A Patch Strategy for Deep Face Recognition

Yanhong Zhang[a], Kun Shang[a], Jun Wang[b], Nan Li[a], Monica M.Y. Zhang[c]

[a] *Center for Applied Mathematics, Tianjin University, Tianjin 300072, P.R. China*
[b] *School of Mathematics, Tianjin University, Tianjin 300072, P.R. China*
[c] *Center for Combinatorics, LPMC Nankai University, Tianjin 300071, P.R. China*

Abstract

Convolutional Neural Network (CNN) has proven to be a highly efficient approach to face recognition. In this paper, we introduce a new layer to embed the patch strategy in convolutional architectures to improve the effectiveness of face representation. Meanwhile, a multi-branch CNN is constructed to learn features of each cropped patch by the patch strategy, and then fuses all the patch features together to form the entire face representation. Compared with the traditional patch methods, our approach has the advantage that no extra space is needed to store the facial patches since the images are cropped online. Moreover, due to the end-to-end training, this approach makes a better use of the interactions between global and local features in the model. Two baseline CNNs (i.e., AlexNet and ResNet) are used to analyze the effectiveness of our method. Experiments show that the proposed system achieves comparable performance with other state-of-the-art methods on the LFW and YTF face verification tasks. To ensure the reproducibility, the publicly available training set CASIA-WebFace is used.

1. Introduction

Convolutional Neural Network (CNN) has attracted extensive attention in image processing, pattern recognition, computer vision [1, 2, 4, 3], improving the state of classification problems. Particularly, benefiting from the discriminative and effective representations extracted by CNN models, face recognition (FR) via CNN has achieved great success [5, 6, 7, 8].

Email address: monicazhang@mail.nankai.edu.cn (Monica M.Y. Zhang)

Many FR techniques based on CNN focus on learning discriminative and generalized representations. As is well known, effective feature extraction plays a crucial role in FR process [9, 6, 10]. Choosing an appropriate face representation can make the subsequent face processing not only computationally feasible but also robust to possible intrinsic and extrinsic facial variations. Existing face features can be divided into two categories: global-based features and local-based features [9]. The global-based face representation captures more semantic information embodied in every part of the face image, which corresponds to some holistic characteristic of the face. In contrast, the local-based feature vector corresponds to certain local face region, and only encodes the detailed attributes within this specific area [9]. Indeed, local features extracted from different face regions have several advantages such as insensitivity to local variations [11], while global features themselves are discriminative but not effective enough for FR. Thus, face recognition can generally benefit more by integrating global features with local features.

Recently, the strategy of fusing patches has been adopted to extract features of various face regions with CNN models. The primary framework of these methods processes multi-patch features by classifiers or multi-patch ensemble CNN models. Sun *et al.* [6] resorted to multi-patch ensemble models to boost performance. They divided the face images into 100 patches including the global and local ones and trained these patches with multiple CNN models separately. Later, they selected the best 25 patches from 400 cropped ones and employed both identification and verification signals as supervision, improving the performance by a large margin [7]. Hu *et al.* [12] sampled 30 patches from five facial regions to explore the spatial information of facial parts by concatenating all the features of 30 patches. Besides, by taking advantage of multiple CNNs, Ding and Tao [13] extracted complementary deep features from six patches and compressed the high-dimensional representations by a three-layer stacked auto-encoder (SAE). However, the existing multi-patch based methods cannot make a better use of the interactions among multiple patches because the models are trained separately on individual patches.

In this paper, we introduce a patch strategy in CNN architectures to learn complementary and effective features in an end-to-end fashion. A new network layer is proposed to uniformly sample several facial patches within the CNN model. Meanwhile, we construct a multi-branch network structure to implement the complementary information extraction and integration. For each patch, we use a convolutional network branch to learn and extract its

feature. Then we fuse features from the entire face and its local regions. Especially, all features are first normalized by BatchNorm and then cascaded together. Finally, the concatenation of features is further integrated by a fully-connected layer without dimensionality reduction. The trade-off between the global features and local features can be completed in the proposed framework due to end-to-end training. We explore two different baseline deep network architectures (AlexNet and ResNet) to verify the effectiveness of our approach. The contributions of our work are summarized below:

- The proposed approach enables the network to divide a face image into patches and to learn patch features simultaneously.
- No extra space is needed to store the local patches since the model takes an entire face image as input and the images are cropped online.
- Compared with existing multi-patch based methods, the face representations can be intensified by optimizing the parameters of each patch in a single model and better performance is acquired.
- Comparable results with the state-of-the-art are achieved with less training data.

The rest of this paper is organized as follows. In Section 2, we review the literature of various feature extraction approaches. Section 3 presents the idea of the patch strategy and the proposed multi-branch networks. In Section 4, we describe the CNN structures in detail. Experimental results are reported in Section 5. Finally, the conclusion is given in Section 6.

2. Related Work

Effective face representation plays a crucial role in face recognition. Several recognition techniques have been developed to capture discriminative features for better performance. Appearance-based methods are studied to extract global information from the entire face, such as principal component analysis (PCA), linear discriminant analysis (LDA) and locality preserving projection (LPP). However, they are sensitive to local variations like poses, illuminations and expressions. To avoid the problems, local characteristics are researched mainly by partitioning the whole image into patches for feature extraction. Two-dimensional PCA and two-dimensional LDA were used to project facial regions into local subspaces for better performance on occluded

faces [14]. In [15], the authors employed dimensionality reduction approach to extract robust features from multiple facial parts. To take advantage of both global and local characteristics, model-based methods also draw the attention. Zhu [16] extracted local features from various face regions and adopted NMF decomposition on different sampled features to achieve better performance. Hariri *et al.* [17] encoded and fused different kinds of features and modalities using covariance-based descriptors.

Recently, deep learning has become a focus topic of face recognition since it is powerful to learn highly discriminative features. In contrast to the above traditional methods with hand-crafted features and classifiers, deep learning based approaches propagate information in a complex hierarchical structure and automatically learn features by mapping low-level features to more abstract high-level ones. The approaches usually include two steps: high-dimensional feature extraction and classifier design. More specifically, a CNN model is first trained to extract a high dimensional feature vector [6, 2, 12]; then, Joint Bayesian [6] or metric learning method [18] is used for classification. The CNN models naturally integrate the feature extractor and the classifier in an end-to-end fashion. The face representations obtained by the methods are more effective and general.

The success of CNN has inspired extensive researches on deep face recognition. Researchers of Facebook in 2014 initiated the feature extraction by CNN and achieved state-of-the-art results at that time [5]. They presented a representative system named DeepFace, which employed explicit 3D face model and learned the face representations in a nine-layer deep neural network. Later on, Sun *et al.* extended this work by DeepID series of papers [6, 7]. The ultimate verification accuracy they attained on LFW is 99.47% with 25 CNN models. Very recently, deeper network architectures such as GoogleNet proposed by Szegedy *et al.* [3] and ResNet presented in [2] have been widely used for face recognition. FaceNet [8] employed the *Inception* model [3] to directly learn the triplet embedding for face verification, achieving the accuracy of 99.63% on LFW. Wen *et al.* [19] supervised the CNN by a novel signal center loss together with the softmax loss and obtained the state-of-the-art accuracy on three important face recognition benchmarks. Following the trend, we learn face features by using CNN and further introduce the patch strategy to improve the performance of face representation in the paper.

3. Patch Strategy Embedding in Networks

In this section, we present the proposed approach based on multi-patch. The patch strategy of sampling patches and the proposed framework are described in detail, respectively. Both the intuitive illustration and the interpretation are also given for a better understanding.

3.1. The Proposed Patch Strategy

We propose a patch strategy to divide an input training image into different patches online. Different from [7] that randomly croped 25 patches over the face image using dense facial landmarks, we uniformly sample six image patches of size 136×136 pixels with sparse facial key points from an aligned face image. Figure 1 shows the six cropped patches. The position of each facial patch is constant for all the images. In contrast to cropping face images offline, the proposed patch strategy is conducted online. A new network layer named "crop-data" is introduced into convolutional networks to implement it. Given a batch of training face images, the crop-data layer crops each image into a facial patch using the corresponding cropping window (x, y, w, h) with top-left corner at (x, y) of size (w, h). In particular, the top left coordinates (x, y) of six patches shown in Fig. 1*b* are $(0, 0)$, $(42, 0)$, $(45, 25)$, $(27, 50)$, $(26, 75)$, $(45, 88)$, respectively. $w \times h$ is just the size of the cropped patches. After cropping, the layer sends the patches to different network branches. This operation and the subsequent convolution, pooling etc. are carried out simultaneously during training. The layer serves as a bridge to connect the network branches about the whole images and the local patches together. It makes us achieve an end-to-end network architecture.

By this operation, no extra space is needed to store the local patches. Besides, not only multimodal face features can be extracted from different facial regions in a single model, but also the complementary information contained in the whole image and its local patches can be explored. In this way, it can capture the interaction among different facial regions and better leverage the holistic information of an image and its details. Furthermore, our patch strategy enables the optimization of the CNN model to utilize the prior knowledge that the face patches belong to.

3.2. Networks With Patch Strategy

Our framework for face representation consists of two parts: feature extraction from the holistic face image and the local patches using CNNs, and feature fusion by a fully-connected layer.

a *b*

Figure 1: (a) The original aligned face image. (b) Image patches uniformly sampled from the aligned image.

For the feature extraction, as shown in Fig. 2, we take a well-designed and widely used network architecture to extract features from an entire face, and denote it by CNN-1 as a baseline network. The CNN-2 is constructed according to CNN-1 for a local face region. The main difference between CNN-1 and CNN-2 is that CNN-1 is deeper than CNN-2. The details of CNN-1 and CNN-2 will be presented in Sect. 4. Using the proposed networks, we select the output of the last layer of CNN-1 and CNN-2 as global features and local features, respectively. For the feature fusion, we first normalize global features and local features by BatchNorm [20] and then cascade them together, which can boost the performance and speed up the convergence of networks. Furthermore, the concatenation of all features is integrated by a fully-connected layer without dimensionality reduction. The output of this layer is taken as the face representation.

In our network architectures, a complex structure is proposed for the entire images that have rich information. For the cropped patches, it is appropriate to use a network with relatively fewer parameters to learn efficient patch features. This is due to the difference in size and the semantic meaning between a holistic image and its patches. Consequently, a structure similar to but simpler than CNN-1 is designed for the local patches with less information. To evaluate the contribution of each sampled patch to the face representation, we adopt the same network architectures for different patches, avoiding the influence from CNN structures. More specifically, the

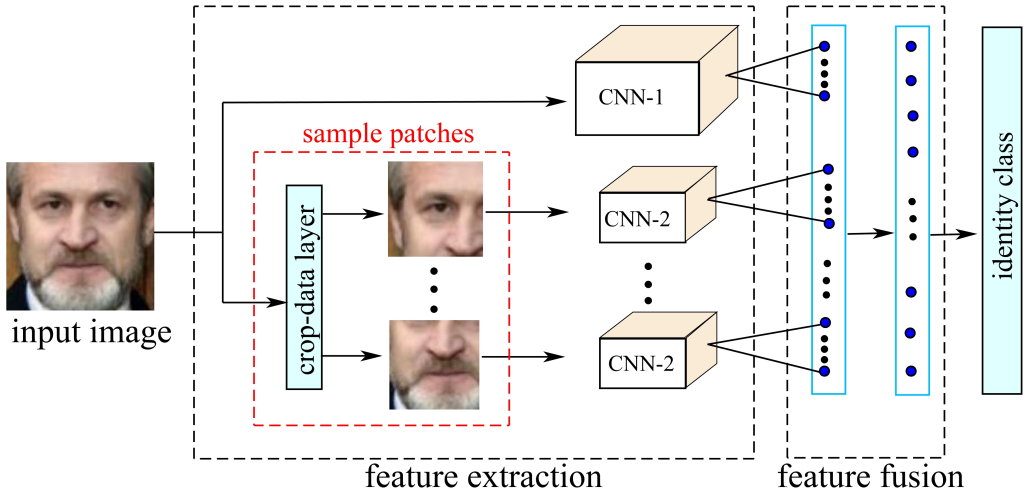


Figure 2: Flowchart of the proposed networks with patch strategy. The CNN extracting features from the entire image is denoted as CNN-1, while a similar but simple structure, denoted by CNN-2, is designed to extract patch features. The representations are extracted from the last hidden layer and used to predict n identity classes ($n = 10575$).

parameters of CNN-2 for various patches are not shared so that the features learned by each CNN-2 are adaptive to multimodal information included in different face regions. In contrast with existing works related to multiple patches, the proposed framework takes a holistic image as input and samples local patches with the crop-data layer, then sends them to different network branches. In this way, multimodal features are learned and interact with each other during training in an end-to-end fashion, which further boosts the performance of the face representation.

3.3. Interpretation of the Proposed Framework

Let $D = \{(X_i, y_i)\}_{i=1}^{N}$ be the training set, where X_i denotes the i-th training sample, y_i is the ground-truth label and N is the number of the training samples in D. For an input image X_i, the feature extraction process of X_i is denoted as $\boldsymbol{x}_i = conv(X_i, \theta_c)$, where $conv(\cdot)$ represents the feature extraction function defined in CNN, θ_c is the parameter to be learned and $\boldsymbol{x}_i$ is the extracted feature vector.

In the proposed framework, the parameters of each branch are learned simultaneously in a CNN model. The final representation $f_J(X)$ of an face

image X in our end-to-end training architecture is the feature vector

$$f_J(X) = \boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{x} + \boldsymbol{b}, \tag{1}$$

where $\boldsymbol{x} = [\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$, $\boldsymbol{W} \in \mathbb{R}^{(n+1)d\times(n+1)d}$ and $\boldsymbol{b} \in \mathbb{R}^{(n+1)d}$. Here $\boldsymbol{x}_i = conv(I_i, \theta_{ci}) \in \mathbb{R}^d$ is the feature vector of the i-th patch I_i cropped from X. They are extracted from the last layer of each branch in Fig. 2. For simplicity, $I_0 = X$ is the holistic image. As is known, effective features are obtained by optimizing the parameters of CNNs using backpropagation. We present the backpropagation gradient of the parameters as following.

Let $L(D; \theta_c)$ be the loss function to measure the error between the prediction value and the ground truth. The prediction score vector of X is the output of the last layer, denoted as $p(X)$. For the parameter $\boldsymbol{w}_i$ of i-th network branch, the backpropagation gradient of $\boldsymbol{w}_i$ is

$$\nabla_{\boldsymbol{w}_i} L(D; \theta_c) = \frac{1}{|M|} \sum_{(X,y)\in M} \frac{\partial L(D; \theta_c)}{\partial p(X)} \times \frac{\partial p(X)}{\partial f_J(X)} \times \frac{\partial f_J(X)}{\partial \boldsymbol{x}} \times \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{w}_i}, \tag{2}$$

where M is a mini-batch randomly drawn from the training set.

In contrast to our approach, the traditional multi-patch based methods usually crop patches offline and take the patches as input to train multiple models separately. They combine all features extracted from pre-trained models together as the final face representations, denoted as

$$f_S(X) = [\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n], \tag{3}$$

All the patch features $\boldsymbol{x}_i$ are extracted from the penultimate layer of the networks. The backpropagation gradient of the parameters in this layer is

$$\nabla_{\boldsymbol{w}_i} L(D_i; \theta_{ci}) = \frac{1}{|M_i|} \sum_{(I_i,y_i)\in M_i} \frac{\partial L(D_i; \theta_{ci})}{\partial p(I_i)} \times \frac{\partial p(I_i)}{\partial \boldsymbol{x}_i} \times \frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{w}_i}, \tag{4}$$

where $D_i = \{(I_i^{(n)}, y_n)\}_{n=1}^N$, $I_i^{(n)}$ is the i-th patch of the n-th sample and M_i is a mini-batch randomly drawn from D_i. Due to the randomness of M_i, different patches may belong to different persons in a mini-batch of distinct models.

From (3), f_S simply concatenates the obtained features for feature fusing. For the proposed approach, we embed the global features and the local features into a new space by an affine mapping. It can enhance the contribution

of certain features to the final representation and reduce the influences of other features. Moreover, the interactions between global and local features by f_J can be better utilized. On the other hand, from the optimization aspect, the update of the parameters of different models is independent of each other as shown in (4). In other words, information from a patch cannot affect the others during training though different patches have a high correlation with each other. Different from (4), each CNN branch of our method shares the gradient propagated back from the last two layers in a mini-batch (see (2)). It implies the update of the parameters of each branch interacts with each other mutually. Overall, a nice tradeoff between global features and local features can be achieved by our system, which benefits effective and discriminative representations.

4. Network Architectures

A large number of the CNN architectures are developed with powerful capabilities to represent face in the literature. Here, we select two typical networks AlexNet [1] and ResNet [2], which contain the most mainstream components of CNN architectures, as our baseline networks. By using the two baseline networks, we make our experiments to have universal applicability as much as possible. More implementation details of the two networks are described as follows.

AlexNet: AlexNet, proposed by Geoffrey and Alex, drew attention on the ImageNet 2012 Challenge for its powerful performance of extracting features. It is the first deep CNN that achieves significant success on large scale datasets. It also has good generalization to extract features in many other application situations, especially computer vision tasks [21]. The architecture contains 5 convolutional layers, 3 max-pooling layers and 3 fully-connected layers, which are typical components in a common CNN architecture. Here we use it as a baseline model for learning face features. To make it comparable with face recognition and to accelerate the training, we conduct a slight modification by reducing the number of neurons of the fc6 layer and the fc7 layer from $4,096$ to $2,048$ and $1,024$, respectively. The output of the fc7 layer is used as the feature of a holistic face.

ResNet: ResNet, emerging as a deeper CNN architecture, won the 1st place on the ILSVRC 2015 classification task. It takes advantage of a well-designed short-cut layer to force the convergence of networks, and makes the training easier when the depth increases larger. Due to its good generaliza-

Table 1: The architectures of CNN-1 and CNN-2 constructed based on AlexNet.

CNN-1			CNN-2		
Name	Fliter Size ,Stride	Output Size	Name	Fliter Size ,Stride	Output Size
input	-	$227 \times 227 \times 3$	input	-	$227 \times 227 \times 3$
			crop-data	-	$136 \times 136 \times 3$
conv1	11×11, 4	$55 \times 55 \times 96$	conv1	7×7, 2	$65 \times 65 \times 96$
max pool1	3×3, 2	$27 \times 27 \times 96$	max pool1	3×3, 2	$32 \times 32 \times 96$
conv2	5×5, 1	$27 \times 27 \times 256$	conv2	3×3, 2	$16 \times 16 \times 256$
max pool2	3×3, 2	$13 \times 13 \times 256$	max pool2	3×3, 2	$13 \times 13 \times 256$
conv3	3×3, 1	$13 \times 13 \times 384$	conv3	3×3, 1	$8 \times 8 \times 384$
conv4	3×3, 1	$13 \times 13 \times 384$	conv4	3×3, 2	$4 \times 4 \times 256$
conv5	3×3, 1	$13 \times 13 \times 256$			
max pool3	3×3, 2	$6 \times 6 \times 256$			
fc6	-	1×2048			
fc7	-	1×1024	**fc1**	-	1×1024

tion power to learn features and the property of easier training, we choose it as another CNN baseline model. In fact, five kinds of depths of ResNet are provided in [2]: 18-layer, 34-layer, 50-layer, 101-layer and 152-layer. We select the 18-layer ResNet and take the output of the ave pool layer as the global facial feature.

The above two baseline networks are used as CNN-1 for a holistic image. For the structure of CNN-2, it is similar to but simpler than CNN-1. Specifically, the construction of CNN-2 is performed on the two baseline networks through selecting the first fewer layers of them appropriately. The neuron activations of the last hidden layer are considered as patch features with the equal dimensions of features extracted from CNN-1. For the sake of convenience, we record the two proposed networks with patch strategy as Joint-Alex and Joint-Res according to the baseline networks, respectively. The details of all the network architectures are provided in Table 1 and Table 2, which makes our implementation reproducible.

5. Experiments

In this section, firstly, the details of datasets and the settings of training and testing are described. A set of experiments are then conducted on the Labeled Faces in the Wild (LFW) [22] and YouTube Faces (YTF) [23] datasets for two face verification tasks, including both image to image face

Table 2: The architectures of CNN-1 and CNN-2 constructed based on ResNet. $B(3,3)$ denotes a residual block composed of two 3×3 convolutional layers. $B(3,3) \times 2$ denotes 2 blocks in groups of convolutions. Downsampling is conducted by the first layers in conv3_x, conv4_x and conv5_x.

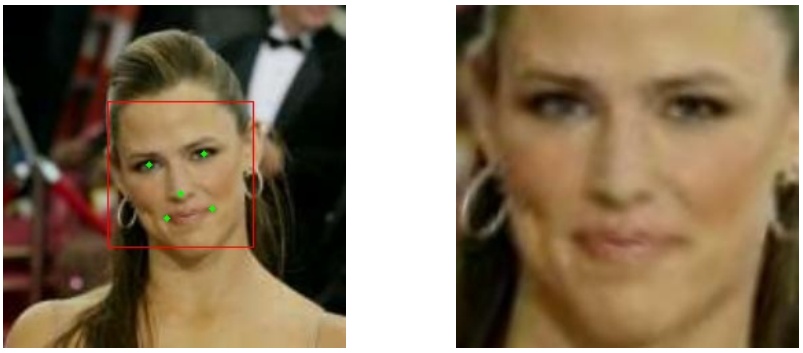
CNN-1			CNN-2		
Name	Filter	Output Size	Name	Filter	Output Size
input	-	$224 \times 224 \times 3$	input	-	$224 \times 224 \times 3$
			crop-data	-	$136 \times 136 \times 3$
conv1	$7 \times 7, 2$	$112 \times 112 \times 64$	conv1	$7 \times 7, 2$	$68 \times 68 \times 64$
max pool	$3 \times 3, 2$	$56 \times 56 \times 64$			
conv2_x	$B(3,3) \times 2$	$56 \times 56 \times 64$	conv2_x	$B(3,3)$	$68 \times 68 \times 64$
conv3_x	$B(3,3) \times 2$	$28 \times 28 \times 128$	conv3_x	$B(3,3)$	$33 \times 33 \times 128$
conv4_x	$B(3,3) \times 2$	$14 \times 14 \times 256$	conv4_x	$B(3,3)$	$16 \times 16 \times 256$
conv5_x	$B(3,3) \times 2$	$7 \times 7 \times 512$	conv5_x	$B(3,3)$	$7 \times 7 \times 512$
ave pool	$7 \times 7, 2$	$1 \times 1 \times 512$	**ave pool**	$7 \times 7, 2$	$1 \times 1 \times 512$

verification and video to video face verification. The experiments present the role of the facial patches and the advantages of the proposed approach over traditional multi-patch based CNNs. Besides, the comparison with the state-of-the-art is provided.

5.1. Datasets and Pre-processing

Face verification, deciding whether two faces belong to one subject or not, is a long-term focused issue of face recognition. To demonstrate the effectiveness of the proposed approach on face verification task, we use several famous and typical face benchmarks, including LFW dataset, YTF dataset and CASIA-WebFace dataset [24]. The details of these datasets are as follows.

LFW: It is a public dataset which contains $13,233$ images of $5,749$ people, where $1,680$ subjects have more than two images and $4,096$ subjects consist of only one image. Moreover, the face images from it are taken under an unconstrained environment with difficult face variations, such as occlusions, poses, expressions and illuminations. The dataset is organized into two "Views". View 1 has $2,200$ pairs for algorithm development. View 2 consists of 10 splits with $3,000$ genuine matches and $3,000$ impostor matches. Here, we focus on View 2 for face verification. Further, we choose $2,592$ and $3,610$ pairs from View 2 to intensify the conditions of occlusions and poses, respectively. Similarly, $4,695$ and $5,470$ pairs are screened out for the analysis of expressions and illuminations, respectively. The four subsets are



a *b*

Figure 3: An example of the face image alignment from the LFW dataset. (a) is the raw image with five detected facial points. (b) is the normalized face image aligned by similarity transformation according to the five facial points.

used to evaluate the adaption of the proposed approach on a given specific face variation.

YTF: The dataset is another famous public face dataset for evaluating the performance for face recognition. It contains $3,425$ videos of $1,595$ different people downloaded from YouTube. Each subject of YTF contains several videos with different size of frames ranging from 48 to $6,070$. Among the large scale of face videos, $5,000$ pairs of face videos are selected for demonstrating the face verification performance. Besides, the face frames of YTF not only contain large face variations (such as poses, illuminations, expressions and occlusions), but also suffer from different levels of low resolutions. Based on these, it is treated as a more challenging testing dataset for face verification.

CASIA-WebFace: The CASIA-WebFace dataset contains face images of celebrities from websites. It is a typical public dataset with wide face subjects, namely, $10,575$ subjects with $494,414$ face images. Moreover, it has no overlap with the LFW and YTF datasets and can be treated as a standard training dataset for developing face recognition methods. Thus, it is fair to evaluate the performance on LFW and YTF. In fact, there exist many noises in CASIA-WebFace such as non-face samples and subjects with incorrect labels. After manually deleting noisy data and false detected samples, $437,633$ images of $10,575$ subjects remain for training.

Next, we describe the details of image pre-processing for the mentioned datasets. For a given face image, five facial landmarks, i.e. the two eye cen-

ters, the nose tip and the two mouth corners, are detected with the proposed approach CFAN in [25], as shown in Fig. 3*a*. According to the detected five facial landmarks, the face image is aligned by similarity transformation and normalized to 256×256, as shown in Fig. 3*b*. The open source detection and alignment tools are available at https://github.com/seetaface/SeetaFaceEngine. Based on the aligned images, we resize all images to 227×227 and 224×224 as the input of networks designed based on AlexNet and ResNet, respectively.

5.2. Training Methodology

All of the models are trained with the open source deep learning framework Caffe [26]. For the proposed networks based on AlexNet, the input is 227×227 RGB images. The size of the input RGB images for the networks designed on the baseline network ResNet is 224×224. We mirror the input images for all the models when training.

In order to mitigate the overfitting and improve the generalization of the CNN models, the following techniques are used in this work. Weight decay parameter [27] as a regular coefficient in the loss function can effectively avoid overfitting under the normal distribution assumption. It is used for all the models. Dropout [28] and Local Response Normalization (LRN) [1] are also adopted to improve the generalization of the neural networks constructed on the basis of AlexNet. Specifically, we apply dropout after each fully-connected layer and set the ratio to 0.5 without declaration. LRN is applied after the first two convolutional layers with the default parameter values. The CNNs designed based on ResNet use BatchNorm [20] instead of dropout to accelerate the training by reducing the internal covariant shift of networks. In addition, Gaussian initialization is used for all the convolutional layers and the fully-connected layers in each network. To alleviate the saturation of each network, we use ReLU [29] after each convolutional layer to avoid the gradient vanishing and to force the network sparse.

The details of the training strategy are presented as follows. Two training stages are employed. We first train the models with the softmax loss to fast evaluate the effect of patches and to compare with traditional multi-patch models. Then we fine-tune the pre-trained models by the center loss proposed in [29] to achieve better performance. All the CNNs that are constructed on the basis of the same baseline networks are provided same parameter settings as the corresponding initial baseline system. Meanwhile, all the parameter settings except the initial learning rate are same for both two training stages. In the following, we present the parameter setting of two

baseline networks when training only with the softmax loss. For AlexNet, the initial learning rate is set to 0.01 and decreases by 0.5 every $20,000$ iterations. The momentum is set to 0.9 and the weight decay is 5×10^{-4} for the convolutional layers and fully-connected layers. For ResNet, we set the initial learning rate to be 0.05 and decrease it similar to AlexNet. The momentum is also set to 0.9 and the weight decay is 1×10^{-4}. When fine-tuning, the learning rate reduces by 0.1 compared with only supervised by the softmax loss. In this paper, all the experiments are conducted with a single Titan-X GPU.

5.3. Details for Testing

The deep features of the original image and its horizontally flipped version are concatenated as the raw representations. For the traditional multi-patch based CNN models, we use the concatenation of the raw representations of the holistic image and the patches. The similarity score is calculated by the cosine distance of a pair of features after transforming the representation by PCA similar to [19]. We report the results on LFW and YTF following the standard protocol of *restricted, labeled outside data* [22]. On the View 2 data of LFW, $6,000$ given pairs are split into 10-fold. Nine splits are selected to train a classifier, and then the classification is performed on the last split. Almost the same testing pattern is adopted on the YTF dataset. The first $2,500$ video pairs are employed for testing and divided into 5 folds, where 4 folds are used for training and the left one is for testing. Besides, we randomly select 100 pairs of frames per video and use the average cosine similarity of 100 pairs as the similarity of a test video pair. The estimated mean classification accuracy and its standard error (SE) are reported [22].

5.4. Evaluations

In this section, firstly, the effect of a single patch and the combinations are discussed. The performance comparison between our system and traditional multi-patch based CNN methods is then presented. Especially, the robustness to unconstrained face variations is analyzed, including occlusions, poses, illuminations and facial expressions. Finally, we provide the comparison between our ultimate results and the state-of-the-art on LFW and YTF datasets.

Figure 4: The verification accuracy (%) of an individual patch on LFW. The features are extracted from the model CNN-1 constructed on the basis of ResNet.

5.4.1. Effect of patches

Different regions of the face contain diverse information and cause different performance. A set of experiments are constructed on the LFW dataset to reveal the contribution of the sampled patches.

First of all, we discuss the role of a single patch. The verification accuracy of each cropped patch is presented in Fig. 4. As is shown, the patch that includes more and intact face landmarks performs best due to more effective information. Besides, the patches containing eyes and nose obtain better accuracy than those including nose and mouse, which indicates that the information from eyes may be more useful for face recognition than that from other landmarks.

To explore the contribution and complement of multiple patches to the performance, we combine the CNN features of multiple patches together with features from the holistic image for the computation of the cosine similarity. It is impossible to present all the combinations of patches, we just provide the performance of the best one. The comparison with traditional multi-patch based methods is also provided. For the traditional ones, we use the same CNN structures as the network branches described in Sect. 4 to extract features

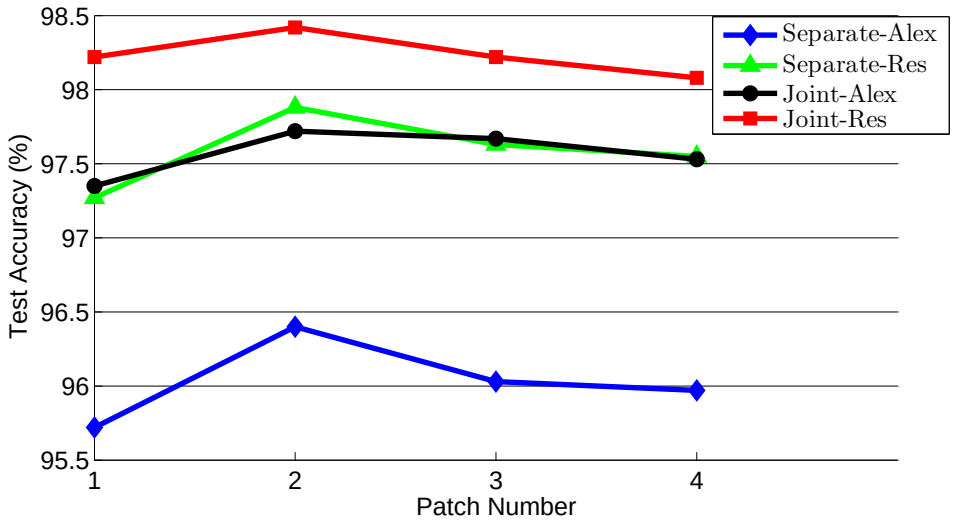


Figure 5: The verification accuracy of the combinations of the holistic image and its local patches on LFW

from individual facial regions. The methods are denoted by Separate-Alex and Separate-Res accordingly. As reported in Fig. 5, the accuracy improves as the number of patches increases, but saturates after two patches. This shows multiple patches contain effective and complementary information but can also bring a large amount of redundancy. Meanwhile, it indicates that the two local patches indeed bring effective supplements to features of the entire face image. Besides, the best combinations are the same for different methods. Especially, the patches from the best combination of the holistic image and its local patches are ones having scores of 94.78% and 93.57% in Fig. 4, respectively.

5.4.2. Comparison with Traditional Multi-patch Based CNNs

In this experiment, we compare the performance between our approach and traditional multi-patch based CNN methods. To fast evaluation, all the models employ only the softmax loss for training and the patches from the best combinations are adopted. The comparison is conducted on the LFW and YTF benchmarks, and the verification results are reported in Table 3.

From the first two rows of Table 3, Joint-Alex achieves better performance than Separate-Alex on both LFW and YTF testing benchmarks, improving the accuracy by 1.32% and 2.6%, respectively. For more powerful CNN archi-

Table 3: The performance comparison between the proposed approach and multi-patch models trained separately on LFW and YTF datasets.

Method	Networks	Acc (%) on LFW	Acc (%) on YTF
Separate-Alex	3	96.40 ± 0.26	89.72 ± 0.73
Joint-Alex	1	97.72 ± 0.25	92.32 ± 0.40
Separate-Res	3	97.88 ± 0.21	89.84 ± 0.92
Joint-Res	**1**	$\mathbf{98.42 \pm 0.24}$	$\mathbf{92.72 \pm 0.31}$

Table 4: The performance under occlusions (O), poses (P), expressions (E) and illuminations (I).

Method	Acc (%) on face variations			
	O	P	E	I
Separate-Alex	94.60	95.71	95.95	96.07
Joint-Alex	96.57	97.53	97.68	97.72
Separate-Res	97.15	97.78	97.81	97.79
Joint-Res	**97.84**	**98.34**	**98.30**	**98.34**

tectures like ResNet, Joint-Res outperforms Separate-Res by clear margins which can be observed from the last two rows of Table 3. It indicates that the performance of the proposed system is superior to that of the traditional multi-patch based CNN methods. Besides, it demonstrates the effectiveness of the system on different and deeper networks. They all imply the representations we learn are more effective and discriminative.

To further verify that our method can capture more complementary information, we analyze the robustness of two methods to occlusions, poses, expressions and illuminations on the LFW dataset. The evaluation is conducted on the four subsets of View 2. Table 4 reports the performance of two methods under the four conditions. Obviously, Joint-Alex performs better than Separate-Alex in the four circumstances, reducing the error by 40% on average. It also shows that the average error of Joint-Res is 24% lower than Separate-Res. It proves our system is more robust to occlusions, poses, expressions and illuminations. Meanwhile, the results of all the methods are comparable in the case of poses, expressions and illuminations, while they are worse under occlusions than other three cases. However, our method improves most under occlusions than the others, increasing the accuracy up to

Table 5: Comparison with the state-of-the-art on the LFW dataset.

Method	Images	Networks	Accuracy ± SE (%)
Web-Scale [32]	4.5M	4	98.37
VGG [30]	2.6M	1	97.27
MultiBatch [31]	2.6M	1	98.20
DeepFace [5]	4.4M	3	97.15 ± 0.27
DeepFace [5]	4.4M	7	97.35 ± 0.25
WebFace [24]	0.4M	1	97.73 ± 0.31
DeepID [6]	–	100	97.45 ± 0.26
DeepID2 [7]	–	25	98.97 ± 0.25
FaceNet [8]	200M	1	99.63 ± 9
Joint-Alex	0.4M	1	98.03 ± 0.23
Joint-Res	**0.4M**	**1**	$\mathbf{98.70 \pm 0.16}$

1.97% by Joint-Alex. These all demonstrate the effectiveness of the proposed approach.

5.4.3. Performance Comparison with the State-of-the-Art

The comparison with the most recent state-of-the-art on the two datasets is given in this section.

As shown in Table 5, Joint-Alex achieves an accuracy of 98.03% and Joint-Res obtains 98.70% accuracy on the LFW dataset. The results of our models outperform the performance of DeepFace [5], WebFace [24], DeepID [6] and VGG [30]. Especially for Joint-Res, it reduces the error remarkably compared with the above methods and also outperforms MultiBatch [31] and Web-Scale [32] slightly. Besides, the comparable results are achieved with less training data. Although our best model lowers the accuracy rate of FaceNet [8] by about 1%, our training sets are far inferior to theirs. With more training data, the performance is expected to be improved.

To further prove the generation of our models, we also evaluate them on YTF and report the results in Table 6. It can be observed that the verification accuracy of 92.32% and 93.12% is obtained by Joint-Alex and Joint-Res, respectively, which outperforms DeepFace [5] and WebFace [24]. Moreover, with fewer patches, the performance of Joint-Res is comparable with DeepID2+ (93.2%). It shows the advantage of our system.

Table 6: Comparison with the state-of-the-art on the YTF dataset.

Method	Images	Networks	Accuracy ± SE (%)
VGG [30]	2.6M	1	97.30
DeepFace [5]	4.4M	1	91.40 ± 1.1
WebFace [24]	0.4M	1	92.24 ± 1.28
DeepID2+ [33]	0.3M	25	93.20 ± 0.2
FaceNet [8]	200M	1	95.12 ± 3.9
Joint-Alex	0.4M	1	92.32 ± 0.40
Joint-Res	**0.4M**	**1**	$\mathbf{93.12 \pm 0.43}$

6. Conclusion

This paper proposes a novel approach by embedding a patch strategy in CNN architectures to learn sufficiently effective features for face recognition. Different from the traditional patch methods, our work provides the network with the ability to crop a face image into patches, such that it needs no extra storage space for face patches. Moreover, the process of cropping images and learning parameters of each patch can be carried out simultaneously in a CNN structure. Additionally, to trade off all patch features, we cascade the extracted normalized global and local features together, and then map the concatenated features to a same dimensional feature space by a fully-connected layer for better fusion. Our method secures effective and robust representation by strengthening local information, which further boosts the performance of the face representation. Extensive experiments on LFW and YTF benchmarks demonstrate the effectiveness and advantages of the method. Future work will focus on dividing feature maps instead of data into multiple patches to strengthen more local information. Meanwhile, a randomized cropping strategy will be explored to reduce redundancy created by multiple patches.

7. Acknowledgments

This work was supported by the National Science Foundation of China. The authors would like to thank the referees for their constructive suggestions.

[1] Krizhevsky, A., Sutskever, I., Hinton, G.E.: 'Imagenet classification with deep convolutional neural networks'. Advances in Neural Information Processing Systems, Lake Tahoe, USA, Jun. 2012, pp. 1097–1105

[2] He, K., Zhang, X., Ren, S., Sun, J.: 'Deep residual learning for image recognition'. IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, Nov. 2016, pp. 770–778

[3] Szegedy, C., Liu, W., Jia, Y., *et al.*: 'Going deeper with convolutions'. IEEE Conference on Computer Vision and Pattern Recognition, Boston, USA, Nov 2015, pp. 1–9

[4] Fu, R., Li, B., Gao, Y., Wang, P.: 'Fully automatic figure-ground segmentation algorithm based on deep convolutional neural network and grabcut', IET Image Processing, 2016, **10**, (12), pp. 937–942

[5] Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: 'Deepface: Closing the gap to human-level performance in face verification'. IEEE Conference on Computer Vision and Pattern Recognition, Columbus, USA, Nov. 2014, pp. 1701–1708

[6] Sun, Y., Wang, X., Tang, X.: 'Deep learning face representation from predicting 10,000 classes'. IEEE Conference on Computer Vision and Pattern Recognition, Columbus, USA, Nov. 2014, pp. 1891–1898

[7] Sun, Y., Wang, X., Tang, X.: 'Deep learning face representation by joint identification-verification'. Advances in Neural Information Processing Systems, Montreal, Canada, Jun. 2014, pp. 1988–1996

[8] Schroff, F., Kalenichenko, D., Philbin, J.: 'Facenet: A unified embedding for face recognition and clustering'. IEEE International Conference on Computer Vision Workshops, Santiago, Chile, Dec. 2015, pp. 815–823

[9] Su, Y., Shan, S., Chen, X., Gao, W.: 'Hierarchical ensemble of global and local classifiers for face recognition', IEEE Transactions on Image Processing, 2009, **18**, (8), pp. 1885–1896

[10] Chen, D., Cao, X., Wen, F., Sun, J.: 'Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification'. IEEE Conference on Computer Vision and Pattern Recognition, Portland, USA, Nov. 2013, pp. 3025–3032

[11] Soodeh, N., Majid, A., Wei, S.: 'Local gradient-based illumination invariant face recognition using local phase quantisation and multi-resolution local binary pattern fusion', IET Image Processing, 2015, **9**, (1), pp. 12–21

[12] Hu, G., Yang, Y., Yi, D., *et al.*: 'When face recognition meets with deep learning: An evaluation of convolutional neural networks for face recognition'. IEEE International Conference on Computer Vision Workshops, Santiago, Chile, Dec. 2015, pp. 142–150

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

[14] Forczmański, P., Łabędź, P.: 'Improving the recognition of occluded faces by means of two-dimensional orthogonal projection into local subspaces'. International Conference Image Analysis and Recognition, Niagara Falls, Canada, Jun. 2015, pp. 229–238

[15] Forczmański, P., Łabędź, P.: 'Recognition of occluded faces based on multi-subspace classification'. International Conference on Computer Information Systems and Industrial Management, Cracow, Poland, Sep. 2013, pp. 148–157

[16] Zhu, Y.L.: 'Sub-pattern non-negative matrix factorization based on random subspace for face recognition'. International Conference on Wavelet Analysis and Pattern Recognition, Beijing, China, Nov. 2007, pp. 1356–1360

[17] Hariri, W., Tabia, H., Farah, N., Benouareth, A., Declercq, D.: '3D face recognition using covariance based descriptors', Pattern Recognition Letters, 2016, **78**, pp. 1–7

[18] Tadmor, O., Wexler, Y., Rosenwein, T., Shalevshwartz, S., Shashua, A.: 'Learning a metric embedding for face recognition using the multi-batch method'. Advances in Neural Information Processing Systems, Barcelona, SPAIN, May 2016, pp. 1388–1389

[19] Wen, Y., Zhang, K., Li, Z., Qiao, Y.: 'A discriminative feature learning approach for deep face recognition'. European Conference on Computer Vision, Amsterdam, USA, Oct. 2016, pp. 499–515

[20] Ioffe, S., Szegedy, C.:'Batch normalization: Accelerating deep network training by reducing internal covariate shift'. International Conference on Machine Learning, Lille, France, Jul. 2015, pp. 448–456

[21] Long, M., Wang, J.: 'Learning transferable features with deep adaptation networks'. International Conference on Machine Learning, Lille, France, Jul. 2015, pp. 97–105

[22] Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E., 'Labeled faces in the wild: a database for studying face recognition in unconstrained environments' (University of Massachusetts Press, Oct. 2007), pp. 07–49

[23] Wolf, L., Hassner, T., Maoz.: 'Face recognition in unconstrained videos with matched background similarity'. IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, Jun. 2011, pp. 529–534

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

[25] Zhang, J., Shan, S., Kan, M., Chen, X.: 'Coarse-to-Fine Auto-Encoder Networks (CFAN) for Real-Time Face Alignment'. European Conference on Computer Vision, Zurich, Sep. 2014, pp. 1–16

[26] Jia, Y., Shelhamer, E., Donahue, J., *et al.*: 'Caffe: Convolutional architecture for fast feature embedding'. Proceedings of the 22nd ACM international conference on Multimedia, Orlando, FL, USA, Nov. 2014, pp. 675–678

[27] Bottou, L.: 'Stochastic Gradient Descent Tricks', in Heidelberg, Berlin: 'Neural Networks: Tricks of the Trade' (Springer Press, 2012), pp. 421–436

[28] Dahl, G.E., Sainath, T.N., Hinton, G.E.: 'Improving deep neural networks for LVCSR using rectified linear units and dropout'. IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC Canada, May 2013, pp. 8609–8613

[29] Nair, V., Hinton, G.E.: 'Rectified linear units improve restricted boltzmann machines'. International Conference on Machine Learning, Haifa, Israel, Jun. 2010, pp. 807–814

[30] Parkhi, O.M., Vedaldi, A., Zisserman, A.: 'Deep face recognition'. Machine Vision Conference, Swansea, British, Sep. 2015, pp. 41.1–41.12

[31] Tadmor, O., Wexler, Y., Rosenwein, T., Shalevshwartz, S., Shashua, A.: 'Learning a metric embedding for face recognition using the multibatch method'. Advances in Neural Information Processing Systems, Barcelona, Spain, Dec. 2016, pp. 1388–1389

[32] Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: 'Web-scale training for face identification'. IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, Jun. 2015, pp. 2746–2754

[33] Sun, Y., Wang, X., Tang, X.: 'Deeply learned face representations are sparse, selective, and robust'. IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, Jun. 2015, pp. 2892–2900