	7	97.35% ± 0.25%	unrestricted	private
DeepID2 [22]	1	95.43%	unsupervised	private
DeepID2	4	97.75%	unsupervised	private
DeepID2	25	98.97%	unsupervised	private
Yi *et al*. [27]	1	96.13% ± 0.30%	unsupervised	CASIA-Webface, public
Yi *et al*. + PCA	1	96.30% ± 0.35%	unsupervised	CASIA-Webface, public
Yi *et al*. + Joint Bayes	1	97.73% ± 0.31%	unrestricted	CASIA-Webface, public
Wang *et al*. [26]	1	96.95% ± 1.02%	unsupervised	CASIA-Webface, public
Wang *et al*.	7	97.52% ± 0.76%	unsupervised	CASIA-Webface, public
Chen *et al*. [6]	1	97.15% ± 0.7%	unsupervised	CASIA-Webface, public
Human, funneled [26]	N/A	99.20%	N/A	N/A
Ours (model A)	1	96.07% ± 0.23%	unsupervised	CASIA-Webface, public
Ours (model B)	**1**	**96.62%** ± 0.17%	**unsupervised**	CASIA-Webface, public

Table 2. The performance on the LFW dataset

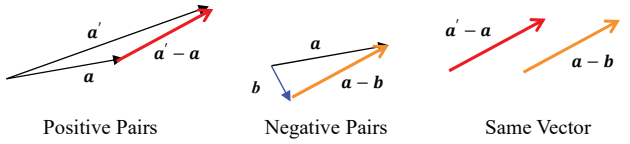

Figure 3. The ambiguity of normal element-wise subtraction. The subtraction vectors of positive and negative pair are same.

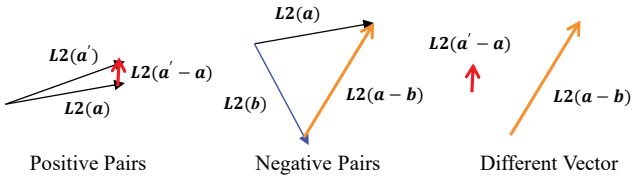

Figure 4. The efficiency of L2-normalization. The subtraction vectors of positive and negative pair are different. It is more discriminative than the normal method.

L2-normalization embeds the vector into a surface of n-dimensional hypersphere. The L2-normalization for a feature vector is defined as:

$$\|\boldsymbol{a}\|_2 = \sqrt{\Sigma_i a_i^2} \tag{2}$$

$$L2(\boldsymbol{a}) = \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|_2} \tag{3}$$

where $\|\boldsymbol{a}\|_2$ means the length of the vector (*i.e*. L2-norms). When comparing two features, element-wise subtraction of L2-normalized feature can resolve ambiguity for normal element-wise subtraction. Figure 3 and 4 show the ambiguity of the normal method and the efficiency of our method. After computing the relationship, the layer concatenates the normalized vectors as well as the L2-normalized subtraction vectors and then conveys the concatenated feature to the DCNN.

3.4. Training Details

The CASIA-WebFace used to train our model contains 494,414 face images and 10,575 subjects. We collect the same pairs as positive sets and the different pairs as negative sets for the training set. As a result of collecting, our training sets contain imbalanced datasets with many more negative than positive sets. Thus, we apply Hard Negative Mining [2] because if the model is trained using the imbalanced datasets, the performance of the model is degraded. After setting the ratio of positive and negative sets to 1:3, we train repeatedly the model whenever the negative sets reset. The negative slope value of PReLU is initialized to 0.25, following [9]. We set the weight decay of all layers to 5e-4 and the initial learning rate to 0.1, which is divided by 10 when the loss plateaus. We use a momentum of 0.9 and stochastic gradient descent [3] to update parameters. The training set randomly is divided into batch size of 180. We don't use dropout [12] to connections with fully connected layer as suggested in [14].

4. Experimental results

4.1. Labeled Face in the Wild

LFW dataset contains 13,233 face images and 5,749 subjects. LFW dataset on the standard protocol for face verification contains 6,000 pairs in total and includes 7,701 face images and 4,281 subjects. It is divided by 10 disjoint subgroups for cross-validation. Each of subgroups includes 5400 training and 600 test pairs. Performance reporting on LFW dataset follow three protocols: restricted, unrestricted and unsupervised protocol. The restricted protocol is that the experimenter should use only the given training and test pairs. In the unrestricted protocol, the experimenter can augment the training pairs by inferring the relationship between the given training pairs. In the unsupervised protocol,

the experimenter should not use LFW dataset to train their model in a supervised way. Regardless of protocol type, the test dataset is fixed, and mean accuracy and standard error of the mean should be recorded for 10 test subgroups.

4.2. Evaluation on LFW

We evaluated our model A and B on LFW dataset on the unsupervised protocol. Feature Normalization & Concatenation layer of model A provides the concatenated feature which consists of three features (two L2-normalized features and one L2-normalized subtraction feature). In order to augment feature, the concatenated feature of model B includes the L2-normalized opposite subtraction feature as well as the concatenated feature of model A. We performed the preprocessing process with the same flow as the training set. Table 2 shows the mean accuracy and standard error of the mean for the proposed model A and B using LFW dataset. The model A and B achieved the mean accuracy about 96.07% and 96.62% for 10 disjoint test subgroups. The performance of the model B is competitive on the unsupervised setting using only single model. The proposed method doesn't require additional metric learning since our model aim to act as metric learning. Note that we use only the public dataset and our model doesn't learn a face representation based on multi-classification to remove the dependency. Thus, our model can be trained to use other datasets. In addition, our model has fewer parameters than the compared models using CASIA-WebFace dataset in Table 2. We use the Google's deep learning library Tensorflow [1] to implement the proposed method.

5. Conclusions

In this paper, we have proposed a method for face verification without extra metric learning using a single model. The proposed model consists of a siamese network, feature normalization & concatenation layer, and a DCNN. Previous methods focused on learning robust face representation to apply to metric learning. The proposed method provides an efficient and robust feature representation learning which helps us to directly verify the relationship between two images. The results using LFW show that our method achieves comparable performance despite removing the constraint of the recent methods.

In future work, we plan to combine face detection and verification model based on DCNN, and then we apply the integrated system to access control in specific areas. We will also focus on developing a novel method which represents the relationship between two inputs to improve the performance.

Acknowledgments

This work was supported by the ICT R&D program of MSIP/IITP. [2014-0-00077 Development of global multi-target tracking and event prediction techniques based on real-time large-scale video analysis] and the Climate Technology Development and Application research project (K07733) through a grant provided by GIST in 2017.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] E. Ahmed, M. Jones, and T. K. Marks. An improved deep learning architecture for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3908–3916, 2015.

[3] L. Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

[4] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In *European Conference on Computer Vision*, pages 566–579. Springer, 2012.

[5] D. Chen, X. Cao, F. Wen, and J. Sun. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3025–3032, 2013.

[6] J.-C. Chen, V. M. Patel, and R. Chellappa. Unconstrained face verification using deep cnn features. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016.

[7] J.-C. Chen, J. Zheng, V. M. Patel, and R. Chellappa. Fisher vector encoded deep convolutional features for unconstrained face verification. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 2981–2985. IEEE, 2016.

[8] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016.

[12] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[13] 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.

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[15] 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.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[17] C. Liu and H. Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Transactions on Image processing*, 11(4):467–476, 2002.

[18] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.

[19] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, volume 1, page 6, 2015.

[20] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[21] 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.

[22] 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.

[23] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3476–3483, 2013.

[24] 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.

[25] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[26] D. Wang, C. Otto, and A. K. Jain. Face search at scale: 80 million gallery. *arXiv preprint arXiv:1507.07242*, 2015.

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

[28] J. Yu, S. Park, S. Lee, and M. Jeon. Representation learning, scene understanding, and feature fusion for drowsiness detection. In *Asian Conference on Computer Vision*, pages 165–177. Springer, 2016.