Deep Convolutional Neural Network using Triplets of Faces, Deep Ensemble, and Score-level Fusion for Face Recognition

Bong-Nam Kang, *Student Member, IEEE*[1], Yonghyun Kim, *Student Member, IEEE*[2], and Daijin Kim, *Member, IEEE*[2]
[1]Department of Creative IT Engineering, POSTECH, Korea
[2]Department of Computer Science and Engineering, POSTECH, Korea
{bnkang, gkyh0805, dkim}@postech.ac.kr

Abstract

This paper proposes a new face verification method that uses multiple deep convolutional neural networks (DCNNs) and a deep ensemble, that extracts two types of low dimensional but discriminative and high-level abstracted features from each DCNN, then combines them as a descriptor for face verification. Our DCNNs are built from stacked multi-scale convolutional layer blocks to present multi-scale abstraction. To train our DCNNs, we use different resolutions of triplets that consist of reference images, positive images, and negative images, and triplet-based loss function that maximize the ratio of distances between negative pairs and positive pairs and minimize the absolute distances between positive face images. A deep ensemble is generated from features extracted by each DCNN, and used as a descriptor to train the joint Bayesian learning and its transfer learning method. On the LFW, although we use only $198,018$ *images and only four different types of networks, the proposed method with the joint Bayesian learning and its transfer learning method achieved* 98.33% *accuracy. In addition to further increase the accuracy, we combine the proposed method and high dimensional LBP based joint Bayesian method, and achieved* 99.08% *accuracy on the LFW. Therefore, the proposed method helps to improve the accuracy of face verification when training data is insufficient to train DCNNs.*

1. Introduction

Face recognition in unconstrained environments is a very challenging problem in computer vision. Faces of the same identity can look very different when presented in different illuminations, facial poses, and facial expressions. Such variations within the same identity could overwhelm the variations due to identity differences and make face recognition more challenging, especially in unconstrained environment; example include Eigen face analysis [24], Independent Component Analysis [2] and their extensions. These approaches commonly assume that face images are well aligned and have a similar pose to the face images in the gallery. However, in wild environments and practical applications, these assumptions are invalid. Therefore, reducing the intra-personal variations while enlarging the inter-personal differences is an important topic in face recognition, especially in face verification.

Face verification only classifies whether two faces are the same or not; the process requires reducing intra-personal variations while enlarging inter-personal variations. To do this, subspace face verification methods such as Linear Discriminant Analysis (LDA) [3], Bayesian face [15], and unified subspace [25][26] have been proposed. For example, LDA approximates intra-personal and inter-personal face variations by using two linear subspaces, then finds the projection directions to maximize the ratio between them. Recent studies have also targeted the same goal, either explicitly or implicitly. For example, metric learning [7][8][14] maps faces to some feature representation such that faces of the same identity are close to each other whereas those of different identities stay apart. These methods also rely on feature representation given by handcrafted image descriptors such as Gabor filter [12], Scale Invariant Feature Transform [13], and Local Binary Patterns [1]. Further accuracy increase has been obtained by combining several of these descriptors [27]. However, these methods are limited by their linear nature and shallow structures, whereas intra-personal and inter-personal variations are complex, highly nonlinear, and observed in high-dimensional image space. Rather than spending time attempting to engineer new image descriptors by hand, we instead propose a method to obtain new representations automatically by supervised feature learning with deep neural networks (DNNs) and new types of training set (triplets of faces).

This paper proposes a new face verification method that

uses DCNNs and a set of triplets of faces. The proposed method uses a triplet-based loss function to directly train its output to be a compact $1,024$-dimensional embedding. Our triplets consist of two positive face images and one negative face image, and the loss aims to separate the positive pair from the negative pair by the minimum ratio margin of distances between them, and to minimize the absolute distance between face images of positive pairs. The proposed method uses two $1,024$-dimension embedding features as multi-scale representations, and these representations are used to generate a deep ensemble. These representations with DCNNs and deep ensembles offer higher-order statistics such as multi-scale corners and multi-scale contours, and can be more readily adapted to new domains whereas the handcrafted descriptors may not be appropriate. With these features, we use the joint Bayesian learning method and its transfer learning method as a classifier. To further increase the accuracy of face verification, we combine our DCNN based approach and the high-dimensional LBP based joint Bayesian method.

The rest of this paper is as follows: in section 2 we describe the proposed face verification method including the triplet of faces, the triplet-based loss function, the deep ensemble, the model architecture, and training procedure; in sections 3 and 4 we present experimental results of the proposed method in comparison with the state-of-the-art and discussion; in section 5 we draw a conclusion.

2. Proposed Methods for Face Verification

2.1. Facial database to train DNNs

Training of a deep learning model for face recognition requires a large-scale facial database. Because hundreds of thousands of parameters must be optimized in DNNs, a large-scale facial database is required to train them.

Therefore, we first built a large-scale facial database by using web crawling and the available public face database such as CASIA WebFace [28]. After collecting, we post-processed to convert original facial images to canonical facial images: First, we used a face detector [11] to find face regions in a given image. Second, we used a facial feature detector [19] to find facial fiducial points in the detected facial regions. Third, we aligned the face images into a predefined 95×95 resolution of template region to transfer their canonical facial images. Finally, we construct a pyramid canonical facial image by down-sampling the canonical facial image that has a resolution of 95×95 to facial images that have resolutions of 67×67 and 47×47. We use these pyramid canonical facial images to train each DCNN.

2.2. Multi-scale Convolution Layer Block

To make multi-scale abstracted features and increase invariance to translation of the input, we design our networks

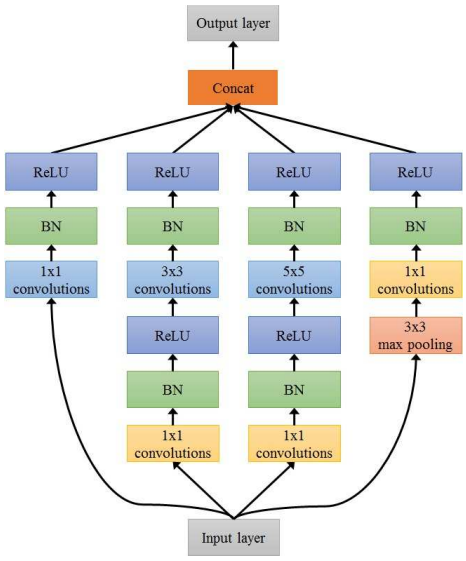


Figure 1: The multi-scale convolution layer block consists of 1×1 convolution, 3×3 convolution, 5×5 convolution, and 3×3 max pooling layers. The function of 1×1 convolution is the dimension reduction; the 3×3 and 5×5 convolutions process at different scales to achieve multi-scale feature abstraction; the 3×3 max pooling is used to be able to learn translation-invariant features. The detailed configuration is described in section 2.5 and Table 1.

with Multi-scale Convolution Layer Blocks (MCLBs). An MCLB is a convolution layer block that consists of 1×1 convolution, 3×3 convolution, 5×5 convolution, and 3×3 max pooling layers (Figure 1).

Because computation time is important in vision tasks, we use 1×1 convolution as dimension reduction, which makes our DNNs possible to train fast enough. The output of 1×1 convolution, 1×1 convolution followed by 3×3 convolution, 1×1 convolution followed by 5×5 convolution, and 3×3 max polling followed by 1×1 convolution are concatenated as one output vector. Compared to the output of a sequence of simple convolution layers, the level of abstraction in this output vector increase as the level of the layer increases. All convolution layers in the MCLB used Batch Normalization (BN) [9] and ReLU [16] as an activation function. We use this MCLB to construct shared networks for learning features for face verification. The detailed configuration is described in section 2.5 and Table 1.

2.3. Triplet of Faces

Triplet T of faces is a set of data in the form of (I_R, I_P, I_N). I_R is a given referenced facial image, I_P is a positive facial image which is similar to I_R, and I_N is a negative facial image which is not similar to I_R. Although

Figure 2: Triplet of faces in the form of (I_R, I_P, I_N) that consists of a positive face image I_P that is similar to the given face image I_R, and negative face image I_N that is not similar to I_R.

triplet T can be easily generated using similarities calculated for each of pairs of faces, we randomly select positive and negative subject images from within a collected training dataset (Figure 2); i.e., I_P is another image of the person in I_R, and I_N is an image of a different person than the one in I_R. We generated 4×10^6 triplets of faces as training data for use in training and optimizing the deep networks of the proposed face verification method.

2.4. Loss function

DNNs are trained and optimized to minimize a defined loss function. This function must be defined well.

To learn the proposed DCNNs, we use triplet loss and pairwise loss to minimize distances between faces that have the same identity and to maximize distances between faces that are of different identity. To consider identity properties, use *softmax* loss, which classifies each face image into one of n different identities (Eq. 1). The proposed loss function reduces the distance between I_R and I_P, and increase the distance between I_R and I_N (Figure 3(b)).

$$L_{total} = L_{tirplet} + L_{pairs} + L_{softmax}. \tag{1}$$

We believe that our defined loss function is more suitable for face verification than the Siamese metric learning method [6], DeepID2 [21], and DeepID3 [29] which use pairs of positive faces and negative faces only.

2.4.1 Triplet Loss

Triplet loss $L_{triplet}$ is a type of loss that uses triplets of faces to train the proposed DCNNs. The output of the network is represented by $F(I) \in R^d$, where I is the input image which is embedded into a d-dimensional space. $L_{triplet}$ is defined as:

$$L_{triplet} = \sum_{\forall T} \max\left(0, 1 - \frac{\|F(I_R) - F(I_N)\|_2}{\|F(I_R) - F(I_P)\|_2 + m}\right), \tag{2}$$

where $F(I_R)$ is the output of the proposed DCNN for I_R, $F(I_P)$ is the output of the proposed DCNN for I_P, $F(I_N)$ is the output of the proposed DCNN, and m is a margin that defines a minimum ratio between the negative pairs of faces and the positive pairs of faces in Euclidean space. $L_{triplet}$ is optimized in the training procedure to maximize the ratio of distances between the positive pairs and the negative pairs. This means that the Euclidean distance of positive pair faces is minimized and the Euclidean distance of negative pair faces is maximized (Figure 3(a)).

After first training using only $L_{triplet}$, we observed an unbalanced range of distance measured between the pairs of data; this result means that although the ratio of the distances is bounded in a certain range of values, the range of the absolute distances is not. Therefore, we also considered differences within pairs.

2.4.2 Pairwise Loss

Pairwise loss L_{pairs} is the sum of the squared Euclidean distances between the descriptor of the given facial image I_R and descriptor of the positive facial image I_P. These positive pairs I_R and I_P are selected in the triplets T.

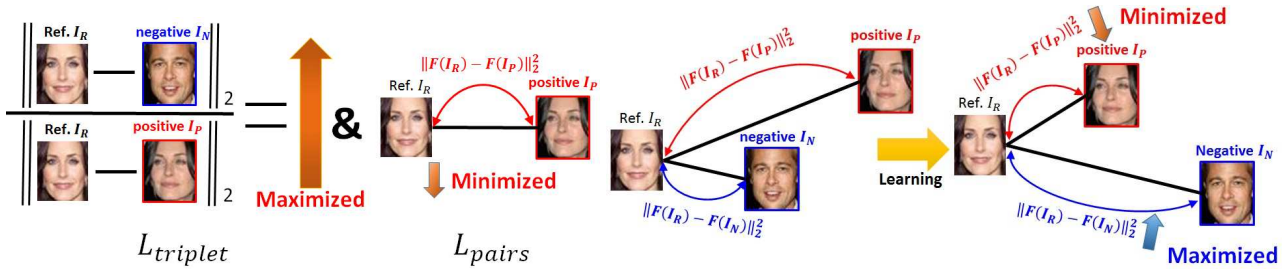
$$L_{pairs} = \sum_{(I_R, I_P) \in T} \|F(I_R) - F(I_P)\|_2^2. \tag{3}$$

The training with L_{pairs} minimizes the absolute distance between data of a given pair. We consider this loss to limit the range of absolute distance between them (Figure 3(a)).

2.5. Shared Network to Train DNNs with Triplet of Faces

Shared networks include the network for I_R, the network for I_P, and the network for I_N. The sub-network can be constructed by stacking MCLBs (Figure 4, Table 1). The pooling layer is always 3×3 max-pooling. If pooling reduces dimensionality, it is denoted with p. Finally, in MCLB, the results of 1×1, 3×3, 5×5, and pooling are concatenated to get the final output. The shared network is 24 layers deep (20 layers if we count only layers with parameters). All convolution layers and fully connected layers use the ReLU non-linear activation [16] and Batch-Normalization [9]. Dropout [20] only is applied to the last fully connected layer for regularization method during training. The average pooling takes the average vector of each feature map and sums out the spatial information. Because the spatial information is summed out, it is also invariant to translation of the input. To prevent the gradient-vanishing problem during training, we add an auxiliary classifier (loss) to the intermediate layer MCLB (5a) (Figure 4).

Networks can be widened by making an ensemble of output features of several networks trained with facial images with different resolutions (such as facial images with 95×95, 67×67, and 47×47 resolutions). The input to the each sub-network is one of the triplet of faces, i.e., I_R, I_P, or I_N. The shared networks encode the input triplet of faces to intermediate features, which are used to generate ensembles of DCNNs, and used later in joint Bayesian learning and its transfer learning. All sub-networks in shared



(a) triplet loss and pairwise loss

(b) The effect of the proposed loss function

Figure 3: The effect of the proposed loss function. The DCNN using the proposed loss is trained to minimize the distance between the given face image I_R and the positive face image I_P both of which have the same identity, and maximize the distance between I_R and the negative face image I_N of the different identity.

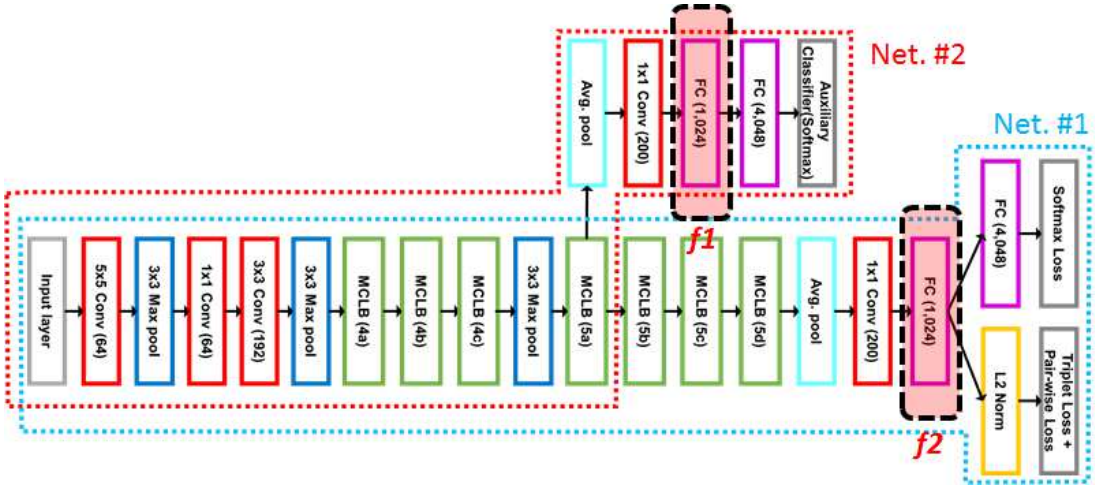


Figure 4: Sub-networks in shared networks. Sub-networks are constructed by stacking MCLBs.

networks are shared during training with T. This means that the sub-networks share all weight parameters with each other. Weight sharing can significantly reduce the number of weight parameters that must be optimized during training.

2.6. Features for Ensemble of DNNs and Face Verification

From each network, we extract two $1,024$-dimensional vectors ($f1$ and $f2$) from the fully-connected layer ($fc1$) and the last 2nd fully-connected layer in the auxiliary classifier that is connected to intermediate layer MCLB ($5a$) (Figure 4, Table 1). After obtaining features of each network, these features are concatenated to one vector feature and the total length of concatenated features is $8,192$ ($1,024 \times 4$). PCA then reduces the dimensionality of the features from $8,192$ to $1,024$. Then, the reduced features are used to train the joint Bayesian model and its transfer learning models [4] on the LFW dataset.

For verification, we use joint Bayesian learning model and its transfer learning. Joint Bayesian learning is a conventional face-verification learning method. It has been successfully used to model the joint probability of given two face images, i.e., whether they depict the same or different persons. This model learns the feature representation f of a given face image as the sum of inter-personal variations and intra-personal variations which are modeled as Gaussian distributions and are estimated from the training data. Face verification is achieved using the log-likelihood ratio $\log \frac{p(f_1,f_2,|H_I)}{p(f_1,f_2,|H_E)}$, where the numerator and denominator are joint probabilities of two face images given the intra-personal variation hypothesis H_I and inter-personal variation hypothesis H_E, respectively. We learned the joint Bayesian learning model for face verification based on the ensemble features of DCNNs of the collected training facial database.

Transfer learning is useful when analyzing plentiful source-domain data with limited samples from some target domain of interest. In face-verification evaluation on the LFW, the number of images in the LFW is less than

Table 1: Configuration of the proposed sub-network for face verification.

type	output size	1×1	3×3 reduce	3×3	5×5 reduce	5×5	pool proj (*p*)
conv1 ($5 \times 5 \times 3, 2$)	$97 \times 97 \times 64$	-	-	-	-	-	-
max pool ($3 \times 3, 2$)	$48 \times 48 \times 64$	-	-	-	-	-	-
conv2 (1×1)	$48 \times 48 \times 64$	-	-	-	-	-	-
conv3 (3×3)	$48 \times 48 \times 192$	-	-	-	-	-	-
max pool ($3 \times 3, 2$)	$24 \times 24 \times 192$	-	-	-	-	-	-
MCLB (4a)	$24 \times 24 \times 256$	64	96	128	16	32	m, $32p$
MCLB (4b)	$24 \times 24 \times 320$	64	96	128	32	64	m, $64p$
MCLB (4c)	$12 \times 12 \times 384$	-	128	256, 2	32	64	m, 2, $64p$
max pool ($3 \times 3, 2$)	$6 \times 6 \times 384$	-	-	-	-	-	-
MCLB (5a)	$6 \times 6 \times 640$	256	96	192	32	64	m, $128p$
MCLB (5b)	$6 \times 6 \times 640$	224	112	224	32	64	m, $128p$
MCLB (5c)	$6 \times 6 \times 640$	192	128	256	32	64	m, $128p$
MCLB (5d)	$3 \times 3 \times 480$	-	144	288, 2	32	64	m, 2, $128p$
avg pool	$1 \times 1 \times 480$	-	-	-	-	-	-
conv6 (1x1)	$1 \times 1 \times 200$	-	-	-	-	-	-
fc1	$1 \times 1 \times 1024$	-	-	-	-	-	-
fc2	$1 \times 1 \times 4048$	-	-	-	-	-	-
L2 normalize	$1 \times 1 \times 1024$	-	-	-	-	-	-

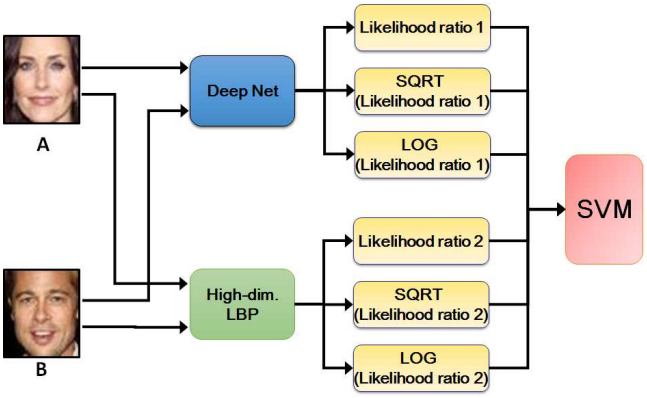


Figure 5: Hybridization with high-dimensional LBP based joint Bayesian method in the manner of the score-level fusion.

the number of images used for training DCNN. Therefore, we also use transfer learning based on the joint Bayesian learning model to increase the accuracy of face verification results.

2.7. Hybridization with High dimensional LBP

To further increase the accuracy, we combine our DCNN based method and the high-dimensional LBP based method [5] in the manner of the score-level fusion. We calculate scores based on log likelihood ratios of joint Bayesian method from each method. Then, we obtain six different types of scores which are transformed to scores of squared root and log-scale. With six different types of scores, we used support vector machine as a final classifier (Figure 5).

3. Experiment

We evaluated the proposed method on the LFW, which reveals the state-of-the-art of face verification in the wild. Although the LFW contains $5,749$ people, only 85 have > 15 images, $4,069$ people have only one image, and the total number of images is $13,233$, so this database is not suitable to train our proposed DCNNs. Instead, we used our collected face image set which contains $4,048$ people with > 10 images. The total number of collected face images is $198,018$. People in the LFW and our collected face image set are mutually exclusive. For data augmentation, each image in the collected image set is flipped horizontally ($198,018 \times 2 = 396,036$). For feature learning, we generated 4×10^6 triplets of faces with $340,289$ face images in our collected image set and used the remaining $55,747$ images as a validation set during training.

We implemented the proposed method based on the GPU version of the Caffe deep learning framework [10]. We used standard back-propagation on the feed-forward net by stochastic gradient descent optimization with momentum [17] set to 0.9 and learning rate set to 0.25 to train the proposed DCNNs. The size of mini-batch is set to 48. We initialized weights in each layer from a zero-mean Gaussian distribution with $s.d. = 0.01$, and biases set to 0.5. We trained the proposed DCNNs for roughly 1×10^6 iterations over the whole set of triplets of faces; this process took three weeks.

3.1. Loss function

We evaluated different design choices of loss function in terms of the accuracy of classification on the validation set. This validated the necessity of using a loss function.

Using the proposed network (Figure 4), we constructed a network $DNN + LT$ using $L_{triplet}$ and $L_{softmax}$, and a network $DNN + LT + LP$ using $L_{triplet}$, L_{pairs}, and $L_{softmax}$. We achieved 91.32% accuracy for $DNN + LT$ and 93.45% accuracy for $DNN + LT + LP$. From this evaluation, when using $DNN + LT$, we observed the unbalanced range of distance measured between the pairs of data. This result means that although the ratio of the distances is bounded in a certain range of values, the range of the absolute distances is not. Therefore, we also considered differences within pairs. In contrast, we observed that the distances are bounded in a certain range of values when using $DNN + LT + LP$. Therefore, we used $DNN + LT + LP$ in all experiments.

3.2. Ensemble of DNNs

We combine multiple networks trained by feeding different resolutions of inputs to the DNN: 1) The network DNN-95-a for face images with 95×95 resolution; 2) The network DNN-67 for face images with 67×67 resolution; 3) The network DNN-47 using for images with 47×47 resolution; 4) The network DNN-95-b for face images 95×95 resolution and a different margin value, 0.5, in $L_{triplet}$ (2). We extracted two $1,024$-dimensional features from the output $f1$ of the fully-connected layer ($fc1$) and the output $f2$ of the 2nd last fully-connected layer in the auxiliary classifier connected to intermediate layer MCLB (5a) on each network (Figure 4). To make an ensemble, we generated different combinations of those features: $EN_1 := F95\text{-}a + F95\text{-}b + F67 + F47$; $EN_2 := F95\text{-}b + F95\text{-}a + F67 + F47$; $EN_3 := F47 + F67 + F95\text{-}a + F95\text{-}b$; $EN_4 := F47 + F67 + F95\text{-}b + F95\text{-}a$. The dimensionality of each ensemble was reduced from $8,192$ to $1,204$.

We evaluated different types of ensembles in terms of the accuracy of the verification on the LFW. We used the joint Bayesian method as a classifier. We achieved 96.23% accuracy of verification for EN_1, 96.01% accuracy of the verification for EN_2, 95.23% accuracy of the verification for EN_3, and 95.01% accuracy of the verification for EN_4. Therefore, we used EN1 to evaluate the accuracy of the proposed method on the LFW.

3.3. Results on the LFW

We evaluated the proposed method in terms of the accuracy of the face the verification on the LFW. We used the transfer learning method based on the joint Bayesian learning method as a classifier. We followed the unrestricted protocol with labeled outside data for the verification. We trained the classifier on $5,400$ pair labels per split, and used the other 600 pair labels as the test set. We used 10-fold cross validation and averaged the accuracy over 10 trials.

We achieved 96.23% accuracy of the proposed method with only the joint Bayesian learning method as a classifier on the LFW, 98.33% accuracy of the proposed method with both joint Bayesian learning and its transfer learning method as a classifier, and achieved 99.08% accuracy of the proposed score-level fusion method. Although the accuracy of the proposed method is less than that of the state-of-the-art (Google's FaceNet), we used only $198,018$ face images to train the proposed DCNNs; this is much fewer than required by DeepFace (4×10^6 images), and FaceNet (2×10^9 - 4×10^9 images). We used only four different types of networks to generate the ensemble of DCNNs. In contrast, DeepFace uses nine different types of networks, DeepID used 120 different types of networks, and DeepID3 used 50 different types of networks (Table 2). Therefore, the proposed face recognition method is useful when the training data is limited to train DCNN models.

4. Discussion

4.1. Loss function

Design of the loss function to train DNNs affects the accuracy of classification. Accuracy was 93.45% when $DNN + LT + LP$ loss function was used, and $91.32\%4$ when $DNN+LT$ loss function was used. Especially, when $DNN+LT+LP$ was used, the distances and the ratios of distances were bounded in a certain range of values. However, when $DNN + LT$ was used, the distances were not bounded in a certain range, but the ratio of distances was bounded in a certain range. These results demonstrate that multi-task learning has the potential to improve the accuracy of classification in face verification because the tasks influence each other through a shared representation (or shared DNN layers) during training. Therefore, these results are important to design the loss function and to train DNN models. Because face verification is a classification problem, the proposed loss function (multi-task learning) may be also widely applicable to other classification problems such as general object classification.

4.2. Ensembles of DNNs

Use of an ensemble of DNNs trained by feeding different resolutions of input facial images improved the accuracy of face verification. Among the different types of ensembles, our best result was 96.23% accuracy of verification for EN_1 by using the joint Bayesian method as a classifier. The deep representation generated from the proposed ensemble has more-powerful abstracted features than the other ensembles including EN_2, EN_3, and EN_4; i.e., this result indicates that the combination order of features to generate an ensemble is important to improve the accuracy of face verification, and also indicates that if the resolution of the facial images used in DNNs is increased, the amount of useful information that DNN models can obtain also increases.

Table 2: Comparison of the number of DCNNs, the number of images, the dimensionality of feature, and the accuracy of the proposed method with the state-of-the-art on the LFW

Method	No. of DNNs	No. of images	Feature dimension	Accuracy (%)
Human	-	-	-	97.53
Joint Bayesian	-	99,773	8,000	92.42
Fisher vector faces	-	N/A	256	93.03
Tom-vs-Pete classifiers	-	20,639	5,000	93.30
High-dim. LBP	-	99,773	2,000	95.17
TL-Joint Bayesian	-	99,773	2,000	96.23
DeepFace	9	$4M$	$4,096 \times 4$	97.25
DeepID	120	202,599	150×120	97.45
DeepID3	50	300,000	300×100	99.52
FaceNet	1	$200 - 400M$	128	99.63
Learning from Scratch	2	494,414	160×2	97.73
Proposed Method (+Joint Bayesian)	4	198,018	1,024	**96.23**
Proposed Method (+TL-Joint Bayesian)	4	198,018	1,024	**98.33**
Proposed Method (Hybridization)	4	198,018	6	**99.08**

4.3. Comparison with state-of-the-art

The proposed method with only the joint Bayesian learning method on the LFW achieved 96.23% accuracy, the method with both the joint Bayesian learning and its transfer learning method on the LFW achieved 98.33% accuracy, and the method based on the score-level fusion on the LFW achieved 99.08% accuracy. Although the accuracy of the proposed method is less than that of the state-of-the-art (Google's FaceNet [18]: 99.63%), we used only $198,018$ facial images to train the proposed DCNNs; this is much fewer than required by DeepFace (4×10^6 images) [23], Learning from Scratch ($494,414$ images) [28], and FaceNet (2×10^9 - 4×10^9 images). We used only four different types of networks to generate the ensemble of DCNNs. In contrast, DeepFace uses nine different types of networks, DeepID [22] uses 120 different types of networks, and DeepID3 [29] uses 50 different types of networks. Therefore, when the training data is not enough to train DNN models, the proposed loss function (multi-task learning) and ensemble can help to improve the accuracy of face verification.

5. Conclusion

This paper presents a face recognition method that uses MCLBs to construct deep and wide networks, and uses an ensemble EN_1 generated from different types of DCNN features trained by using a set of triplets and the proposed loss function L_{total} (multi-task learning). On the LFW, the proposed method with only the joint Bayesian learning method achieved 96.23% accuracy, the proposed method with both the joint Bayesian learning and its transfer learning method achieved 98.33% accuracy, and the proposed method based on the score-level fusion achieved 99.08% accuracy. Although the accuracy of the proposed method is less than that of the state-of-the-art (Google's FaceNet: 99.63%), the proposed method has two contributions: 1) the proposed method used only $198,018$ facial images to train the DCNNs; this is much fewer than required by DeepFace (4×10^6 images), Learning from Scratch ($494,414$ images), and FaceNet (2×10^9-4×10^9 images), 2) the proposed method used only four different types of networks to generate the ensemble of DCNNs. In contrast, DeepFace uses nine different types of networks, DeepID uses 120 different types of networks, and DeepID3 uses 50 different types of networks. Therefore, the proposed method including EN_1, and L_{total} helps to improve the accuracy of face recognition when the amount of training data is insufficient to train DCNN models.

Acknowledgment

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ICT Consilience Creative Program (IITP-R0346-16-1007) supervised by the IITP (Institute for Information & communications technology Promotion) and also supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the SW STARLab support program (IITP-2017-0-00897) supervised by the IITP (Institute for Information & communications Technology Promotion)

References

[1] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(12):2037–2041, Dec. 2006. 1

[2] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski. Face recognition by independent component analysis. *IEEE Transactions on Neural Networks*, 13(6):1450–1464, Nov 2002. 1

[3] P. N. Belhumeur, J. a. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):711–720, July 1997. 1

[4] X. Cao, D. Wipf, F. Wen, G. Duan, and J. Sun. A practical transfer learning algorithm for face verification. In *2013 IEEE International Conference on Computer Vision*, pages 3208–3215, 2013. 4

[5] D. Chen, X. Cao, F. Wen, and J. Sun. Blessing of dimisionality: High dimensional feature and its efficient compression for face verification. In *CVPR 2013*, 2013. 5

[6] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1, June 2005. 3

[7] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *2009 IEEE 12th International Conference on Computer Vision*, pages 498–505, Sept 2009. 1

[8] C. Huang, S. Zhu, and K. Yu. Large scale strongly supervised ensemble metric learning, with applications to face verification and retrieval. *CoRR*, abs/1212.6094, 2012. 1

[9] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 448–456, 2015. 2, 3

[10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 5

[11] B. Jun, I. Choi, and D. Kim. Local transform features and hybridization for accurate face and human detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1423–1436, June 2013. 2

[12] 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, Apr 2002. 1

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 1

[14] A. Mignon and F. Jurie. Pcca: A new approach for distance learning from sparse pairwise constraints. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2666–2672. IEEE, 2012. 1

[15] B. Moghaddam, T. Jebara, and A. Pentland. Bayesian face recognition. *Pattern Recognition*, 33(11):1771 – 1782, 2000. 1

[16] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 807–814, 2010. 2, 3

[17] N. Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145 – 151, 1999. 5

[18] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. *ArXiv e-prints*, Mar. 2015. 7

[19] J. Shin and D. Kim. Hybrid approach for facial feature detection and tracking under occlusion. *IEEE Signal Processing Letters*, 21(12):1486–1490, Dec 2014. 2

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, Jan. 2014. 3

[21] Y. Sun, X. Wang, and X. Tang. Deep Learning Face Representation by Joint Identification-Verification. *ArXiv e-prints*, June 2014. 3

[22] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 1891–1898, Washington, DC, USA, 2014. IEEE Computer Society. 7

[23] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, June 2014. 7

[24] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1):71–86, Jan. 1991. 1

[25] X. Wang and X. Tang. Unified subspace analysis for face recognition. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 679–686 vol.1, Oct 2003. 1

[26] X. Wang and X. Tang. A unified framework for subspace face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1222–1228, Sept 2004. 1

[27] L. Wolf, T. Hassner, and Y. Taigman. *Similarity Scores Based on Background Samples*, pages 88–97. ACCV'09. Springer-Verlag, Berlin, Heidelberg, 2010. 1

[28] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *CoRR*, abs/1411.7923, 2014. 2, 7

[29] X. W. Yi Sun, Ding Liang and X. Tang. Deepid3: Face recognition with very deep neural networks. *CoRR*, abs/1502.00873, 2015. 3, 7