Neurocomputing 314 (2018) 94–100

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Split-Net: Improving face recognition in one forwarding operation

Ge Wen*, Yi Mao, Deng Cai, Xiaofei He

The State Key Lab of CAD&CG, Zhejiang University, No. 388 Yu Hang Tang Road, Hangzhou 310058, China

ARTICLE INFO

Article history:
Received 22 January 2018
Revised 2 June 2018
Accepted 7 June 2018
Available online 28 June 2018

Communicated by Dr. Ran He

Keywords:
Deep face representation
Region based models
Feature fusion

ABSTRACT

The performance of face recognition has been improved a lot owing to deep Convolutional Neural Network (CNN) recently. Because of the semantic structure of face images, local part as well as global shape is informative for learning robust deep face feature representation. In order to simultaneously exploit global and local information, existing deep learning methods for face recognition tend to train multiple CNN models and combine different features based on various local image patches, which requires multiple forwarding operations for each testing image and introduces much more computation as well as running time. In this paper, we aim at improving face recognition in only one forwarding operation by simultaneously exploiting global and local information in one model. To address this problem, we propose a unified end-to-end framework, named as Split-Net, which splits selective intermediate feature maps into several branches instead of cropping on original images. Experimental results demonstrate that our approach can effectively improve the accuracy of face recognition with less computation increased. Specifically, we increase the accuracy by one percent on LFW under standard protocol and reduce the error by 50% under BLUFR protocol. The performance of Split-Net matches state-of-the-arts with smaller training set and less computation finally.

1. Introduction

Face recognition has attracted lots of attention in the domain of computer vision since decades ago because of its various applications in the area of biometrics, access control, surveillance, *etc*. There are also many works focused on face sketch recognition [16,26], which have wide applications ranging from digital entertainments to law enforcements. And more and more works of face sketch synthesis have been proposed [24,25,27]. Based on the definitions, face recognition can be divided into two specific subtopics, face identification and face verification. Face identification is to assign a label or a name from a candidate set to a face image, and face verification is to tell whether two face images belong to the same person or not. For face identification, it should be ensured that the identities in training set and testing set are identical while face verification is much more flexible. Moreover, given a gallery set, face identification problem can also be solved with face verification by repeated one-vs-one comparison.

Since the huge success was achieved by convolutional neural network [12] in ImageNet [3] object classification competition in 2012 with a large margin advanced to traditional methods, convolutional neural networks have been adopted to more and more computer vision problems, including face recognition [22]. In the past years, more and more face recognition algorithms based on CNN [14,15,19,21] have been proposed whose accuracies are superior to that of human or even higher than 99% on LFW [10], a widely used face verification benchmark. Almost all of them train several models of the same architecture or extract multiple features based on various face patches for ensemble so as to improve the performance further by simultaneously taking advantage of global and local facial information as illustrated in Fig. 1. However, there exist many limits in such an ensemble way. Multiple forwarding operations either on several models [14,21] or one model [15] are required for each image to extract features during testing, which will introduce linearly increased computation or running time and slow down the recognition process. In addition, the improvement is usually not linearly with the number of models used and may get saturated. Take DeepID2 [21] as an example. It trains 200 CNN models based on different face patches and the corresponding horizontally flipped counterparts. At testing time, up to 25 highly selective features are picked out of the total 400 feature vectors to be concatenated into a new one for evaluation. Table 1 shows the face verification accuracy and corresponding extraction time reported in [21] on LFW of concatenating different numbers of features. We can find that the improvement 1.85% from using only one patch to integrating two patches contributes the most to the improvement of accuracy, which is even

* Corresponding author.
E-mail addresses: zjuwenge@gmail.com (G. Wen), yimao.zju@gmail.com (Y. Mao), dengcai@cad.zju.edu.cn (D. Cai), xiaofeihe@cad.zju.edu.cn (X. He).

https://doi.org/10.1016/j.neucom.2018.06.030
0925-2312/

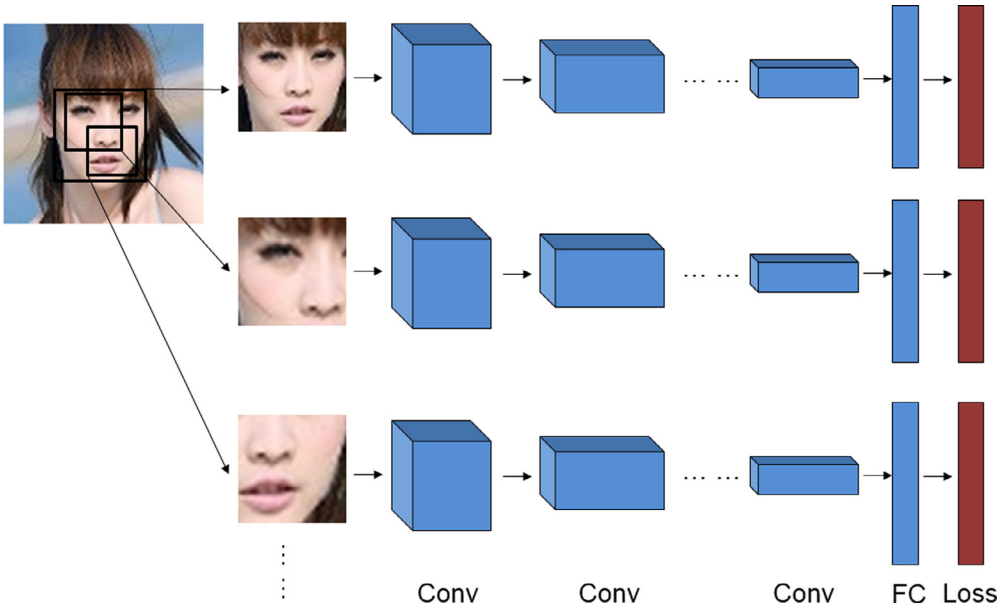

Fig. 1. Overview of deep CNN model using multi-patch.

Table 1
Accuracies on LFW of DeepID2.

#Patches	1	2	4	8	16	25
Accuracy (%)	95.43	97.28	97.75	98.55	98.93	98.97
Time (ms)	1.7	3.4	6.1	11	23	35

larger than that from 2 patches to 25 patches, 1.69%. It is also difficult and requires much effort to select patches as effective as possible. The performance may even declines using features of inappropriate patches.

To address these issues mentioned above, we aim at improving face recognition performance with forwarding each image only once during testing by simultaneously exploiting global and local facial information in one global patch based model. Different from existing methods cropping on the original images, we propose a unified end-to-end framework, named as Split-Net (S-Net), which splits the feature maps of intermediate layers to several branches. Based on the CNN architecture, the size of corresponding receptive filed in the original images for each layer can be calculated. Appropriate feature maps are then chosen to operate on in order to exploit local information and meanwhile preserve global information in lower layers. Performance improvement is achieved using this framework with much less computation introduced than that of multi-patch setting. With S-Net, we enhance basic model one percent on LFW benchmark under standard protocol and reduce the error by 50% under BLUFR protocol. The performance of S-Net matches state-of-the-arts with smaller training set and less computation finally.

Our contributions can be summarized as follows:

- A new unified end-to-end framework, Split-Net, is proposed, which can simultaneously exploit global and local information in one model and learn a better representation with much less increase of computation. Not only for face recognition, it can also be applied to other computer vision applications based on CNN and even those already proposed CNN models.
- Comparative experiments on face recognition are conducted to demonstrate the superior performance of S-Net to original model and that cropping on images.
- Performance matched to state-of-the-arts is achieved using S-Net on LFW with smaller training set and less computation.

2. Related work

2.1. Deep face representation

DeepFace [22] is the first method that proposed to use deep learning method for face recognition problem. Owing to a large private face dataset and an effective facial alignment system based on explicit 3D modeling of faces, 97.35% accuracy is achieved on LFW using one CNN model trained with softmax loss.

After that, lots of deep methods have been proposed for face recognition. Part of them tend to improve the performance of deep learning models by applying different loss functions other than the traditional softmax loss for learning more discriminative features. DeepID2 [21] innovatively combines two loss functions together to train deep convolutional neural networks. Except for softmax loss, contrastive loss is used which takes two samples as inputs and is aimed at minimizing the distance between samples from the same class and maximizing it between samples from different classes until a margin is reached. By combining up to 25 models, 98.97% is achieved on LFW. FaceNet [19] proposes to use triplet loss for face recognition. A triplet consists of three images, an anchor image, a positive image and a negative image. The goal of triplet loss is to maximize the difference between two distances until it is larger than a margin. One is the distance between the anchor and the positive, and the other one is the distance between the anchor and the negative. Carefully hard negative mining strategy is employed in it for choosing more representative samples. Based on only one large inception model, it achieves 99.63% accuracy on LFW without using any local patch. But it is trained with an extremely huge private face dataset with nearly 260 million images, which is 500 times larger compared to ours.

Some methods share the intuition that metric learning can boost the performance and at the same time reduce the feature to a low dimension. Baidu [14] and VGG-Face [15] are two representatives. Both of them train CNN model guided by a face classifier based on softmax loss at first, and then fine-tune it by learning a face embedding with triplet loss that was used in [19]. Similar to almost all the other methods, Liu et al. [14] trains up to 9 models based on various face patches for ensemble and achieves an accuracy of 99.68% on LFW. Parkhi et al. [15] reveals that the widely used alignment operation on training data does not provide any

promotion for face recognition while doing it during testing helps. Different from Liu et al. [14] and many other methods, only one model is trained in [15]. But five patches are cropped from the four corners and the center with horizontal flip based on images of three scales during testing time. The average of these 30 features is taken to get an accuracy of 99.13% on LFW. Ding and Tao [4] also hold the idea that multimodal information and deep embedding method are helpful for face recognition. In their method, two global patches and six local patches are used to train 8 convolutional neural networks in total with two different kinds of models. Different from Liu et al. [14] and Parkhi et al. [15] using triplet loss for metric learning, all the features of images and flipped version from the 8 models are concatenated to train an auto-encoder for metric learning as well as dimension reducing in [4]. With the same training dataset to ours, 98.43% accuracy was achieved on LFW using one model by Ding and Tao [4] and 99.02% using eight models.

There also exist many other methods using multitask learning to assist the learning of face recognition. Ranjan et al. [17] combines various face related tasks together in one single deep convolutional neural network for simultaneous learning, including face detection, face alignment, pose estimation, gender recognition, smile detection, age estimation and face recognition. Unfortunately, the performance on LFW is not reported by this method.

2.2. Region based models

Receptive field is not a new topic for deep learning methods. Local weight sharing proposed in [9] is adopted by Sun et al. [21] to learn kernels of different weights on different facial regions. But different from ours, the information from different regions is then combined together in one feature map. Based on the pioneering object detection framework RCNN [6], Fast-RCNN [5] proposes to forward each image only once to extract the features of proposals by mapping the positions of proposal's local region on original image to the last feature maps according to the sizes of receptive fields. To get rid of the time-consuming selective search operation, region proposal network is proposed in Faster-RCNN [18] to generate proposals based on intermediate feature maps with the same idea.

The most related work to ours is the local adversarial loss proposed in [20]. Based on the intuition that any local patch in a generated image should have similar statistics to that of a real image, local adversarial loss is proposed by carefully designing the CNN architecture so that a probability map with every entry's receptive field in the original image limited to only a local region was generated. The difference between this work and ours is that we propose a framework of splitting intermediate feature maps to limit the size of receptive fields instead of carefully designing the CNN architecture so that our method can be applied to any existing CNN architecture proposed before. And the task of ours is more complicated compared to a two-way, true or generated, classification problem in [20].

3. Method

3.1. S-Net framework

The core motivation of our proposed S-Net framework is to reduce the sizes of corresponding receptive fields in original image of each layer so that the network branches after the operated layer can focus more on local patches and exploit local information meanwhile still preserving global information in lower layers. According to the ratio of the receptive field size on original image for each layer, an appropriate layer can be selected for the balance between global information and local information. Feature maps of the selected layer are then split uniformly without overlap horizontally and vertically to $N \times N$ parts. Fig. 2 shows the operation of the proposed S-Net framework when $N = 2$. Supposing a feature map with height H, width W and channel C, $H \times W \times C$ for abbreviation, is chosen to be split, four feature maps of size $H/2 \times W/2 \times C$ will be produced, whose receptive fields in original images are nearly a quarter of the original $H \times W \times C$ feature maps' as the four images showed on each branch in Fig. 2. For all the layers after the split one, the structures are kept the same as that in original model including the loss layer and the weights are not shared between each other for learning more robust local features independently. This framework is then trained in an end-to-end manner guided by four loss functions. It should be noticed that although the number of parameters is nearly four times compared to the original model, the computation are only two times with the lower layers being kept unchanged.

As the sizes of the $N \times N$ split feature maps are smaller than that of the original one, which cannot be directly attached to the successive layers in the original model, an additional size matching layer should be added to keep the final dimension consistent. There are several ways to solve this. Interpolation or adding an additional deconvolutional [29] layer are two available choices [8]. Through splitting and size matching, four features which have the same dimension as original model will be produced at the penultimate layer. During testing time, the four features can be evaluated separately or after some kinds of post-processing, such as maximizing, averaging or concatenating. The effects of these post-processing methods will be studied in the experiment.

3.2. Architecture

In this section, we will describe the CNN model used in this paper and illustrate S-Net framework taking this model as an example.

We adopt the network used in [28], denoted as *casia* model, with some modifications as our *basic* CNN model. The *casia* model consists of 10 convolutional layers, 5 pooling layers, and no fully connected layer except the classification layer *fc10575* used in the softmax loss. Kernels with small size 3×3 are used in all the convolutional layers for learning more discriminative features. The sizes of feature maps keep unchanged before and after every convolutional layer with stride equal to 1, while the pooling layers with kernel size 2×2 and stride 2 reduce the height/width of feature maps by half. To learn more robust features, we replace the global pooling layer *pool5* by a global convolutional layer named *conv5* with the same number of channels. The ReLU activation function is modified to the PReLU [7] to avoid gradient vanishing. Similar to Sun et al. [21], Yi et al. [28] combines softmax loss and contrastive loss. In our model only softmax loss is used for simplicity and avoiding trivial hard negative mining. The output of layer *conv5* is extracted as feature for evaluation. Because of the modified *conv5* layer, the total number of parameters of this modified model is a little larger than that of *casia* model, as well as the number of computation. Table 2 shows the modified *casia* model with PReLU omitted.

The last column in Table 2 shows the corresponding receptive filed size of each sub feature map if one layer is chosen to be split to 2×2 parts. For example, if feature map splitting is operated on layer *conv22*, as the output feature map size of layer *conv22* is $50 \times 50 \times 128$, four feature maps with size $25 \times 25 \times 128$ will be produced. And it can be figured out that each $25 \times 25 \times 128$ feature map receives the information of a region with size 56×56 on original image. With the layer becoming deeper, the corresponding size of receptive field becomes larger accordingly. As the pooling layer reduces the height and width of feature maps by a half, we

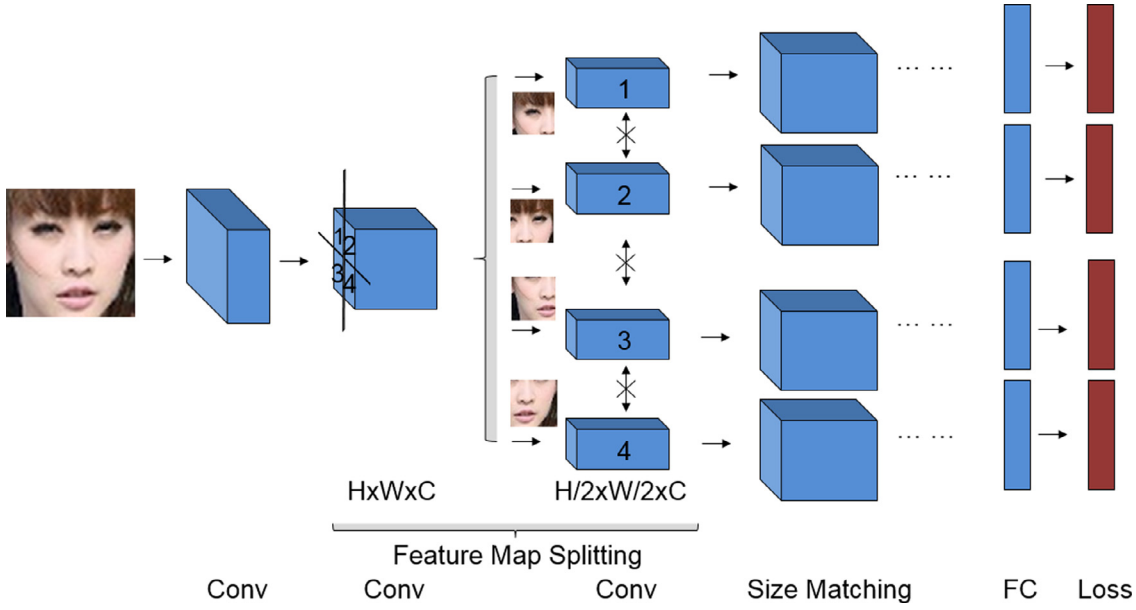

Fig. 2. Framework of S-Net.

Table 2
Modified *casia* model. The channel numbers of *basic2x* are shown in parentheses.

Name	Type	Filter size/stride	Output size	Receptive field size
conv11	convolution	3 × 3 / 1	100 × 100 × 32	51 × 51
conv12	convolution	3 × 3 / 1	100 × 100 × 64	52 × 52
pool1	max pooling	2 × 2 / 2	50 × 50 × 64	52 × 52
conv21	convolution	3 × 3 / 1	50 × 50 × 64	54 × 54
conv22	convolution	3 × 3 / 1	50 × 50 × 128(256)	56 × 56
pool2	max pooling	2 × 2 / 2	25 × 25 × 128(256)	58 × 58
conv31	convolution	3 × 3 / 1	25 × 25 × 96(192)	62 × 62
conv32	convolution	3 × 3 / 1	25 × 25 × 192(384)	66 × 66
pool3	max pooling	2 × 2 / 2	13 × 13 × 192(384)	70 × 70
conv41	convolution	3 × 3 / 1	13 × 13 × 128(256)	78 × 78
conv42	convolution	3 × 3 / 1	13 × 13 × 256(512)	86 × 86
pool4	max pooling	2 × 2 / 2	7 × 7 × 256(512)	94 × 94
conv51	convolution	3 × 3 / 1	7 × 7 × 160(320)	100 × 100
conv52	convolution	3 × 3 / 1	7 × 7 × 320(640)	100 × 100
conv5	convolution	7 × 7 / 1	1 × 1 × 320	100 × 100
dp	dropout(40%)		1 × 1 × 320	
fc10575	fully connected		10,575	
loss	softmax		10,575	

can simply remove *pool2* for size matching and directly link the split four feature maps to layer *conv31* if splitting on *conv22*. Interpolation or adding an additional deconvolutional layer can also be used here. The layers after *pool2* are then appended to each of the four branches. Four features with the same dimension 320 as layer *conv5* will be produced. The total number of computation is only two times of that without splitting, compared to four times larger if training four models with multi-patch setting.

4. Experiment

4.1. Settings

We use CASIA-WebFace dataset [28] for our CNN training and conduct all the experiments on LFW under both standard protocol and BLUFR [13] protocol. CASIA-WebFace dataset is one of the largest public datasets for face recognition. It consists of 494,414 face images crawled from IMDb movie website belonging to 10,575 actors/actresses. The identities in CASIA-WebFace and those in LFW are ensured to have no intersection. We use our own algorithms to detect the face bounding boxes and corresponding landmarks in all the images of CASIA-WebFace and LFW. Different from that a fixed size 100 × 100 is used as input in [28], all the faces are aligned by three points, left eye, right eye and the center of mouth, to match a template of size 112 × 112 in our model. Before fed into the CNN model, the 112 × 112 image is cropped to 100 × 100 randomly in training while in testing the center crop is used. For both training and testing procedure, the values of pixels in image are divided by 256 to accelerate the training and avoid weight explosion.

We use the popular Caffe [11] framework to implement our method. The S-Net framework can be easily implemented in Caffe using *Slice* layer and *group* parameter in *Convolution* layer. The code will be released public on GitHub. The model is initialized with MSRA method proposed by He et al. [7]. Stochastic gradient descent method with momentum is used to train the model. The decay and momentum are set to 0.0005 and 0.9, respectively, as common. The learning rate is 0.01 at beginning and degrades to 0.00001 gradually with the model training. All the models are trained on four NVIDIA K20 GPUs.

After training, all features of the aligned images from LFW are extracted. For each image, only the original image is used without horizontally flipped one or any local patch. That is to say, every

Table 3
Accuracies(%) on LFW under standard protocol of different settings.

Model	Time (ms)	1	2	3	4	Max	Avg	Concat
basic	1.82	–	–	–	–	–	–	98.02
basic2x	3.12	–	–	–	–	–	–	98.10
image(56)	7.27	95.78	95.62	95.65	96.33	93.93	98.17	98.55
image(50)	7.25	93.97	94.47	92.87	93.60	92.12	96.93	98.13
fusion of *baisc* and *image(56)*	9.09	–	–	–	–	–	–	98.76
S-Net(conv52)	1.61	97.73	97.83	97.67	97.90	92.48	97.17	97.92
S-Net(conv22-pool2)	3.80	95.92	95.85	95.95	96.32	93.73	97.97	**99.02**
S-Net(conv22+bilinear interpolation)	5.61	95.86	95.93	95.87	96.35	93.70	97.91	**99.11**
S-Net(conv22+deconv)	5.62	96.01	95.87	95.90	96.24	93.78	97.88	**99.08**

image is fed into CNN model only once for feature extraction. All the features are reduced to a dimension of 180 by PCA method and then are normalized to have a unit norm. L2 distance is used as the metric for final performance evaluation.

For the standard protocol of LFW benchmark, we follow the unrestricted with labeled outside data protocol. Under this protocol, ten groups with 300 positive pairs and 300 negative pairs in each one are provided. Results are reported using a cross validation strategy by repeatedly training, or selecting threshold in other words, on nine groups and testing on the remaining one. The final result should be an average of the results on all of these ten splits.

In order to better illustrate the effect of S-Net, three layers are chosen to be split for comparison, the image layer, the *conv22* layer (*S-Net(conv22)*) and the *conv52* layer (*S-Net(conv52)*). As $N = 3$ will introduce nearly 9 times more computation, we set $N = 2$ for the trade-off between performance and efficiency. Moreover, because the face images are aligned before feeding into the networks, the cropped four patches are approximately based on four landmarks, center of left eye, center of right eye and two mouth corners. For cropping on image, all the patches are resized to 112×112 before being fed into *basic* model. For splitting on *conv52*, each *conv5* layer of the four branches is connected to a patch in the corner with size 4×4 of the feature maps in layer *conv52*, thus 1 row/column of pixels is overlapped between patches. In addition, we also implement a *basic2x* network. The number of channels from layer *conv*22 to layer *conv*52 are doubled so that the *basic2x* network shares nearly the same computation to *S-Net(conv22)*. The differences between *basic* and *basic2x* network are represented in Tabel 2.

4.2. Ablation study

Table 3 shows the results under standard protocol of different settings. The four *1, 2, 3, 4* columns represent the results with the features of four branches evaluated individually as marked in Fig. 2. The columns of *avg* and *max* mean average or maximum is taken before testing respectively. Column *concat* shows the result of concatenating the four features to a whole one with the dimension 1280. The running time for one image is provided in the second column. Note that for *basic* and *basic2x* model, there is only one feature to be evaluated for each image. We fill the result in column *concat* for clarity.

4.2.1. S-Net of splitting on layer conv52

Each individual feature of the four branches can achieve about 98% with small differences between each other. This is within our expectation as each one 4×4 region of the *conv52* layer acquires the whole information of the original image. It is rich enough for them to learn a representative feature for recognition. Concatenation or other post-processing operations then become redundant and provide little additional information. As the loss function is applied on each *conv5* layer individually, the results of taking average or maximum even degrade and only a little improvement is achieved through concatenation to get a comparable result to the baseline. All the results are lower than *basic* model 98.02%, which may be because of the loss of necessary information when decreasing the number of connections between layer *conv52* and *conv5*. Splitting on layer *conv52* introduces little computation increase and the running time is nearly the same as *basic* model.

Table 4
Results(%) on LFW under BLUFR protocol of different settings.

model	VR@FAR=0.1%	DIR@FAR=1% Rank=1
basic	88.01	57.53
basic2x	88.50	60.44
image(56)	92.64	68.12
image(50)	87.66	57.65
fusion of *baisc* and *image(56)*	92.86	70.23
S-Net(conv52)	85.02	50.88
S-Net(conv22-pool2)	**93.89**	**72.40**

4.2.2. S-Net of splitting on layer conv22

In accordance with our expectation, using each feature of all the four branches we can only get very poor performances, about 96%, which is inferior to *basic* model with a large margin. The reason is that each one of the four branches only receives part information of the original image, 56×56 of 100×100 and lacks necessary global information. Taking the maximum leads to a large drop as it cannot make full use of all the information contained in each one and lose some necessary information. Taking the average achieves a comparable result to *basic* model. Via concatenation an accuracy over 99% is achieved, which advances *basic* model by one percent and is even superior to *basic2x* model with similar running time. This is due to the fact that by splitting intermediate feature maps, the size of receptive field in original image for each branch is limited to a local region and then each one can be focused more on learning detailed local information and meanwhile preserves global information in lower layers. Via concatenating these local information are combined together thus details of the whole image are exploited. Both *S-Net(conv22+bilinear interpolation)* and *S-Net(conv22+deconv)* are superior to *S-Net(conv22-pool2)* because of the added up-sampling layer which will increase the model complexity and representation power. However, the running time is much larger, nearly 1.5 times during testing and 2.3 or 4.6 times respectively during training compared to *S-Net(conv22-pool2)*. For the consideration of efficiency, we choose *S-Net(conv22-pool2)* as representative in following experiments.

4.2.3. Cropping on image

Each one quarter 25×25 feature map of the *conv22* layer gets a receptive field of size 56×56, so for cropping on image, two different cropping operations are experimented. One is to crop the image to four 56×56 parts with overlap and the other one is four 50×50 parts without overlap, denoted as *image(56)* and *image(50)*, respectively, in Table 3.

Table 5
Accuracies(%) on LFW compared with other state-of-the-arts. P stands for public and R stands for private.

Method	Input size	#Models #Patches	Computation	Dataset	Metric	Accuracy
S-Net(conv22)	100×100	1/1	1692M	490K/P	L2	99.02
CASIA [28]	100×100	1/1	775M	490K/P	JB	97.30
MM-DFR [4]	165×120	1/2	4.70B (2354M)	490K/P	JB	98.43
	165×120	8/2	41.86B (2616M)	490K/P	JB	99.02
DeepID2 [21]	55×47	25/1	200M (8M)	200K/P	JB	98.97
FaceNet [1,19]	220×220	1/1	1610.5M	260M/R	L2	99.63
	96×96	1/1	218M	500K/P	L2	92.92
Baidu [14]	Unknown	1/1	Unknown	1.2M/R	L2	99.13
	Unknown	7/1	Unknown	1.2M/R	L2	99.68
VGG-Face [15]	224×224	1/30	325.26B (10842M)	2.6M/P	L2	98.95
Fusion [23]	152×152	5/1	2506M (501M)	500M/R	SVM	98.37

Both of the two cropping methods represent an inferior performance compared to *S-Net(conv22)* even with much more running time. We contribute this to it that in *S-Net(conv22)*, the global information is exploited in lower layers and can guide the learning of the four branches to some extent. While for cropping on image layer, because of that the four local image patches are generated uniformly, the four models cannot capture sufficient global facial information to learn a robust feature during training and the improvement achieved via concatenation is less than *S-Net(conv22)*. Cropping on image with size 50×50 is worse than that on 56×56 because of less information contained in the images.

In Table 3, we also represent the result of fusing *basic* and *image(56)*. The result is a little inferior to that of *S-Net(conv22-pool2)*, which shows the effectiveness of joint training with local and global information in one model proposed by S-Net.

4.2.4. BLUFR protocol

There are only 6000 testing pairs for the standard protocol of LFW, which are not enough for evaluating face recognition at low FARs. Therefore, Liao et al. [13] developed a more challenging benchmark BLUFR to make full use of all the images in LFW. It contains both verification and open-set identification scenarios with a focus at low FARs. We report the result by the standard benchmark toolkit [13]. As can be seen in Table 4, the results under BLUFR protocol are consistent with that under standard protocol. *S-Net(conv22)* achieves an giant improvement and reduce the error by 50% in terms of VR compared to *basic* and *basic2x*, which verifies the effectiveness of S-Net.

4.3. Compared with state-of-the-arts

Table 5 shows the performance on LFW of our framework and the state-of-the-arts as well as the computation needed for each testing image. For those methods using multi-patch, computation of one forwarding operation are provided in brackets. With the same dataset, our framework reduces the error of CASIA by nearly three times, and achieves a better or comparable result to MM-DFR with much less computation. The computation of VGG-Face is fairly large because of a big input size and deep architecture. DeepID2 achieves a little inferior accuracy to ours with low computation. However, the 25 features are selected from 400 features of 200 models, whose total training computation is the same to ours without mentioning the effort for selecting. Moreover, Joint Bayesian (JB) [2] is used in it as metric for boosting while our method uses L2 distance directly. It may be not easy to compare S-Net with FaceNet or Baidu which uses a much larger dataset directly. However, it should be noticed that OpenFace [1] implemented the popular FaceNet system with a small model based on a similar dataset to ours and only achieved 92.92% on LFW. In addition, the same NN2 model as [19] trained with this 500K dataset failed to converge and the performance is worse, which reveals that the giant training dataset is crucial for training a well performed FaceNet system. Our proposed S-Net framework achieves a comparable performance to these state-of-the-arts with only one forwarding operation and smaller training set.

5. Conclusion

In this paper, we propose a unified end-to-end framework Split-Net (S-Net), which is aimed at improving the performance of face recognition in only one forwarding operation by simultaneously exploiting local and global information in one convolutional neural network. Instead of cropping on original images which introduces much more computation and running time, S-Net splits the intermediate feature maps with less computation increased. S-Net can be applied to those fine-grained recognition problem besides face recognition where each object shares the same or similar structure. With S-Net, we enhance original model one percent on LFW benchmark under standard protocol and reduce the error by 50% under BLUFR protocol. We hope our work could provide a new insight for others on CNN model especially on simultaneously exploiting local and global information in one model.

References

[1] B. Amos, B. Ludwiczuk, M. Satyanarayanan, OpenFace: A General-purpose Face Recognition Library with Mobile Applications, Technical Report, CMU-CS-16-118, CMU School of Computer Science, 2016.
[2] D. Chen, X. Cao, L. Wang, F. Wen, J. Sun, Bayesian face revisited: a joint formulation, in: Proceedings of European Conference on Computer Vision, Springer, 2012, pp. 566–579.
[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, IEEE, 2009, pp. 248–255.
[4] C. Ding, D. Tao, Robust face recognition via multimodal deep face representation, IEEE Trans. Multimed. 17 (11) (2015) 2049–2058.
[5] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
[6] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
[7] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1026–1034.
[8] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[9] G.B. Huang, H. Lee, E. Learned-Miller, Learning hierarchical representations for face verification with convolutional deep belief networks, in: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2012, pp. 2518–2525.
[10] G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, Technical Report, University of Massachusetts, Amherst, 2007. Technical Report 07-49.
[11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, in: Proceedings of the 22nd ACM International Conference on Multimedia, ACM, 2014, pp. 675–678.

[12] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[13] S. Liao, Z. Lei, D. Yi, S.Z. Li, A benchmark study of large-scale unconstrained face recognition, in: Proceedings of 2014 IEEE International Joint Conference on Biometrics, IJCB, IEEE, 2014, pp. 1–8.
[14] J. Liu, Y. Deng, C. Huang, Targeting Ultimate Accuracy: Face Recognition via Deep Embedding, (2015) arXiv preprint arXiv:1506.07310.
[15] O.M. Parkhi, A. Vedaldi, A. Zisserman, Deep face recognition, in: Proceedings of British Machine Vision Conference, 2015.
[16] C. Peng, X. Gao, N. Wang, J. Li, Sparse Graphical Representation Based Discriminant Analysis for Heterogeneous Face Recognition, 2016. arXiv preprint arXiv: 1607.00137.
[17] R. Ranjan, S. Sankaranarayanan, C.D. Castillo, R. Chellappa, An All-in-One Convolutional Neural Network for Face Analysis, 2016. arXiv preprint arXiv: 1611.00851.
[18] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[19] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: a unified embedding for face recognition and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 815–823.
[20] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, R. Webb, Learning from Simulated and Unsupervised Images thzough Adversarial Training, 2016. arXiv preprint arXiv: 1612.07828.
[21] Y. Sun, Y. Chen, X. Wang, X. Tang, Deep learning face representation by joint identification-verification, in: Advances in Neural Information Processing Systems, 2014, pp. 1988–1996.
[22] Y. Taigman, M. Yang, M. Ranzato, 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, 2014, pp. 1701–1708.
[23] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Web-scale training for face identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2746–2754.
[24] N. Wang, X. Gao, J. Li, Random sampling for fast face sketch synthesis, Pattern Recognit. 76 (2018) 215–227.
[25] N. Wang, X. Gao, L. Sun, J. Li, Bayesian face sketch synthesis, IEEE Trans. Image Process. 26 (3) (2017) 1264–1274.
[26] N. Wang, J. Li, L. Sun, B. Song, X. Gao, Training-free Synthesized Face Sketch Recognition Using Image Quality Assessment Metrics, 2016. arXiv preprint arXiv: 1603.07823.
[27] N. Wang, D. Tao, X. Gao, X. Li, J. Li, A comprehensive survey to face hallucination, Int. J. Comput. Vis. 106 (1) (2014) 9–30.
[28] D. Yi, Z. Lei, S. Liao, S.Z. Li, Learning Face Representation from Scratch, 2014. arXiv preprint arXiv: 1411.7923.
[29] M.D. Zeiler, G.W. Taylor, R. Fergus, Adaptive deconvolutional networks for mid and high level feature learning, in: Proceedings of 2011 IEEE International Conference on Computer Vision, ICCV, IEEE, 2011, pp. 2018–2025.

Ge Wen received the B.S. degree in Computer Science from Zhejiang University of China in 2014. He is currently pursuing the Ph.D. degree at the State Key Laboratory of CAD&CG, Zhejiang University. His research interests include computer vision and machine learning.

Yi Mao is a Software Engineer in JingDong at Beijing, China. He received the Master degree in computer science from Zhejiang University in 2017. His research interests include computer vision and data mining.

Deng Cai is a Professor in the State Key Lab of CAD&CG, College of Computer Science at Zhejiang University, China. He received the Ph.D. degree in computer science from University of Illinois at Urbana Champaign in 2009. His research interests include machine learning, data mining, computer vision and information retrieval.

Xiaofei He received the B.S. degree in Computer Science from Zhejiang University, China, in 2000 and the Ph.D. degree in Computer Science from the University of Chicago, in 2005. He is a Professor in the State Key Lab of CAD&CG at Zhejiang University, China. Prior to joining Zhejiang University, he was a Research Scientist at Yahoo! Research Labs, Burbank, CA. His research interests include machine learning, information retrieval, and computer vision.