# VIPLFaceNet: An Open Source Deep Face Recognition SDK

In this work, we propose an open source face recognition method with deep representation named as VIPLFaceNet. The training set of our open-source VIPLFaceNet is the CASIA-Web dataset.

## Architecture

10-layer deep convolutional neural network with 7 convolutional layers and 3 fully-connected layers.

Compared with AlexNet, our VIPLFaceNet design has six main features: 1) we use 9 × 9 size for the first convolutional layer. 2) We remove all local response normalization layers, as we found it unnecessary provided proper parameter initialization. 3) we decompose the second 5 × 5 convolutional layer of AlexNet to two 3 × 3 layers,. 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 (https://bit.ly/2TWKGVS). 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.

## Fast Normalization Layer

The fast normalization layer aims at normalizing the output of each network node to be of zero mean and unit variance. It does not have the recovery operation.

**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,..., $N$.

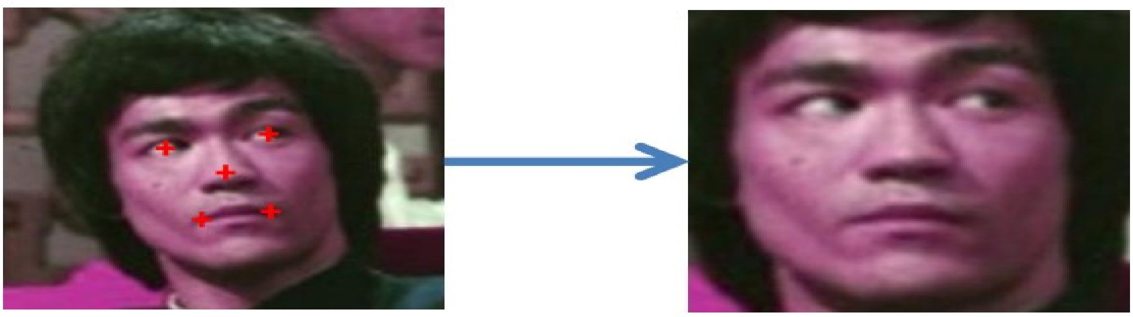

Figure 1: Example of face normalization using five points.

## Technical Details

In all experiments, the face images are preprocessed with three steps including face detection by the face detection toolkit developed by VIPL lab of CAS, facial landmark localization by the Coarse-toFine Auto-Encoder Networks (CFAN) to detect the five facial landmarks in the face and face normalization using five facial landmarks..

## SDK Architecture

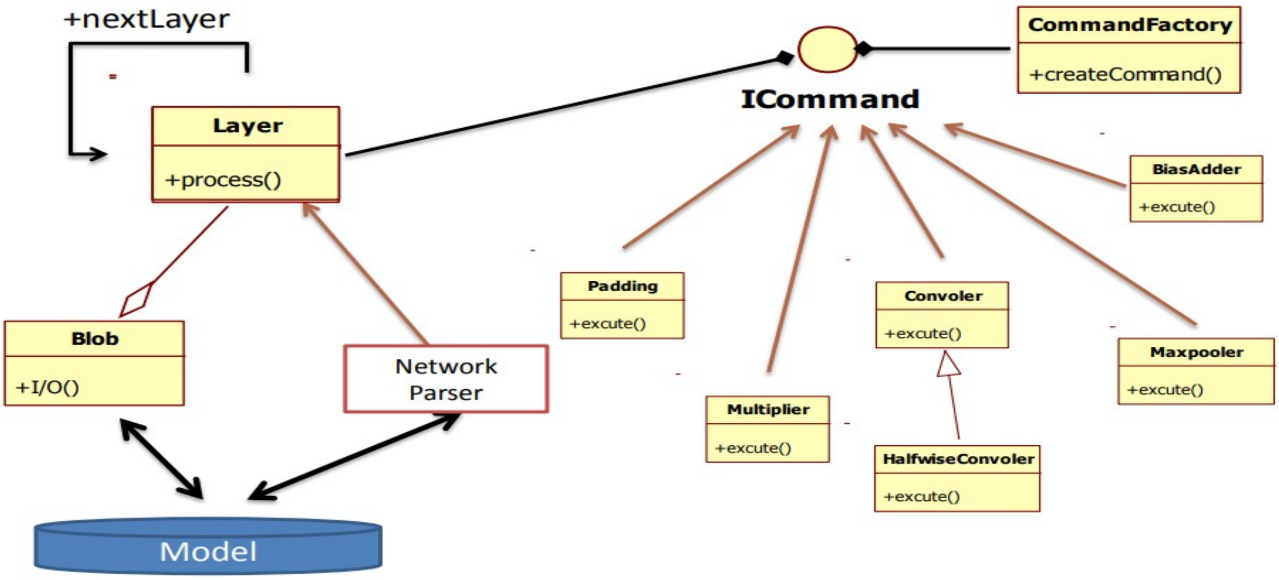


Fig. 2: The VIPLFaceNet SDK Architecture
Blob: The blob is a container to hold the matrix in deep convolutional neural network
Command: The command is an interface that provides basic network elements
Network Parse:. To facilitate the definition of network architecture

## Time & Accuracy

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.

| 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% |

| 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 |

Table 1: The performance of our VIPLFaceNet with different crop size on LFW View2 under the verification protocol.

Table 2: The time cost of our VIPLFaceNet with or without FNL

VIPLFaceNet: VIPLFaceNetFull + reduce the number of filters in the convolutional layers,