Front.Comput.Sci.
DOI

RESEARCH ARTICLE

VIPLFaceNet: An Open Source Deep Face Recognition SDK

Xin Liu[1,2], **Meina Kan**[1,2], **Wanglong Wu**[1,2], **Shiguang Shan** (✉)[1,2], **Xilin Chen**[1,2]

1 Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, 100190, China
2 University of Chinese Academy of Sciences, Beijing 100049, China

Abstract Robust face representation is imperative to highly accurate face recognition. In this work, we propose an open source face recognition method with deep representation named as VIPLFaceNet, which is a 10-layer deep convolutional neural network with 7 convolutional layers and 3 fully-connected layers. Compared with the well-known AlexNet, our VIPLFaceNet takes only 20% training time and 60% testing time, but achieves 40% drop in error rate on the real-world face recognition benchmark LFW. Our VIPLFaceNet achieves 98.60% mean accuracy on LFW using one single network. An open-source C++ SDK based on VIPLFaceNet is released under BSD license. The SDK takes about 150ms to process one face image in a single thread on an i7 desktop CPU. VIPLFaceNet provides a state-of-the-art start point for both academic and industrial face recognition applications.

Keywords Deep Learning, Face Recognition, Open Source, VIPLFaceNet.

1 Introduction

Face recognition, as one of the typical problems in computer vision and machine learning, plays an important role in many applications, such as video surveillance, access control, computer-human interface and mobile entertainments [1]. Generally speaking, a conventional face recognition system consists of four modules, face detection, face alignment, face representation and identity classification. In this pipeline, the key component for accurate face recognition is the third module, i.e. extracting the representation of an input face, which this paper mainly focuses on.

The main challenges of face representation lie in the small inter-person appearance difference caused by similar facial configurations, as well as the large intra-person appearance variations due to large intrinsic variations and diverse extrinsic imaging factors, such as head pose, expression, aging, and illumination. In the past decades, face representation is mostly based on hand-crafted local descriptors [2–8] and shallow learning-based representation models [9–14]. As the development of deep learning technology, it becomes a more potent approach for face representation learning, especially in the real-word scenarios. Compared with the previous hand-crafted routine, deep face representation is learned in a data-driven style which can guarantee better performance as validated in [15–19]. Taking the de-facto real-world face recognition benchmark LFW as an example, hand-crafted descriptor recorded 95.17% set by high-dimensional LBP [4], while 99.63% accuracy achieved by the latest deep FaceNet in [19].

In spite of many decades of research and development on face recognition, few open-source face recognition systems are publicly available yet. An open-source SDK with high accuracy in general scenarios is in great need for both academic research and industrial applications. So in this work, we meet this requirement and propose a deep face recognition model named as VIPLFaceNet, which is released as a BSD-license open source software with

Received month 02, 2016; accepted month 08, 2016

VIPLFaceNet is a powerful deep network for face representation with ten layers including 7 convolutional layers and 3 fully-connected layers. As a BSD-license open source software, VIPLFaceNet allows both academic research and industrial face recognition applications in different software and hardware platforms for free.

The contributions of this paper are summarized as follows:

1. We propose and release an open source deep face recognition model, VIPLFaceNet, with high-accuracy and low computational cost, which is a 10-layer deep convolutional neural network that achieves 98.60% mean accuracy on the real-world face recognition benchmark LFW.

2. We investigate the network architecture design and simplification. By careful design, VIPLFaceNet reduces 40% computation cost and cuts down 40% error rate on LFW compared with the AlexNet [20].

3. The VIPLFaceNet SDK code is written in pure C++ code under the BSD license. It is free and easy to be deployed in various software or hardware platforms for both academic research and industrial face recognition applications.

In summary, VIPLFaceNet is an open source deep face recognition SDK with high accuracy in general scenarios, which is built for facilitating the academic and industrial application of various real-world face recognition tasks. The rest of this paper is organized as follows. Section 2 presents the related works on face representation learning and introduce the face recognition benchmarks. Section 3 presents the network architecture design and technical details of our VIPLFaceNet. Section 4 conducts the experimental evaluation with comprehensive discussions and section 5 concludes this paper.

2 Related Works

In this section, we give a brief review of the related works on face representation learning. Moreover, we give a brief review of the face recognition benchmarks and discuss the performance evolution on the de-facto real-world face recognition benchmark LFW.

2.1 Face Representation before Deep Learning

In the past decades, numerous hand-crafted local features were proposed for face representation, e.g. Gabor wavelets [2], Local Binary Pattern (LBP) [3] and its high dimensional variant [4], Scale Invariant Feature Transform (SIFT) [8], Histogram of Oriented Gradients (HOG) [5], patterns of oriented edge magnitudes (POEM) [6], Local Quantized Pattern (LQP) [7] etc. However, designing an effective local descriptor demands considerable domain specific knowledge and a great deal of efforts.

Besides the hand-crafted local features, learning-based representation is also popular and reports promising accuracy. In [9] and [10], filters are learned to maximize the discriminative power for face recognition. In [21], faces are represented from its responses to many pre-trained object filters. In [11], [12], [13] and [22], codebook learning technologies are utilized for robust face representation. More recently, faces are represented with mid-level or high-level semantic information. For instance, the attributes and simile classifier [23] represent faces by the mid-level face attributes and so-called simile feature. Tom-vs-Pete classifier [14] encodes faces with high-level semantic information by the output scores of a large number of person-pair classifiers. Different from the deep learning approaches, the above methods are still shallow models and mostly rely on hand-crafted local features.

2.2 Deep Face Representation Learning

In recent years, deep learning methods are exploited to learn hierarchical representation and report state-of-the-art performance on LFW [15–19, 24].

DeepFace is an early attempt of applying deep convolutional neural network in real-world face recognition. There are four highlights in DeepFace: 1) A 3D model based face alignment to frontalize facial images with large pose. 2) A very large scale training set with 4 million face images of 4,000 identities. 3) Deep convolutional neural network with the local connected layer that learns separate kernel for each spatial position. 4) A Siamese network architecture to learn deep metric based on the features of the deep convolutional network.

The DeepID [16], DeepID2 [17] and DeepID2+ [18] are a series of works, which provide a very good example of deep network evolution. In DeepID, 25 CNN networks are trained on each face patch independently. Besides, Joint Bayesian method [25] is applied to learn robust face similarity metric. Finally, an ensemble of 25 deep networks achieve 97.45% mean accuracy on LFW. The DeepID2 introduces the joint identification and verification losses. The performance of DeepID2 on LFW is improved to 99.15%. The DeepID2+ just makes the network deeper and adds auxiliary loss signal on lower layer. Besides, the activation of the feature

embedding layer is also studied as sparse, selective and robust. The mean accuracy of DeepID2+ on LFW is 99.47% with 25 CNN models.

Learning face representation from scratch [24] presents a semi-automatic way to collect face images from the internet and builds a large scale dataset CASIA-Web containing about 494,414 images of 10,575 subjects. Then a 13-layer deep network with 10 convolutional layers and 3 fully-connected layer is trained with joint identification and verification losses, reporting 97.73% accuracy on LFW.

Another promising deep neural network is FaceNet [19] proposed by Google, which uses a super large scale face dataset containing 200 million face images of 8 million face identities to train a GoogLeNet network. Given such a large number of identities, the classical Softmax loss which needs the same number of 8 millon output nodes consumes too much GPU memory. So instead, a triplet loss which does not consume extra memory is introduced in FaceNet to directly optimize the embedding feature and achieves 99.63% mean accuracy on LFW.

All the above deep learning methods achieve quite promising face recognition accuracy on the challenging LFW dataset. The superiority demonstrates the superiority of the favorable feature learning ability of the deep neural networks.

2.3 Evolution of Benchmarks

In early years, most datasets for face recognition were collected in controlled environment, e.g. ORL [26], AR [27], FERET [28], PIE [29], FRGC [30], Extended-Yale-B [31], CAS-PEAL [32] and MultiPIE [33]. Among these datasets, AR is specially regarded as a benchmark to study occlusion robust face recognition. FERET, CAS-PEAL, PIE and MultiPIE are often used as general benchmarks to evaluate different factors of face recognition, such as aging, pose, expression, illumination, accessories etc. Yale-B is often cited as a lighting robust face recognition benchmark. Among these datasets, FRGC is widely used as a more challenging face recognition benchmark as it consist of over 50,000 images collected in varying lighting condition, e.g. atria, hallways, or outdoors.

In recent years, lots of real-world face datasets have been released, such as Labeled Face in the Wild [34], PubFig [23], CelebFaces [16], WDRef [25], SFC [15], CACD [35], WLFDB [36], CASIA-Web [24], MSRA-CFW [37] etc. These datasets are built for different motivations. Among

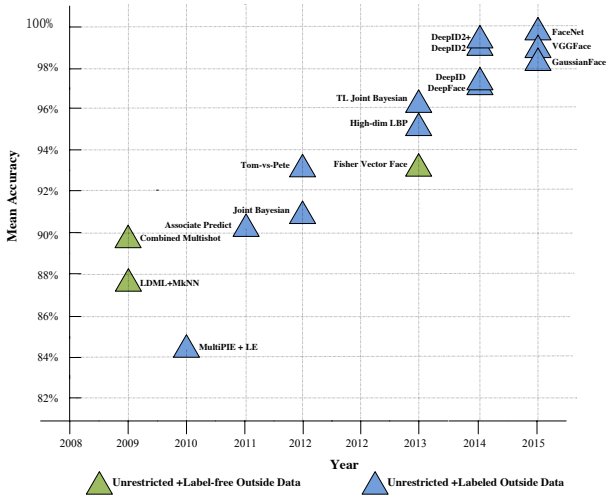

Fig. 1: Evolution of the face recognition techniques on LFW from 2009 to 2015. In the figure, triangles with different color refer to what type of outside training data is used in the experiments.

to train model for testing on LFW [34], but only CASIA-Web is a public dataset. WLFDB is a weakly labeled dataset without accurate identity annotation and acts as a search-based face tagging benchmark. CACD is built for studying cross-age face recognition, but only 10% of the subjects in CACD are manually annotated. The MSRA-CFW dataset is built for face retrieval evaluation and the face identity is automatically annotated by algorithms.

In the past several years, LFW has become the de-facto benchmark for real-world face recognition. According to the standard LFW protocol, the performance measurement should be the mean accuracy over 10-fold face verification task with each fold containing 300 inter-class and 300 intra-class face pairs. Besides the standard verification protocol, a face identification protocol is also available in [38]. A brief history of the performance evolution of LFW is demonstrated in Figure 1. Representative methods including LDML-MkNN [39], Multishot [40], LE [12], Associate-Predict [41], Tom-vs-Pete [14], Fisher Vector Face [22], High-dim LBP [4], TL Joint Bayesian [42], DeepFace [15], DeepID [16], DeepID2 [17], Gaussian Face [43], VGGFace [44], DeepID2+ [18] and FaceNet [19] are shown.

Figure 1 illustrates the evolution of face recognition techniques along with the accuracy increases: from early hand-crafted local features to shallow representation learning until recent deep representation learning. The first performance breakthrough is made by LDML-MkNN [39], which is a metric learning method, then the representation

Table 1: Comparisons of AlexNet [20], VIPLFaceNetFull and VIPLFaceNet. In the table, S denotes stride, G denotes group and Pad denotes padding. The ReLU layers are not shown in this table for the efficiency of presentation.

AlexNet	VIPLFaceNetFull	VIPLFaceNet
Conv1: 96x11x11, S:4, Pad:0	Conv1: 96x9x9, S:4, Pad:0	Conv1: 48x9x9, S:4, Pad:0
LRN	–	–
Pool1: 3x3,S:2	Pool1: 3x3,S:2	Pool1: 3x3,S:2
Conv2: 256x5x5, G:2, S:1, Pad:2	Conv2: 192x3x3,S:1, Pad:1	Conv2: 128x3x3,S:1, Pad:1
LRN	–	–
–	Conv3: 192x3x3,S:1, Pad:1	Conv3: 128x3x3,S:1, Pad:1
Pool2: 3x3,S:2	Pool2: 3x3,S:2	Pool2: 3x3,S:2
Conv3: 384x3x3, S:1, Pad:1	Conv4: 384x3x3,S:1, Pad:1	Conv4: 256x3x3,S:1, Pad:1
Conv4: 384x3x3, G:2, S:1, Pad:1	Conv5: 256x3x3,S:1, Pad:1	Conv5: 192x3x3,S:1, Pad:1
–	Conv6: 256x3x3,S:1, Pad:1	Conv6: 192x3x3,S:1, Pad:1
Conv5: 256x3x3, G:2, S:1, Pad:1	Conv7: 192x3x3,S:1, Pad:1	Conv7: 128x3x3,S:1, Pad:1
Pool3: 3x3,S:2	Pool3: 3x3,S:2	Pool3: 3x3,S:2
FC1, 4,096	FC1, 4,096	FC1, 4,096
Dropout1: dropout_ratio:0.5	Dropout1: dropout_ratio:0.5	Dropout1: dropout_ratio:0.5
FC2, 4,096	FC2, 2,048	FC2, 2,048
Dropout2: dropout_ratio:0.5	Dropout2: dropout_ratio:0.5	Dropout2: dropout_ratio:0.5
FC3, 10,575	FC3, 10,575	FC3, 10,575

improvement [4, 14, 22, 40]. Finally, deep learning approaches reach the best results on LFW [15–19, 44].

3 Proposed VIPLFaceNet

This section presents the details of our proposed VIPLFaceNet. Firstly, we introduce the network architecture design and simplification. Then, to accelerate the training of the deep network, we introduce the fast normalization layer. Finally, we present the technical details of face pre-processing and deep network training.

3.1 Network Architecture

The network architecture is the essential part of a deep model. Recently, some network architectures has been well recognized, such as AlexNet [20], GoogLeNet [45] and VGGNet [46]. Among them, the AlexNet is the simplest, with 5 convolutional layer and 3 fully-connected layers [20]. Besides the convolutional layer and fully-connected layer, the ReLU layer and the dropout operation, proposed in AlexNet, build the basis of the latest deep convolutional neural networks. Another important component of AlexNet is its local response normalization layer (LRN) which can improve the generalization ability of AlexNet. Aiming to go deeper, GoogLeNet is designed to be a 22-layer deep network and introduces an inception structure which extracts multi-scale features [45]. Differently, the VGGNet uses only 3×3 convolutional kernel and the stride is always set to 1 [46], while the feature map size is reduced only by the pooling operation. Among the above three networks, VGGNet is the slowest, while AlexNet is the simplest.

Design and simplification of VIPLFaceNet Network. Considering the success and efficiency of AlexNet, our network is designed by adapting AlexNet to incorporate some recent new findings. Compared with AlexNet, our VIPLFaceNet design has six main features: 1) we use 9×9 size for the first convolutional layer rather than 11×11, to reduce the computational cost. 2) We remove all local response normalization layers, as we found it unnecessary provided proper parameter initialization [47]. 3) we decompose the second 5×5 convolutional layer of AlexNet to two 3×3 layers, inspired by He *et al.*'s work [48]. 4) Specially, we remove all group structures in AlexNet as we exploit a more efficient way to do parallel training, i.e. asynchronous stochastic gradient descent [49]. 5) Further, we reduce the number of feature maps in each layer and add one more convolutional layer. 6) The number of nodes in the FC2 fully-connected layer is reduced to 2,048 from 4,096 inspired by the experimental analysis in [50].

The above six features lead to our VIPLFaceNetFull network consisting of 7 convolutional layers followed by 3 fully-connected layers, as shown in Table 1. Its computational cost is almost 90% of AlexNet, which is still very high. To further reduce the computational cost, we simplified VIPLFaceNetFull to VIPLFaceNet by reducing

in Table 1. Eventually, the VIPLFaceNet network consumes only 60% computations of AlexNet.

3.2 Fast Normalization Layer

As shown in LeCun *et al.*'s work [51], data normalization can speed up convergence, which is recently extended as the batch normalization algorithms [52]. Inspired by these works, we also exploit a fast normalization layer in our VIPLFaceNet before the ReLU layer to speed up the convergence and improve the generalization.

Specifically, the fast normalization layer aims at normalizing the output of each network node to be of zero mean and unit variance. Unlike the batch normalization in [52], our fast normalization layer does not have the recovery operation and thus consumes less GPU memory and computation cost. Suppose the output of the network consists of C dimensions, and the normalization is applied to each dimension independently. Next we take the k-th dimension as an example for illustrating and omit k for simplicity. The k-th dimension of the network output for all N training samples in a mini-batch is denoted as $\mathcal{B}_x = x_1, x_2, ..., x_N$. We denote the fast normalization layer (FNL) as:

$$FNL: x_1, x_2, ..., x_N \rightarrow \hat{o}_1, \hat{o}_2, ..., \hat{o}_N, \forall i, \hat{o}_i \sim N(0,1), \quad (1)$$

where $N(0,1)$ denotes the standard normal distribution with zero mean and unit variance. We present the detail of fast normalization layer (FNL) in Algorithm 1. In the algorithm, μ_x is initialized as 0 and σ_x is initialized as 1, and ω is the momentum value and set as 0.99 by default. In the test phase, μ_x and σ_x obtained in the final training stage are directly adopted.

During training, the fast normalization layer backpropagates the gradient using the chain rule as follows:

$$\begin{aligned}
\frac{\partial L}{\partial \sigma} &= -\frac{1}{2}\sum_{i=1}^{N}\frac{\partial L}{\partial \hat{o}_i}(x_i-\mu)\sigma^{-3/2}. \\
\frac{\partial L}{\partial \mu} &= (\sum_{i=1}^{N}\frac{\partial L}{\partial \hat{o}_i}\frac{-1}{\sqrt{\sigma}}) + \frac{\partial L}{\partial \sigma}\frac{-2\sum_{i=1}^{N}(x_i-\mu)}{N}. \\
\frac{\partial L}{\partial x_i} &= \frac{\partial L}{\partial \hat{o}_i}\frac{1}{\sqrt{\sigma}} + \frac{\partial L}{\partial \sigma}\frac{2(x_i-\mu)}{N} + \frac{1}{N}\frac{\partial L}{\partial \mu}.
\end{aligned} \quad (2)$$

Additionally, as observed from extensive experiments, the dropout operation can be safely removed for deep network with fast normalization layer. It is observed that not only the deep network training is greatly accelerated but also the generalization ability is improved. In the experimental sections, we will validate the effectiveness of the fast normalization layer.

Algorithm 1 Fast Normalization Layer (FNL)

Input: DCNN Network and mini-batch $\mathcal{B}_x$
Output: Normalized output for each sample in $\mathcal{B}_x$

1: Calculate the batch mean: $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$
2: Calculate the batch variance: $\sigma = \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2$
3: Calculate the normalized value: $\hat{o}_i = \frac{x_i-\mu}{\sqrt{\sigma}}$
4: Update the global mean: $\mu_x = \omega * \mu_x + (1-\omega) * \mu$
5: Update the global variance: $\sigma_x = \omega * \sigma_x + (1-\omega) * \sigma$.
6: **return** $\hat{o}_i$, i = 1, 2,..., N.

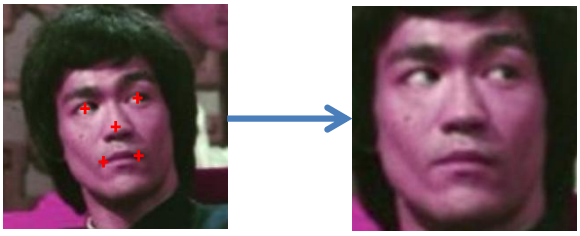

Fig. 2: Example of face normalization using five points.

3.3 Technical Details

In all experiments, the face images are preprocessed with three steps including face detection, facial landmark localization and face normalization.

Face Detection: In face detection stage, we adopt the face detection toolkit developed by VIPL lab of CAS [53]. One can refer to [54] for more details.

Facial Landmark Localization: We apply the Coarse-to-Fine Auto-Encoder Networks (CFAN) [55] to detect the five facial landmarks in the face, i.e. the left and right center of the eyes, the nose tip, the left and right corner of mouth.

Face Normalization: As shown in Figure 2, the face image is normalized to 256 × 256 pixels using five facial landmarks.

Training details In all experiments, for those deep networks without fast normalization layer , the base_lr is set as 0.01 and the learning rate is reduced following polynomial curve with gamma value equal to 0.5. For those deep networks with the fast normalization layer, the base_lr is set as 0.04. The momentum is set as 0.9 and the weight decay is set as 0.0005. All the experiments are conducted in Titan-X GPU with 12GB memory using a modified Caffe deep learning toolbox [56].

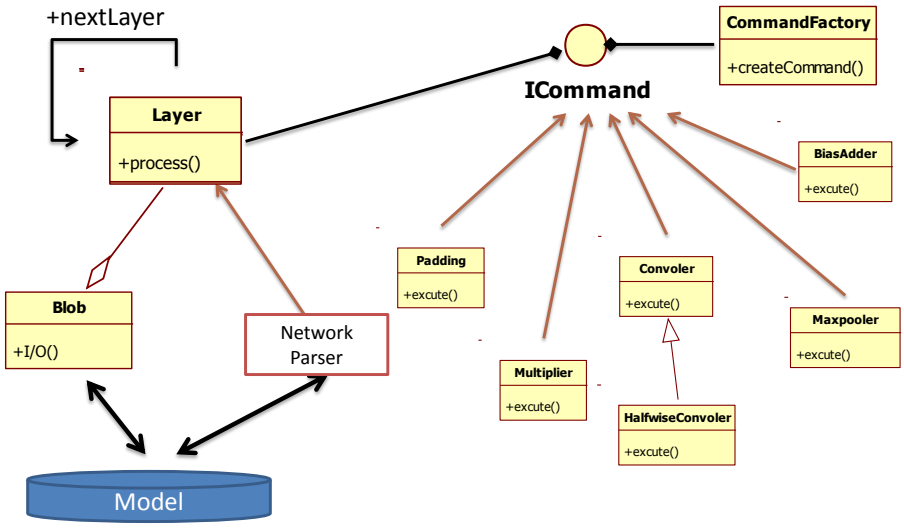

Fig. 3: The VIPLFaceNet SDK Architecture.

4 SDK Architecture

As mentioned previously, the source code of our VIPLFaceNet is released under the BSD license, and now available on https://github.com/seetaface. In this section, we introduce the highlights and architecture of the SDK for a better understanding of the source code.

4.1 Highlights of VIPLFaceNet SDK

This open-source SDK provides a powerful toolkit for testing and deploying face recognition applications, setting a good starting point for researchers and developer to experience the state-of-the-art face recognition technology.

High-performance. The VIPLFaceNet achieves state-of-the-art performance on the LFW benchmark with only single network. Further performance improvement can be expected by using metric leaning approaches, e.g. Joint Bayesian method [25] and MRMD [57] or classifier ensemble approaches, e.g. LibD3C [58].

Object-oriented. The VIPLFaceNet SDK is designed from the beginning to be an object-oriented software, allowing easy extension to new network layers and any user-defined network architecture. It can also be easily integrated into industrial face recognition systems for various tasks.

Configurable Network Architecture. VIPLFaceNet SDK facilities the network architecture configuration independent of the SDK code. The network definitions is saved in the model file using pre-defined format. VIPLFaceNet supports network architectures in the form of arbitrary directed acyclic graphs. Upon initialization, VIPLFaceNet parses the network architecture from the model file and loads the network parameters into memory.

Pure C++ Code. The VIPLFaceNet is implemented in fully C++ code. It is very efficient to deploy VIPLFaceNet in multiple hardware platforms and operation systems.

Community Cooperation. We will put our source code in the GitHub and leverage the whole community to improve the SDK for better performance and flexibility. More features, such as python interface and PHP interface can be expected with the support of the whole community.

4.2 Implementation Details

In this part, we will introduce the software architecture of the VIPLFaceNet. In Figure 3, the architecture of VIPLFaceNet SDK is presented. Main components in VIPLFaceNet SDK includes Blob, Command, Layer and Network Parser.

Blob. The blob is a container to hold the matrix in deep convolutional neural network. The Blob provides a mapping of the logic multi-dimensional matrix to physical one-dimensional array.

Command. The command is an interface that provides basic network elements, e.g. convolution, rectifier linear unit(ReLU), max pooling or mean pooling, and inner-product operation. As a SDK implementation for deployment, the implementation of the loss layers is unnecessary.

deep neural network. In VIPLFaceNet, a layer is composed by one or more commands, e.g. a convolutional layer is composed by a convolution command and a ReLU command.

Network Parser. To facilitate the definition of network architecture, VIPLFaceNet SDK sets up a network parser. A network is defined by multiple layers and organized as a directed acyclic graph. The SDK code does not need to be modified regardless the network architecture changes.

Matrix Multiplication Accelerating. The single instruction multiple data (SIMD) instructions is used in VIPLFaceNet SDK to accelerate the matrix multiplication operation in both convolutional layer and fully-connected layer. Meanwhile, VIPLFaceNet also supports the third-party linear algebra library, e.g. Armadillo [59].

5 Experiments and Analysis

In this section, we compare the proposed VIPLFaceNet with the existing methods on the real-world face recognition benchmark LFW. Then, we discuss how VIPLFaceNet can be extended to other face recognition scenarios.

5.1 Dataset and Evaluation Protocol

The training set of our open-source VIPLFaceNet is the CASIA-Web dataset [24]. All the training data are pre-processed as illustrated in section 3.3. Totally, we have 479,777 detected and normalized facial images of 10,575 identities in the training set. The VIPLFaceNet is learned from scratch, and the weights of both the convolutional kernels and the fully-connected layers are initialized using the MSRA filler, i.e. $Var[w] = 2/n$, where n is the number of input neurons [47]. The mean of training images are subtracted firstly. During training, face patches equal to crop_size × crop_size pixel are randomly sampled from the input images and the images are also randomly flipped with 50% probability. In all the experiments, the default crop_size is set as 227.

The evaluation dataset is Labeled Faces in the Wild dataset (LFW) [34], which contains 13,233 images of 5,749 identities. For the standard 10-fold face verification experiment on LFW, we follow the unrestricted setting using external labeled data. Each fold of the test set consists of 300 inter-class and 300 intra-class face pairs. We also conducted an additional experiment following the face identification protocol as in [38].

Table 3: The performance of our VIPLFaceNet and state-of-the-art methods on LFW under the identification protocol [38].

Method	Rank-1	DIR @ FAR = 1%
COTS-s1 [38]	56.70%	25.00%
COTS-s1+s4 [38]	66.50%	35.00%
DeepFace [15]	64.90%	44.50%
WSTFusion [60]	82.50%	61.90%
AlexNet+FNL [20]	89.26%	58.72%
VIPLFaceNetFull+FNL	**92.79%**	**68.13%**
VIPLFaceNet+FNL	**91.95%**	**63.26%**

5.2 Experimental Results on LFW

In this part, we report the accuracy of VIPLFaceNet on LFW under both verification protocol and identification protocol. In all the experiments, we take the 2,048 outputs of the FC2 fully-connected layer as the representation of each face images and exploit the cosine function as the similarity metric between features.

Comparisons with the state-of-the-art methods. In Table 2, we compare VIPLFaceNet with the state-of-the-art methods on LFW View 2 in terms of 10-fold mean accuracy. Using only single 10-layer deep convolutional neural network, our VIPLFaceNet achieves accuracy comparable to that of VGGFace [44] with a 16-layer deep network and DeepID2+ with 25 deep networks, even though VIPLFaceNet is trained with less training data than DeepFace, VGGFace and FaceNet. VIPLFaceNet also reduces 40% computation cost with slight performance degradation. Compared with AlexNet, VIPLFaceNet cuts down 40% error rate on LFW (1.4% vs. 2.3%). The superiority of our method comes from the careful design of network architecture and simplification.

We further evaluate the VIPLFaceNet in the close-set and open-set face identification tasks, following the protocol in [38]. The close-set identification protocol reports the Rank-1 identification accuracy and the open-set identification reports the detection and identification rate (DIR) at False Alarm Rate (FAR) equal to 1%. The comparisons with state-of-the-art methods are shown in Table 3. The proposed VIPLFaceNet also achieves state-of-the-art performance in the face identification tasks.

Comparisons of the crop size. The crop size is related to both the performance and computational cost of deep networks. Due to the random data argumentation, the smaller the crop size, the bigger randomness in training. In

Table 2: The performance of our VIPLFaceNet and state-of-the-art methods on LFW View2 under the verification protocol.

Method	Accuracy	# of Network	# of Training Images
High-dim LBP [4]	95.17%	–	–
Fisher Vector Face [22]	93.03%	–	–
DeepFace [15]	97.35%	3	4M
DeepID [16]	97.45%	25	200K
DeepID2 [16]	99.15%	25	200K
Gaussian Face [43]	98.52%	–	–
DeepID2+ [18]	99.47%	25	290K
DeepID2+(Single) [18]	98.70%	1	290K
WSTFusion [60]	98.37%	–	10M
VGGFace [44]	98.95%	1	2.6M
FaceNet [19]	99.63%	1	200M
AlexNet + FNL [20]	97.70%	1	500K
VIPLFaceNetFull + FNL	**98.62%**	1	500K
VIPLFaceNet + FNL	**98.60%**	1	500K

Table 4: The performance of our VIPLFaceNet with different crop size on LFW View2 under the verification protocol.

Network Architecture	Crop Size	Accuracy
VIPLFaceNet	256	98.12%
VIPLFaceNet	248	98.53%
VIPLFaceNet	**227**	**98.60%**
VIPLFaceNet	200	98.57%
VIPLFaceNet	180	98.21%

under the LFW verification protocol. It can be concluded that a moderate crop size 227 yields the best performance.

Evaluation of the Fast Normalization Layer. In Figure 4, we evaluate the effectiveness of the fast normalization layer (FNL) on LFW. As can be seen, adding FNL significantly improves the performance. Besides improving the generalization, FNL also significantly improves the convergence speed. By adding FNL, we can set the base learning rate as 0.04 and set the total echo as 15, while the baseline network needs 80 echoes for a good convergence. In Table 5, we compare the training time of VIPLFaceNet and VIPLFaceNet with FNL on the CASIA-Web training set. With the FNL, it only consumes 20% training time and only slightly increases the online test cost.

Extension of the VIPLFaceNet. We can easily extend the VIPLFaceNet for more applications. For example, fine-tuning the VIPLFaceNet for other-domain applications such as age estimation [61].

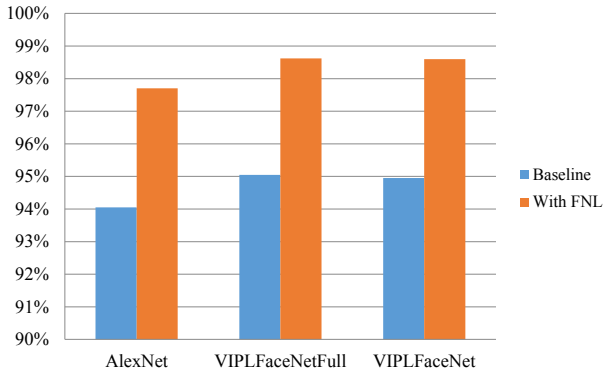

Fig. 4: The mean accuracy of our VIPLFaceNet on LFW View2 with or without FNL.

Table 5: The time cost of our VIPLFaceNet with or without FNL. The training time of VIPLFaceNet with fast normalization layer is reduced by 80%.

Method	Training Time	Test speed on CPU
AlexNet	67 hours	250ms / per image
VIPLFaceNetFull	60 hours	235ms / per image
VIPLFaceNet	40 hours	145ms / per image
VIPLFaceNetFull + FNL	12 hours	245ms / per image
VIPLFaceNet + FNL	8 hours	150ms / per image

6 Conclusions

In this paper, we propose and release an open-source deep face recognition SDK with carefully designed network architecture and simplification. By sticking a fast normalization layer to the ReLU layer, the training time is reduced by 80% and the performance is significantly

improved. On the real-world face recognition benchmark LFW, VIPLFaceNet achieves a mean accuracy of 98.60%, which is comparable to the state-of-the-art. A fully C++ implementation of the VIPLFaceNet SDK is released as an open source SDK under the BSD license. VIPLFaceNet can serve as a good start point for both academic research and industrial applications under various real-world face recognition scenarios.

In the future, we would advocate the whole community to improve the SDK, e.g. supporting more language interface such as Python or PHP. Besides, we intend to build a VIPLFaceNet-based active face recognition development community and support more related application scenarios such as ID photo vs. real-word image verification, facial attribute analysis and age estimation.

Acknowledgements This work is partially supported by 973 Program under contract No. 2015CB351802, Natural Science Foundation of China under contracts Nos. 61402443, 61390511, 61379083, 61222211.

References

1. Zhao W Y, Chellappa R, Phillips P J, Rosenfeld A. Face recognition: A literature survey. Acm Computing Surveys, 2003, 35(4): 399–458
2. Liu C, Wechsler H. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. IEEE Transactions on Image Processing, 2002, 11(4): 467–476
3. Ahonen T, Hadid A, Pietikainen M. Face description with local binary patterns: Application to face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, 28(12): 2037–2041
4. Chen D, Cao X D, Wen F, Sun J. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2013, 3025–3032
5. Albiol A, Monzo D, Martin A, Sastre J, Albiol A. Face recognition using hog–ebgm. Pattern Recognition Letters, 2008, 29(10): 1537–1543
6. Vu N S, Caplier A. Enhanced patterns of oriented edge magnitudes for face recognition and image matching. IEEE Transactions on Image Processing, 2012, 21(3): 1352–1365
7. Hussain S U, Napoléon T, Jurie F, others . Face recognition using local quantized patterns. In: Proceedings of British Machive Vision Conference. 2012, 11–20
8. Bicego M, Lagorio A, Grosso E, Tistarelli M. On the use of sift features for face authentication. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2006, 35–35
9. Kumar R, Banerjee A, Vemuri B C, Pfister H. Trainable convolution filters and their application to face recognition. IEEE Transactions on
10. Lei Z, Yi D, Li S Z. Discriminant image filter learning for face recognition with local binary pattern like representation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2012, 2512–2517
11. Xie S F, Shan S G, Chen X L, Meng X, Gao W. Learned local gabor patterns for face representation and recognition. Signal Processing, 2009, 89(12): 2333–2344
12. Cao Z M, Yin Q, Tang X O, Sun J. Face recognition with learning-based descriptor. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2010, 2707–2714
13. Cui Z, Li W, Xu D, Shan S G, Chen X. Fusing robust face region descriptors via multiple metric learning for face recognition in the wild. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2013, 3554–3561
14. Berg T, Belhumeur P N. Tom-vs-pete classifiers and identity-preserving alignment for face verification. In: Proceedings of British Machine Vision Conference. 2012, 5
15. Taigman Y, Yang M, Ranzato M, Wolf L. Deepface: Closing the gap to human-level performance in face verification. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2014, 1701–1708
16. Sun Y, Wang X, Tang X. Deep learning face representation from predicting 10,000 classes. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2014, 1891–1898
17. Sun Y, Chen Y H, Wang X G, Tang X O. Deep learning face representation by joint identification-verification. In: Proceedings of Advances in Neural Information Processing Systems. 2014, 1988–1996
18. Sun Y, Wang X G, Tang X O. Deeply learned face representations are sparse, selective, and robust. arXiv preprint arXiv:1412.1265, 2014
19. Schroff F, Kalenichenko D, Philbin J. Facenet: A unified embedding for face recognition and clustering. arXiv preprint arXiv:1503.03832, 2015
20. Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks. In: Proceedings of Advances in Neural Information Processing Systems. 2012, 1097–1105
21. Liu X, Shan S G, Li S X, Hauptmann A G. Everything is in the face? represent faces with object bank. In: Proceedings of Asian Conference on Computer Vision Workshops. 2014, 180–193
22. Simonyan K, Parkhi O M, Vedaldi A, Zisserman A. Fisher vector faces in the wild. In: Proceedings of British Machive Vision Conference. 2013, 7
23. Kumar N, Berg A C, Belhumeur P N, Nayar S K. Attribute and simile classifiers for face verification. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2009, 365–372
24. Yi D, Lei Z, Liao S C, Li S Z. Learning face representation from scratch. arXiv preprint arXiv:1411.7923, 2014
25. Chen D, Cao X, Wang L, Wen F, Sun J. Bayesian face revisited: A joint formulation. In: Proceedings of European Conference on Computer Vision, 566–579. Springer, 2012
26. Samaria F S, Harter A C. Parameterisation of a stochastic model for human face identification. In: Proceedings of IEEE Workshop on Applications of Computer Vision. 1994, 138–142

27. Martinez A M. The ar face database. CVC Technical Report, 1998, 24
28. Phillips P J, Moon H, Rizvi S A, Rauss P J. The feret evaluation methodology for face-recognition algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(10): 1090–1104
29. Sim T, Baker S, Bsat M. The cmu pose, illumination, and expression (pie) database. In: Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition. 2002, 46–51
30. Phillips P J, Flynn P J, Scruggs T, Bowyer K W, Chang J, Hoffman K, Marques J, Min J, Worek W. Overview of the face recognition grand challenge. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2005, 947–954
31. Lee K C, Ho J, Kriegman D. Acquiring linear subspaces for face recognition under variable lighting. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(5): 684–698
32. Gao W, Cao B, Shan S G, Chen X L, Zhou D L, Zhang X H, Zhao D B. The cas-peal large-scale chinese face database and baseline evaluations. IEEE Transactions on Systems, Man and Cybernetics Part A System and Humans, 2008, 38(1): 149
33. Gross R, Matthews I, Cohn J, Kanade T, Baker S. Multi-pie. Image and Vision Computing, 2010, 28(5): 807–813
34. Huang G B, Ramesh M, Berg T, Learned-Miller E. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007
35. Chen B C, Chen C S, Hsu W H. Cross-age reference coding for age-invariant face recognition and retrieval. In: Proceedings of European Conference on Computer Vision, 768–783. Springer, 2014
36. Y W D, Hoi S, K Z J. Wlfdb: Weakly labeled face databases. Technical Report, 2014
37. Zhang X, Zhang L, Wang X J, Shum H Y. Finding celebrities in billions of web images. IEEE Transactions on Multimedia, 2012, 14(4): 995–1007
38. Best-Rowden L, Han H, Otto C, Klare B F, Jain A K. Unconstrained face recognition: Identifying a person of interest from a media collection. IEEE Transactions on nformation Forensics and Security, 2014, 9(12)
39. Guillaumin M, Verbeek J, Schmid C. Is that you? metric learning approaches for face identification. In: Proceedings of International Conference on Computer Vision. 2009, 498–505
40. Taigman Y, Wolf L, Hassner T, others . Multiple one-shots for utilizing class label information. In: Proceedings of British Machive Vision Conference. 2009, 1–12
41. Yin Q, Tang X O, Sun J. An associate-predict model for face recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2011, 497–504
42. Cao X, Wipf D, Wen F, Duan G Q, Sun J. A practical transfer learning algorithm for face verification. In: Proceedings of International Conference on Computer Vision. 2013, 3208–3215
43. Lu C C, Tang X O. Surpassing human-level face verification performance on lfw with gaussianface. arXiv preprint arXiv:1404.3840, 2014
44. Parkhi O M, Vedaldi A, Zisserman A. Deep face recognition. Proceedings of the British Machine Vision, 2015, 1(3): 6
45. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. arXiv preprint arXiv:1409.4842, 2014
46. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014
47. He K M, Zhang X Y, Ren S Q, Sun J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. arXiv preprint arXiv:1502.01852, 2015
48. He K M, Sun J. Convolutional neural networks at constrained time cost. arXiv preprint arXiv:1412.1710, 2014
49. Zhang S S, Zhang C, You Z, Zheng R, Xu B. Asynchronous stochastic gradient descent for dnn training. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing. 2013, 6660–6663
50. Chatfield K, Simonyan K, Vedaldi A, Zisserman A. Return of the devil in the details: Delving deep into convolutional nets. arXiv preprint arXiv:1405.3531, 2014
51. LeCun Y, Bottou L, Orr G B, Müller K R. Efficient backprop. In: Neural networks: Tricks of the trade, 9–48. Springer, 2012
52. Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. Proceedings of Internatioal Conference on Machine Learning, 2015, 448–456
53. www.vipl.ict.ac.cn . Visual information processing and learning (vipl) group in institute of computing technology (ict) chinese academy of sciences (cas)
54. Yan S, Shan S G, Chen X, Gao W. Locally assembled binary (lab) feature with feature-centric cascade for fast and accurate face detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2008, 1–7
55. Zhang J, Shan S G, Kan M N, Chen X L. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In: Proceedings of European Conference on Computer Vision, 1–16. Springer, 2014
56. Jia Y Q, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T. Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia. 2014, 675–678
57. Zou Q, Zeng J C, Cao L J, Ji R R. A novel features ranking metric with application to scalable visual and bioinformatics data classification. Neurocomputing, 2016, 173: 346–354
58. Lin C, Chen W Q, Qiu C, Wu Y F, Krishnan S, Zou Q. LibD3C: ensemble classifiers with a clustering and dynamic selection strategy. Neurocomputing, 2014, 123: 424–435
59. http://arma.sourceforge.net/docs.html . Armadillo c++ linear algebra library
60. Taigman Y, Yang M, Ranzato M A, Wolf L. Web-scale training for face identification. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2015, 2746–2754
61. Liu X, Li S X, Kan M N, Zhang J, Wu S Z, Liu W X, Han H, Shan S G, Chen X L. Agenet: Deeply learned regressor and classifier for robust apparent age estimation. In: Proceedings of IEEE International Conference on Computer Vision Workshops. 2015, 258–266

Xin Liu recieved the B.S. degree at the Chongqing University, Chongqing, China, in June 2011. Currently, he is a Ph.D candidate at the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include face recognition, image retrieval and deep learning.

Meina Kan is an Associate Professor with the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS). She received the Ph.D. degree from the University of Chinese Academy of Sciences (CAS), Beijing, China. Her research mainly focuses on Computer Vision especially face recognition, transfer learning, and deep learning.

Wanglong Wu recieved the B.S. degree at the Beijing Jiaotong University, Beijing, China, in June 2014. Currently, he is a Ph.D candidate at the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include face recognition and deep learning.

Shiguang Shan received M.S. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1999, and Ph.D. degree in computer science from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China, in 2004. He joined ICT, CAS in 2002 and has been a Professor since 2010. He is now the Deputy Director of the Key Lab of Intelligent Information Processing of CAS. His research interests cover computer vision, pattern recognition, and machine learning. He especially focuses on face recognition related research topics. He has published more than 200 papers in refereed journals and proceedings.

Xilin Chen received the B.S., M.S., and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 1988, 1991, and 1994, respectively. He is a Professor with the Institute of Computing Technology, Chinese Academy of Sciences (CAS). He has authored one book and over 200 papers in refereed journals and proceedings in the areas of computer vision, pattern recognition, image processing, and multimodal interfaces.