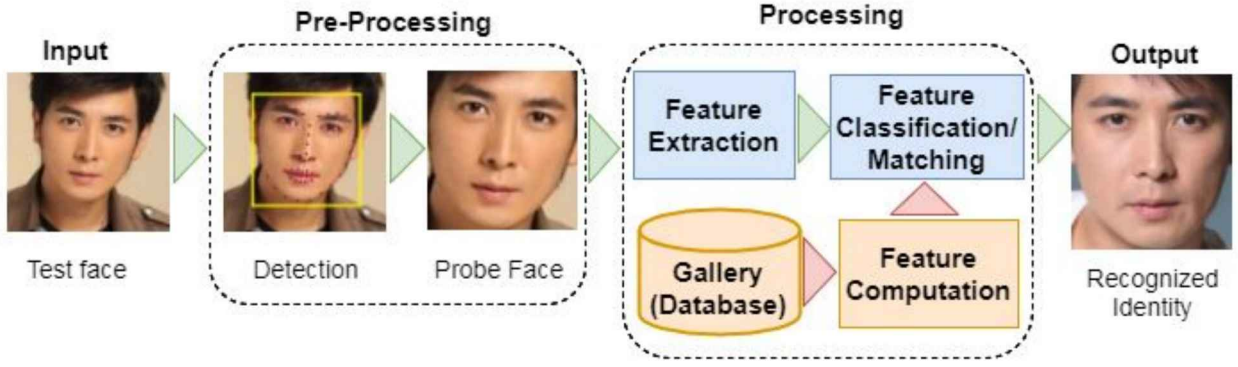
A Survey on Deep Learning based Face Recognition

Na Zhang

Part I: Introduction to Deep Learning and CNN

- Introduction
- Deep Learning Methods
- Some Specific Face Recognition Problems
- Databases

- Face Recognition (FR)



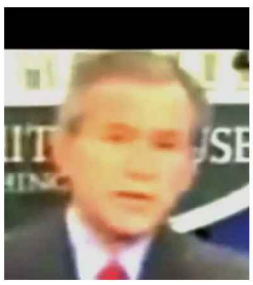
- FR has been applied widely to daily lives

access control

video surveillance

- The demands of FR are also growing quickly in recent years
- In practice, FR is affected by many factors in unconstrained FR

low resolution



pose variation

complex illumination

motion blur

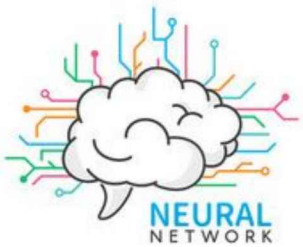
- Traditional algorithms
  - may not do well for unconstrained face matching

- Eigenfaces
- Fisherfaces
- Bayesian face
- Metaface
- SVM
- Boosting
- .....

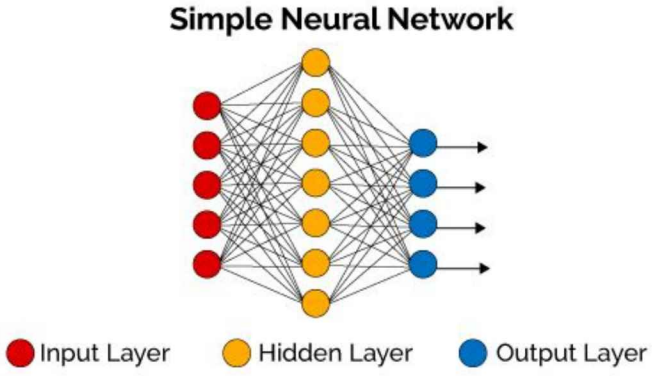
- Neural network (NN)
  - ✔ Brings in a new direction for FR research
- In many pattern recognition systems, compared to traditional methods
  - ✔ NN is an effective construct
  - ✔ Has been shown to demonstrate many tangible advantages with regards to its learning ability, generalization aspect, and robustness

So, what is NN?

NN



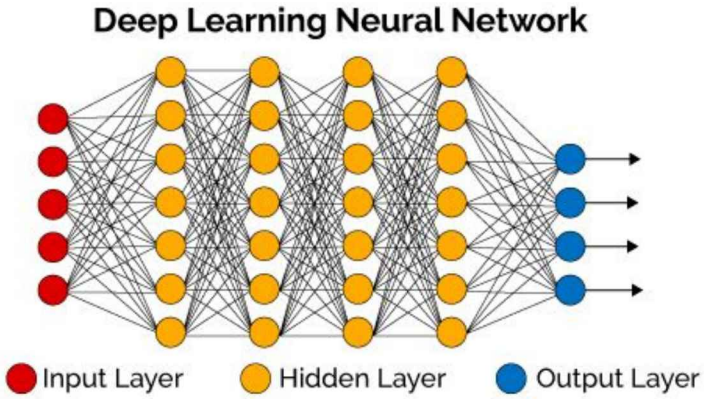
- ✔ A biologically inspired mathematical model
- ✔ A machine learning algorithm inspired from the working of human brain which enable a system to learn from some observational data
- ✔ A simple NN consist of an input layer, a single hidden layer and an output layer
- ✔ which contain a number of neurons that can be activated or not



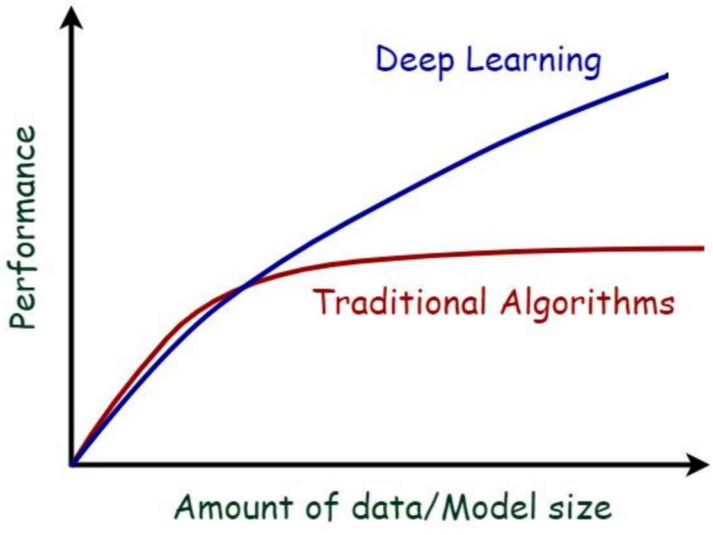
- Advantages of NN
  - ✔ Adaptive Learning
  - ✔ Self-Organization
  - ✔ Real Time Operation
  - ✔ Fault Tolerance via redundant information coding
  - ✔ the abilities to handle complicated or imprecise data
  - ✔ Can capture the complex face patterns for FR

Deep Learning

- A machine learning technique that performs learning in multiple hidden layers of nonlinear processing units
- It is a deep neural network(DNN)
- Each successive layer of DNN uses the output from the previous layer as input



- Recently, Deep Neural Network has established itself as a dominant technique in machine learning
- Deep and large networks have exhibited impressive results when there are large training data sets and computation resources (many CPU cores and/or GPUs)



- Due to deep learning techniques, there have been significant advances in face recognition

- In early time, research interests mainly concentrated on face recognition with deep networks on visible light face images and/or video faces

Visible images

Video

- With the emergence of various types of face data, research concentrations have also focused on some specific tasks
  - robust to changes of pose, illumination, expression, age, etc.
  - improving performance of video, 3D, and heterogeneous FR
    - ✔ e.g., NIR-VIS, photo-sketch, still-to-video

RGB-D

photo-sketch

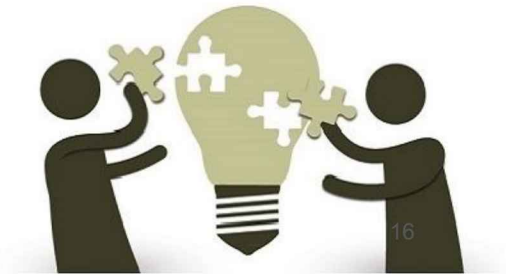
NIR-VIS

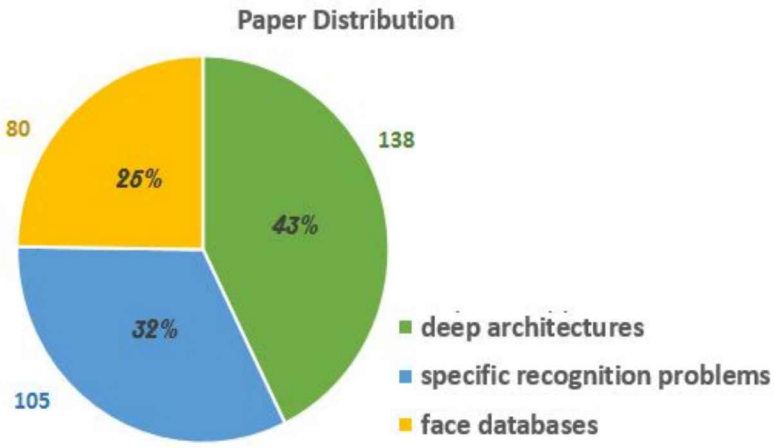
previous work

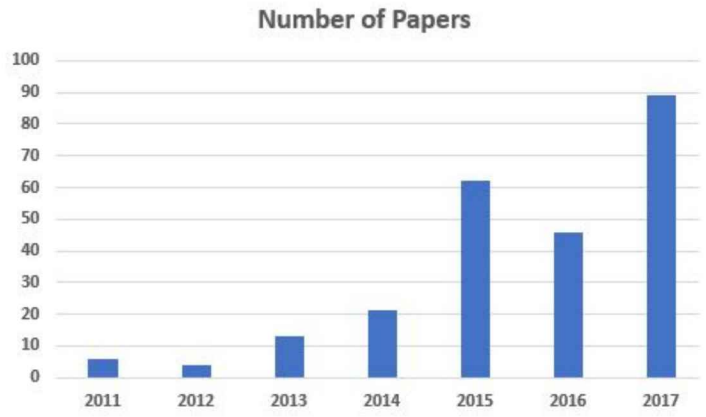
- Some related surveys overviewed methods on handling:
  - pose, expression, occlusion
  - Infra-red, single-modal and multimodal
  - video, 3D, heterogeneous face matching
  - … etc.
- However, most of them focus on the traditional methods
- Few has been related to deep learning methods

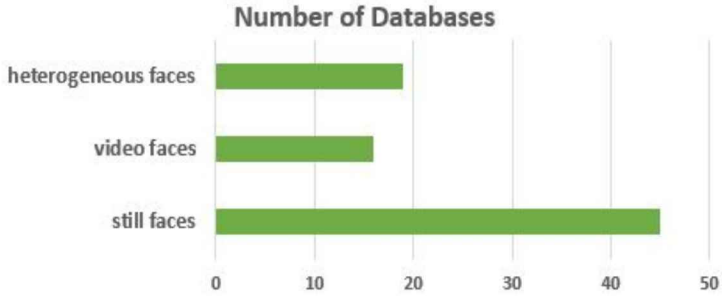
OUR WORK:

- We present a complete, comprehensive overview of FR works using deep learning
  - Consider both the deep architectures and specific recognition problems
  - Most are within the recent five years
  - Give a review of related face databases
- It is expected to cover most, if not all, of the works incorporating deep learning methods for face recognition.



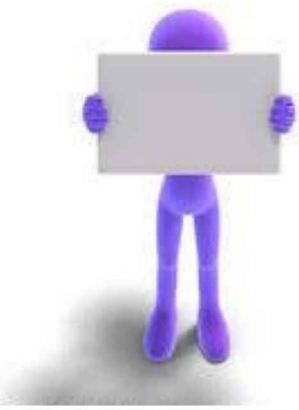






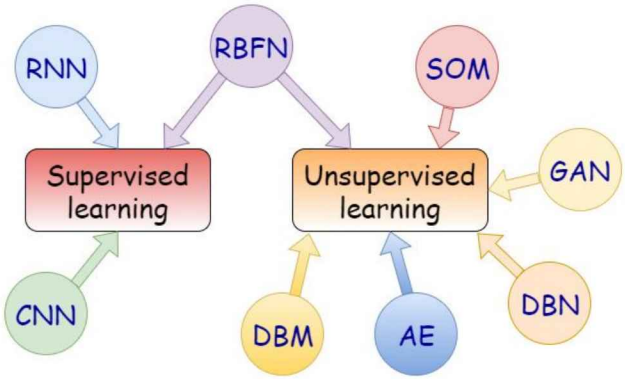
By this survey, we show that:

- Deep learning methods have been fully used in face recognition and played an important role
- Many specific issues or challenges to address in FR, such as pose, illumination, expression, 3D, heterogenous matching, etc.
- Lots of face datasets have been collected in recent years, including still images, videos, and heterogeneous data useful for cross-modal face matching

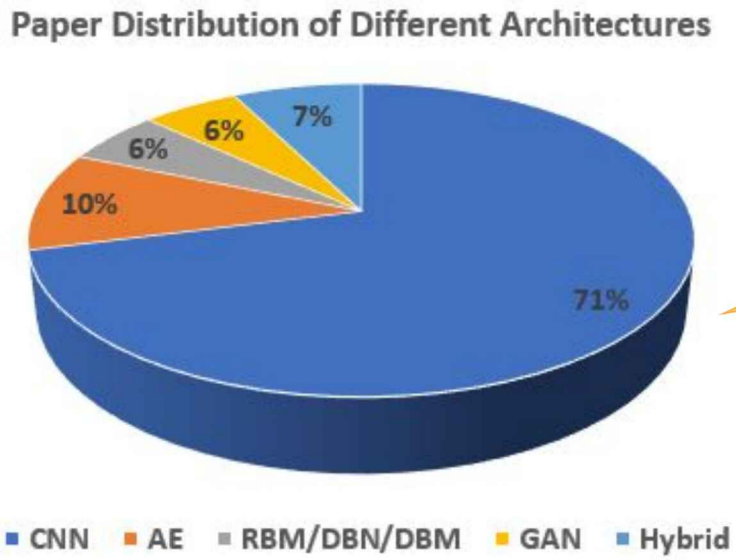


- Introduction
- Deep Learning Methods
- Some Specific Face Recognition Problems
- Databases

- Most deep neural networks can be grouped in two categories:
- Supervised learning:
  - Use class labels directly for the deep models
  - Find model parameters that best predict the data with loss function(s)
- Unsupervised learning:
  - Process data without using class labels
  - To find patterns, such as latent subspaces



- ✔ RBFN: Radial Basis Function Network
- ✔ CNN: Convolutional Neural Network
- ✔ RNN: Recurrent Neural Network
- ✔ DBN: Deep Belief Network
- ✔ AE: Autoencoder
- ✔ DBM: Deep Boltzmann Machine:
- ✔ SOM: Self-Organizing Map
- ✔ GAN: Generative Adversarial Network



RNN: not much to FR
SOM, RBFN: not much using deep learning technique

CNN: Convolutional Neural Network

Supervised learning

CNN

- Contains convolutional, pooling, fully connected layers
- Convolutional layers:

$$X_k^t = \sigma(W_k^{t-1} * X^{t-1} + b_k^{t-1})$$

$W = W_1, W_2, \ldots, W_k$: learnable filters
$B = b_1, b_2, \ldots, b_k$: added biases
X_k: feature map
$\sigma(\cdot)$: element-wise nonlinear transform
t: *t-th* convolutional layer

- Pooling layers:
  - A form of nonlinear down-sampling
  - max pooling, average pooling, *L*2-norm pooling

Most Popular

- CNN could decrease the dimension dramatically by convolutional layers and pooling layers with convolution filters of a small extent, then send to a fully connected layer

- LeNet (1998): probably the first successful real world application of CNN for hand-written digit recognition

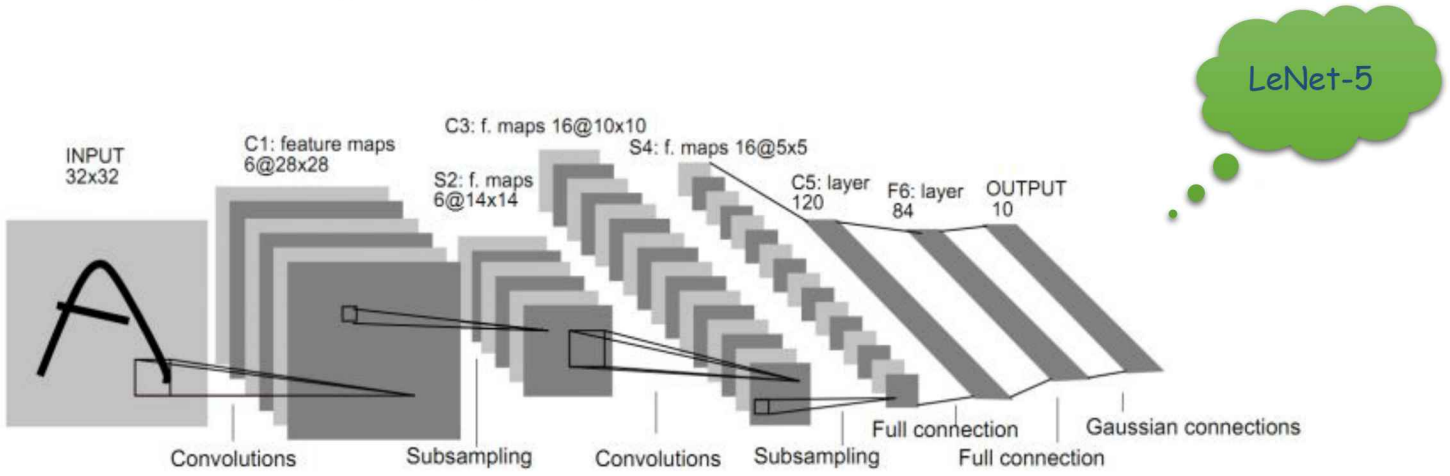


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11):2278–2324

- When AlexNet (2012) was proposed, further progress has been made using deeper architectures

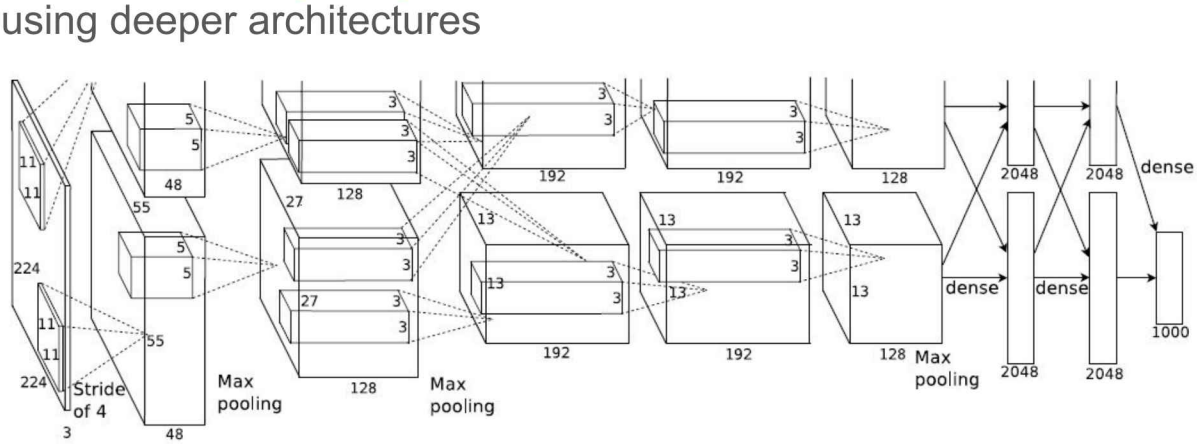
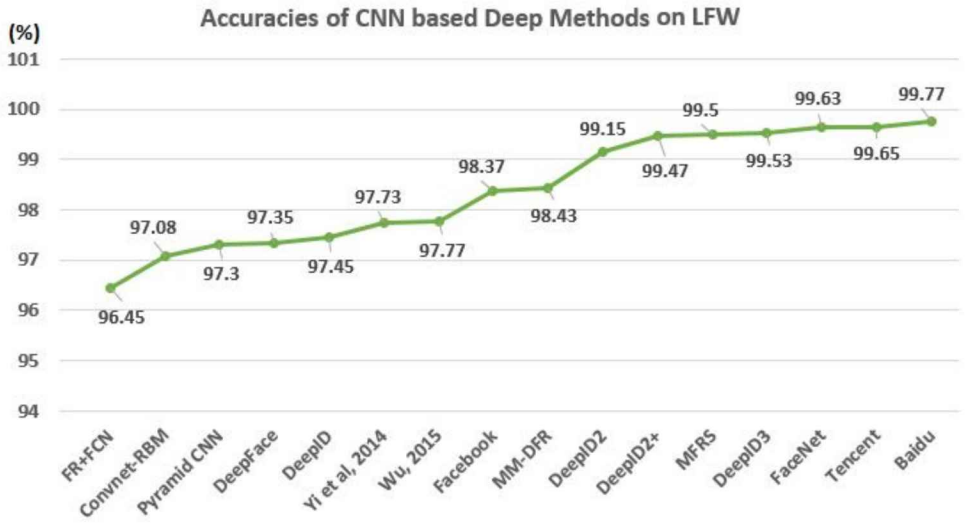
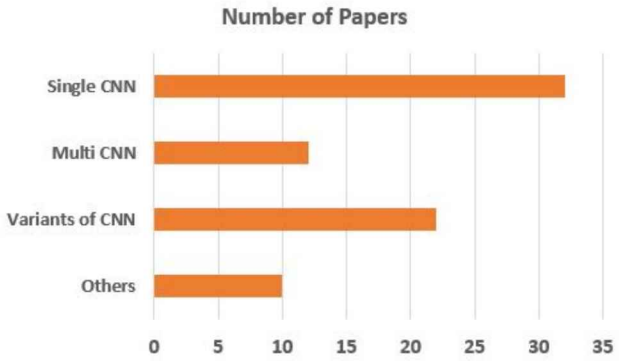


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105

- A large number of current CNN based face recognition methods obtained robust features and outperformed the traditional methods





25

❖ Single CNN

- Typical deep face recognition approaches use a single CNN

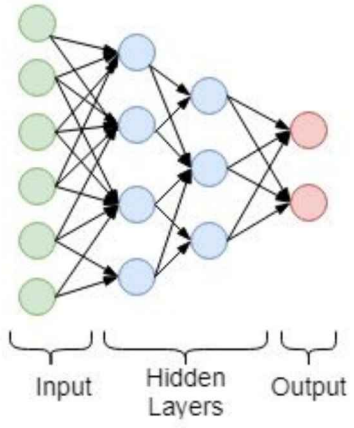


Table 1 Overview of deep learning methods based on single CNN

Algorithm	Description/Remark
DeepFace (Taigman et al, 2014)	Employ 3D face modeling to apply a piecewise affine tranformation to derive feature
Web-Scale (Taigman et al, 2015)	Use a bootstrapping process to select an efficient training set from a large dataset to alleviate performance saturation
Wu (2015)	Use MFM activation function and get better performance than DeepFace, WebFace
FaceNet (Schroff et al, 2015)	An end-to-end system; Directly learn a mapping from face images to a compact Euclidean space; Has great representational efficiency
VGGFace (Parkhi et al, 2015)	Combine very deep convolution neural network and the triplet embedding
Wang et al (2017c)	Apply a Discriminative Covariance oriented Representation Learning framework
Li et al (2015b)	Batch learning strategy; Mahalanobis metric and distance threshold for optimization
Grundström (2015)	An algorithm suitable for real time use in an embedded environment with limited space and restricted computational resources
Seo et al (2015)	A multi-task learning; Use two-stage learning strategy to minimize error functions
Light CNN (Wu et al, 2015)	Light frameworks with reduced parameters and time to learn a 256-D compact embedding on the large scale face data with massive noisy labels
DeepVisage (Hasnat et al, 2017)	Incorporate residual learning framework; Normalized features used for softmax loss

26

Sankaranarayanan et al (2016)	Deep CNN based approach combined with a low-dimensional discriminative embedding which are learned by triplet probability constraints
Gruber et al (2017)	Use a 50-layer deep residual network ResNet to face recognition task
Center Loss (Wen et al, 2016b)	With the joint supervision of softmax loss and center loss
VLAD-DCNN (Zheng et al, 2016)	Combine VLAD feature encoding with DCNN features
SphereFace (Liu et al, 2017b)	Learn features with angular margin; Discriminative on hypersphere manifold
Smirnov et al (2017)	Insert sampling method into feature learning process
NR-Network (Ding et al, 2017)	Learn noise-robust deep feature representation
Yeung et al (2017)	A constrained triplet loss layer to be replaced at the bottom of neural network
He et al (2015b)	A predictable hash code algorithm; Map face samples to Hamming space
Sparse ConvNets (Sun et al, 2016)	With sparse neural connections in an iterative way from the previously learned denser models with a neural correlation based weight selection criterion
Grm et al (2016)	A two-structural parts network; Convolutional layers try to capture the joint characteristics of input image pair; Fully-connected layers produce a similarity index
Yang et al (2017)	A fully convolutional structure with higher speed and less computational cost; Use max-feature-map as activation function
Hayat et al (2017)	A data-driven approach which can jointly learn registration with representation
Park et al (2017)	Get features directly used to determine if two input images are identical
Hsieh et al (2017)	A multi-task learning model; Incorporate identity and high-level human attributes
FV-DCNN (Chen et al, 2016b)	Combine deep feature and Fisher vector representation
Hu et al (2017a)	Fuse facial attribute feature with face recognition features
Lumini et al (2016)	Combine deep features and hand-crafted features
Chen et al (2016a)	Use a joint Bayesian metric learning to assess the similarity
NormFace (Wang et al, 2017a)	Use normalized features to train DCNN
Jones and Kobori (2017)	Use hyperplane similarity to train CNN

DeepFace (Taigman et al, 2014)

- ✔ a 9-layer CNN
- ✔ input is preprocessed with 3D-aligned, 3-channel (RGB) face images
- ✔ Several locally connected convolutional layers are adopted without weight sharing
- ✔ every location in feature maps of these layers learns a different set of filters

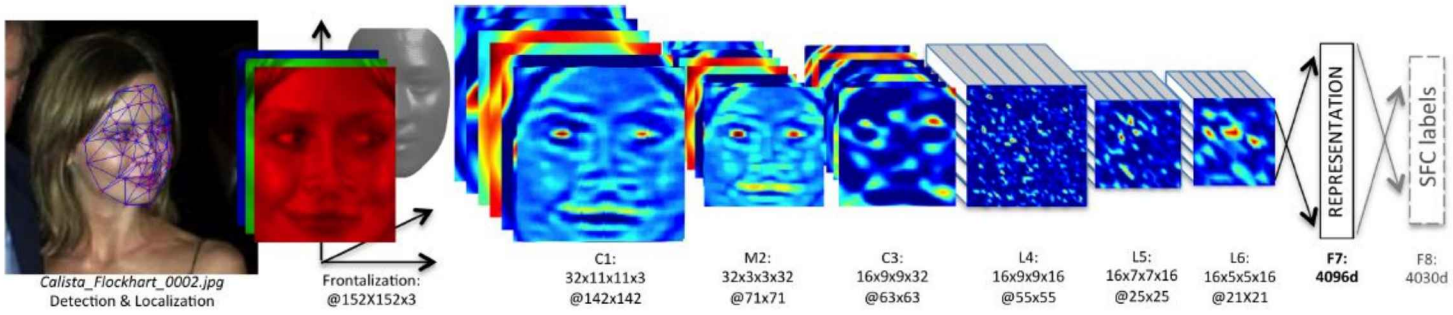


Figure 2. **Outline of the *DeepFace* architecture**. A front-end of a single convolution-pooling-convolution filtering on the rectified input, followed by three locally-connected layers and two fully-connected layers. Colors illustrate feature maps produced at each layer. The net includes more than 120 million parameters, where more than 95% come from the local and fully connected layers.

Taigman Y, Yang M, Ranzato M, Wolf L (2014) Closing the gap to human-level performance in face verification. deepface. In: IEEE Computer Vision and Pattern Recognition

Web-Scale (Taigman et al, 2015)

- ✔ an extension of DeepFace
- ✔ replace the naive random subsampling of training set
- ✔ the Web-Scale used a bootstrapping process to select a more efficient training set

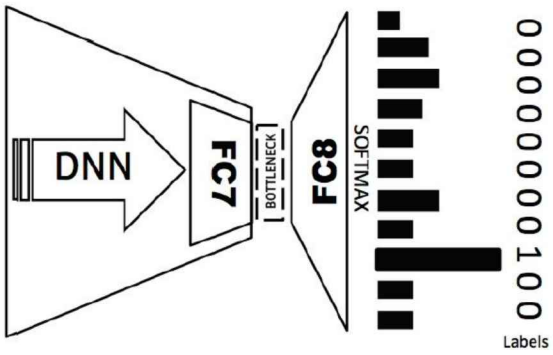


Figure 2. **The bottleneck.** The representation layer splits the network between the part that converts the input into a *generic* face descriptor and the part that performs linear classification to *specific* K classes. FC7 and FC8 are the low-rank matrices that project to- and from the bottleneck.

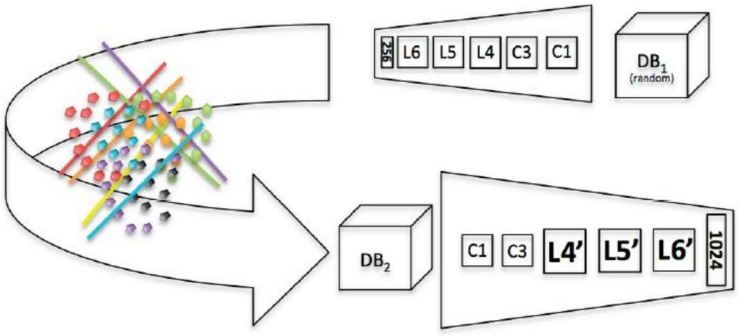


Figure 3. **The bootstrapping method.** An initial 256D-compressed representation trained on DB_1 is used to find the semantically-nearest identities of randomly picked 100 seeds, in a large pool of pre-trained hyperplanes. The union of all 100 groups of selected identities define the bootstrapped dataset DB_2. A larger capacity network with enlarged locally-connected layers and a 1024D representation is then trained.

Taigman Y, Yang M, Ranzato M, Wolf L (2015) Web-scale training for face identification. In: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, pp 2746–2754

FaceNet (Schroff et al, 2015)

- ✔ learned a mapping from face images to an Euclidean space where distances directly correspond to a measure of face similarity
- ✔ It optimizes the embedding itself by a triplet loss rather than an intermediate bottleneck layer in some previous networks.

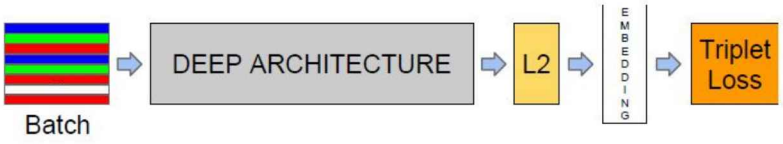


Figure 2. **Model structure.** Our network consists of a batch input layer and a deep CNN followed by L_2 normalization, which results in the face embedding. This is followed by the triplet loss during training.

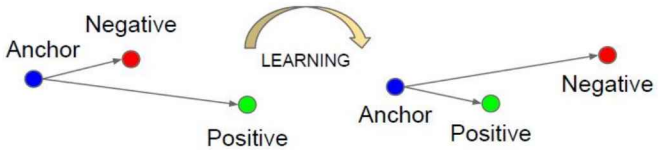


Figure 3. The **Triplet Loss** minimizes the distance between an *anchor* and a *positive*, both of which have the same identity, and maximizes the distance between the *anchor* and a *negative* of a different identity.

Schroff F, Kalenichenko D, Philbin J (2015) Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, pp 815–823

VGGFace (Parkhi et al,2015)

- ✔ is a deep convolutional neural network
- ✔ they comprise a long sequence of convolutional layers
- ✔ fine-tunes the model via a triplet-based metric learning method like FaceNet

layer	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
type	input	conv	relu	conv	relu	mpool	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv
name	–	conv1_1	relu1_1	conv1_2	relu1_2	pool1	conv2_1	relu2_1	conv2_2	relu2_2	pool2	conv3_1	relu3_1	conv3_2	relu3_2	conv3_3	relu3_3	pool3	conv4_1
support	–	3	1	3	1	2	3	1	3	1	2	3	1	3	1	3	1	2	3
filt dim	–	3	–	64	–	–	64	–	128	–	–	128	–	256	–	256	–	–	256
num filts	–	64	–	64	–	–	128	–	128	–	–	256	–	256	–	256	–	–	512
stride	–	1	1	1	1	2	1	1	1	1	2	1	1	1	1	1	1	2	1
pad	–	1	0	1	0	0	1	0	1	0	0	1	0	1	0	1	0	0	1

layer	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
type	relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	softmx
name	relu4_1	conv4_2	relu4_2	conv4_3	relu4_3	pool4	conv5_1	relu5_1	conv5_2	relu5_2	conv5_3	relu5_3	pool5	fc6	relu6	fc7	relu7	fc8	prob
support	1	3	1	3	1	2	3	1	3	1	3	1	2	7	1	1	1	1	1
filt dim	–	512	–	512	–	–	512	–	512	–	512	–	–	512	–	4096	–	4096	–
num filts	–	512	–	512	–	–	512	–	512	–	512	–	–	4096	–	4096	–	2622	–
stride	1	1	1	1	1	2	1	1	1	1	1	1	2	1	1	1	1	1	1
pad	0	1	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0

Table 3: **Network configuration.** Details of the face CNN configuration A. The FC layers are listed as "convolution" as they are a special case of convolution (see Section 4.3). For each convolution layer, the filter size, number of filters, stride and padding are indicated.

Parkhi OM, Vedaldi A, Zisserman A, et al (2015) Deep face recognition. In: BMVC, vol 1, p 6

Yeung et al (2017)

- ✔ introduced a constrained triplet loss layer (CTLL) to improve the recognition performance
- ✔ this loss layer helps the deep learning model to specify further distinguishable clusters between different people (classes) by:
  - placing extra constraints on images of the same person (intra-person)
  - while putting margins on images of a different person (inter-person)

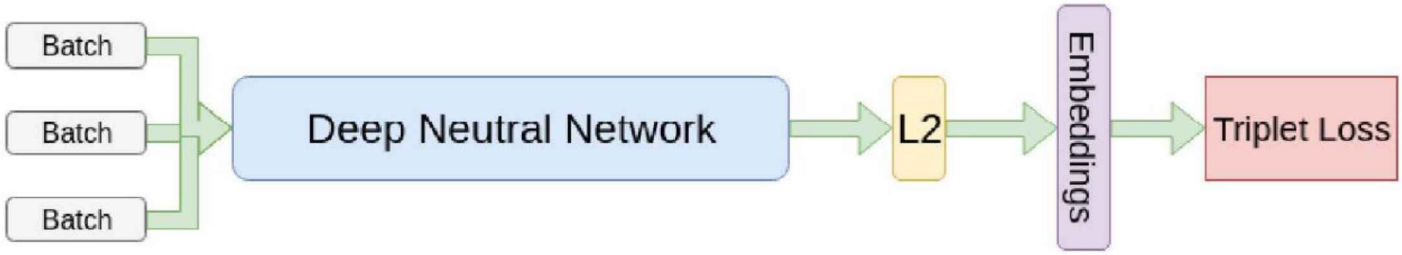


Fig. 1. Triplet Model Structure. The neural network takes a number of batches as input and followed by the deep architecture (CNN). The L2 normalization layer will produce the face representations (embeddings). At the end, the triplet loss function is applied on these embeddings

Yeung HWF, Li J, Chung YY (2017) Improved performance of face recognition using cnn with constrained triplet loss layer. In: Neural Networks, Intl. Joint Conf. on, IEEE, pp 1948–1955

Li et al (2015b)

- ✔ a 7-layer CNN model for age-invariant face verification
- ✔ can learn features, distance metrics and threshold simultaneously
- ✔ two tricks to overcome insufficient memory capacity and reduce computational cost

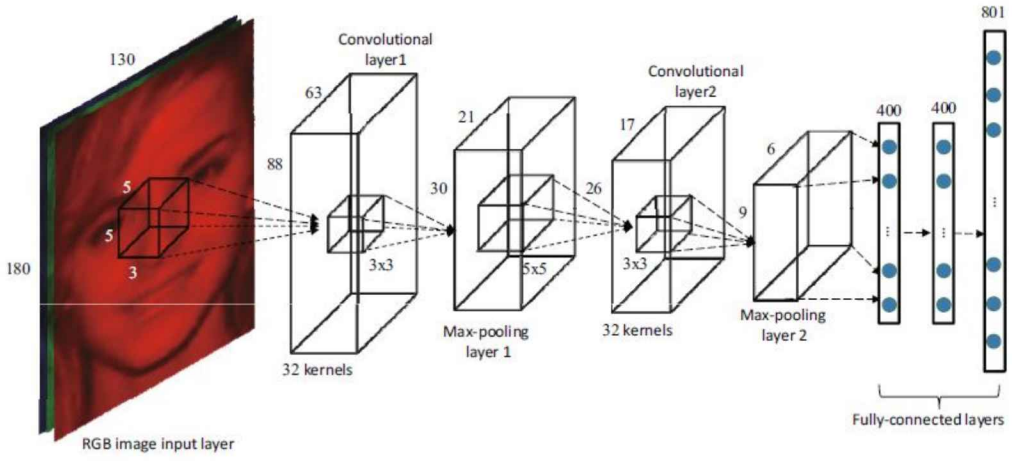


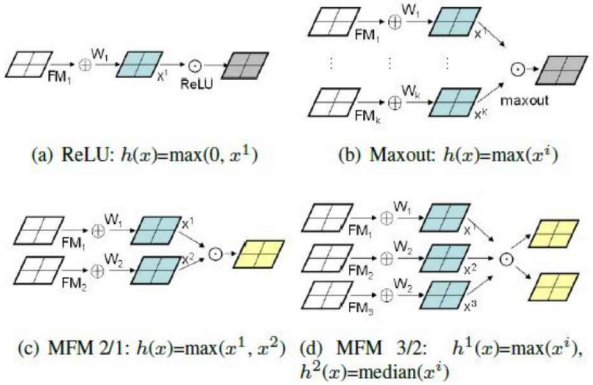
Fig. 2. Architecture of our model. The first and the third layers are convolutional layers, the second and the fourth are max-pooling layers. The last three layers are fully-connected layers.

Li Y, Wang G, Lin L, Chang H (2015b) A deep joint learning approach for age invariant face verification. In: CCF Chinese Conf. on Computer Vision, Springer, pp 296–305

Light CNN (Wu et al, 2015)

- ✔ presented 3 light frameworks following the idea of:
  - AlexNet
  - VGGFace
  - ResNet (residual networks)
- ✔ not only obtain better performance but also reduce parameters and time-consuming
- ✔ introduce Max-Feature-Map (MFM) activation into each convolutional layer of CNN
- ✔ adopt a semantic bootstrapping method to deal with noisy labels (re-label the training data)
- ✔ learn a compact embedding (256-D) on the large-scale face data with massive noisy labels

Light CNN-4
Light CNN-9
Light CNN-29



(a) ReLU: $h(x)=\max(0, x^1)$ (b) Maxout: $h(x)=\max(x^i)$

(c) MFM 2/1: $h(x)=\max(x^1, x^2)$ (d) MFM 3/2: $h^1(x)=\max(x^i)$, $h^2(x)=\text{median}(x^i)$

Figure 1. A comparison of different neural inhabitation. (a) ReLU suppresses a neuron by thresholding magnitude responses. (b) Maxout with enough hidden units makes a piecewise linear approximation to an arbitrary convex function. (c) MFM 2/1 suppresses a neuron by a competitve relationship. It is the simplest case of maxout activations. (d) MFM 3/2 activates two neurons and suppresses one neuron.

Wu X, He R, Sun Z, Tan T (2015) A light cnn for deep face representation with noisy labels. arXiv preprint arXiv:151102683

Center Loss (Wen et al, 2016b)

- ✔ proposed a center loss function to learn a more discriminative feature
- ✔ learns a center for deep features in each class
- ✔ penalizes the distances between the deep features and their corresponding class centers
- ✔ It effectively characterizes the intra-class variations

Wen Y, Zhang K, Li Z, Qiao Y (2016b) A discriminative feature learning approach for deep face recognition. In: European Conf. on Computer Vision, Springer, pp 499–515

C: The convolution layer
P: The max-pooling layer
LC: The local convolution layer
FC: The fully connected layer

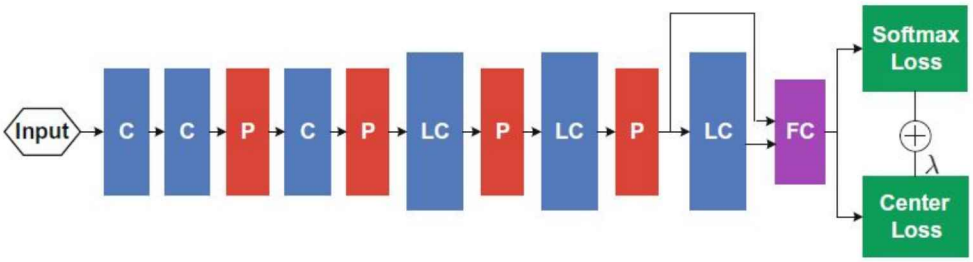


Fig. 4. The CNN architecture using for face recognition experiments. Joint supervision is adopted. The filter sizes in both convolution and local convolution layers are 3×3 with stride 1, followed by PReLU [12] nonlinear units. Weights in three local convolution layers are locally shared in the regions of 4×4, 2×2 and 1×1 respectively. The number of the feature maps are 128 for the convolution layers and 256 for the local convolution layers. The max-pooling grid is 2×2 and the stride is 2. The output of the 4th pooling layer and the 3th local convolution layer are concatenated as the input of the 1st fully connected layer. The output dimension of the fully connected layer is 512. **Best viewed in color.** (Color figure online)

Sankaranarayanan et al (2016)

- ✔ to address the unconstrained face verification
- ✔ couples a deep CNN-based approach with a low-dimensional discriminative embedding step
- ✔ a triplet probability embedding learning method to improve the performance of deep features

Layer	Kernel Size/Stride	#params
conv1	11x11/4	35K
pool1	3x3/2	
conv2	5x5/2	614K
pool2	3x3/2	
conv3	3x3/1	885K
conv4	3x3/1	1.3M
conv5	3x3/1	2.3M
conv6	3x3/1	2.3M
conv7	3x3/1	2.3M
pool7	3x3/2	
fc6	1024	18.8M
fc7	512	524K
fc8	10548	10.8M
Softmax Loss		Total: 39.8M

Table 1: Deep Network architecture details

Sankaranarayanan S, Alavi A, Castillo CD, Chellappa R (2016) Triplet probabilistic embedding for face verification and clustering. In: Biometrics Theory, Applications and Systems, Intl. Conf. on, IEEE, pp 1–8

Sparse ConvNets (Sun et al, 2016)

- ✔ learned an effective DCNN model with sparse neural connections
- ✔ to get good initializations and avoid bad local minima
- ✔ which is derived from a baseline high-performance VGG-like deep network

type	patch size/ stride	output size	params
convolution (1a)	$3 \times 3/1$	$112 \times 96 \times 64$	1.8K
convolution (1b)	$3 \times 3/1$	$112 \times 96 \times 64$	37K
max pool	$2 \times 2/2$	$56 \times 48 \times 64$	
convolution (2a)	$3 \times 3/1$	$56 \times 48 \times 96$	55K
convolution (2b)	$3 \times 3/1$	$56 \times 48 \times 96$	83K
max pool	$2 \times 2/2$	$28 \times 24 \times 96$	
convolution (3a)	$3 \times 3/1$	$28 \times 24 \times 192$	166K
convolution (3b)	$3 \times 3/1$	$28 \times 24 \times 192$	332K
max pool	$2 \times 2/2$	$14 \times 12 \times 192$	
convolution (4a)	$3 \times 3/1$	$14 \times 12 \times 256$	443K
convolution (4b)	$3 \times 3/1$	$14 \times 12 \times 256$	590K
max pool	$2 \times 2/2$	$7 \times 6 \times 256$	
local connection (5a)	$3 \times 3/1$	$5 \times 4 \times 256$	11.8M
local connection (5b)	$3 \times 3/1$	$3 \times 2 \times 256$	3.5M
full connection (f)		512	786K

Table 1. Baseline ConvNet structures.

- starting point: the high-performance well trained baseline model
- Then delete connections in the baseline model in a layer-wise fashion, from the last fully-connected layer to the previous locally-connected and convolutional layers
- When a layer is sparsified, a new model is re-trained initialized by its previous model
- Therefore, a sequence of models with fewer and fewer connections are trained
- Finally, the final sparse ConvNet is obtained

Sun Y, Wang X, Tang X (2016) Sparsifying neural network connections for face recognition. In: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, pp 4856–4864

Zheng et al (2016)

- ✔ proposed a DCNN based approach for unconstrained face verification
- ✔ Combine the Vector of Locally Aggregated Descriptor (VLAD) feature encoding with DCNN features
- ✔ VLAD-encoded DCNN (VLAD-DCNN) features, is that spatial and appearance information are simultaneously processed to learn an improved discriminative representation

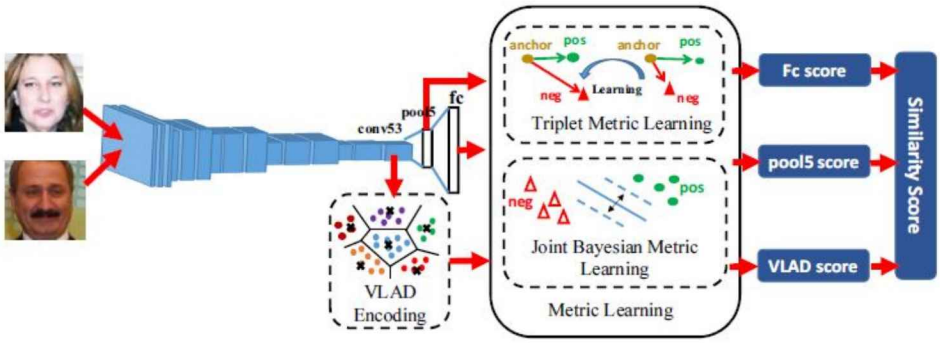


Fig. 1. An overview of the proposed fusion framework to combine the global average pooling, fully-connected layer features and VLAD features for unconstrained face verification.

Zheng J, Chen JC, Bodla N, Patel VM, Chellappa R (2016) Vlad encoded deep convolutional features for unconstrained face verification.
In: Pattern Recognition, Intl. Conf. on, IEEE, pp 4101–4106

DeepVisage (Hasnat et al, 2017)

- ✔ an efficient framework with 27 convolutional and 1 fully connected layers
- ✔ It incorporates residual learning framework
- ✔ uses normalized features to compute softmax loss

	Input	CoPr	CoPr	Pool	ResBl	CoPr	Pool	ResBl	CoPr	Pool	ResBl	CoPr	Pool	ResBl	FC	FN	Output
Filt Support	112 x 96	3	3	2	3	3	2	3	3	2	3	3	2	3	512		Softmax
Stride		1	1	2	1	1	2	1	1	2	1	1	2	1			
Pad		1	1	0	1	1	0	1	1	0	1	1	0	1			
# Filts		32	64		64	128		128	256		256	512		512			
# Replications		1	1	1	1	1	1	**2**	1	1	**5**	1	1	**3**	1	1	

Figure 1: Illustration of the proposed CNN architecture. ***CoPr*** indicates convolution followed by the PReLU activation function. ***ResBl*** is a residual block which computes $output = input + CoPr(CoPr(input))$. ***# Replication*** indicates how many times the same block is sequentially replicated in the CNN model. ***# Filts*** denotes the number of feature maps. ***FN*** denotes feature normalization.

Hasnat A, Bohn´e J, Gentric S, Chen L (2017) Deepvisage: Making face recognition simple yet with powerful generalization skills. arXiv preprint arXiv:170308388

Gruber et al (2017)

- ✔ presents initial experiments of an application of deep residual network to face recognition task
- ✔ utilize 50-layer deep neural network ResNet architecture
- ✔ The neural network was modified and then fine-tuned for face recognition purposes
- ✔ The experiments of classification of closed and open subset show the great potential of residual learning for face recognition

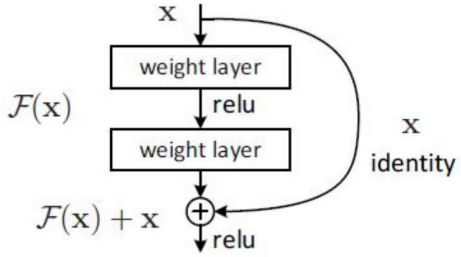


Figure 2. Residual learning: a building block.

Formally building block is defined as:

$$y = F(x, \{W_i\}) + x,$$

Gruber I, Hlav´a˘c M, ˘Zelezn`y M, Karpov A (2017) Facing face recognition with resnet: Round one. In: Intl. Conf. on Interactive Collaborative Robotics, Springer, pp 67–74

SphereFace (Liu et al, 2017b)

- a deep hypersphere embedding approach
- with angular softmax (A-Softmax) as the loss function

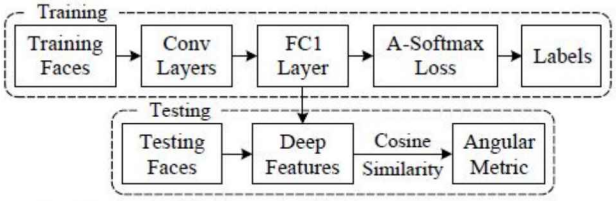


Figure 4: Training and Extracting *SphereFace* features.

Layer	4-layer CNN	10-layer CNN	20-layer CNN	36-layer CNN	64-layer CNN
Conv1.x	[3×3, 64]×1, S2	[3×3, 64]×1, S2	[3×3, 64]×1, S2 $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times1$	[3×3, 64]×1, S2 $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	[3×3, 64]×1, S2 $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$
Conv2.x	[3×3, 128]×1, S2	[3×3, 128]×1, S2 $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times1$	[3×3, 128]×1, S2 $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	[3×3, 128]×1, S2 $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	[3×3, 128]×1, S2 $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times8$
Conv3.x	[3×3, 256]×1, S2	[3×3, 256]×1, S2 $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	[3×3, 256]×1, S2 $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times4$	[3×3, 256]×1, S2 $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times8$	[3×3, 256]×1, S2 $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times16$
Conv4.x	[3×3, 512]×1, S2	[3×3, 512]×1, S2	[3×3, 512]×1, S2 $\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times1$	[3×3, 512]×1, S2 $\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	[3×3, 512]×1, S2 $\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$
FC1	512	512	512	512	512

Table 2: Our CNN architectures with different convolutional layers. Conv1.x, Conv2.x and Conv3.x denote convolution units that may contain multiple convolution layers and residual units are shown in double-column brackets. E.g., [3×3, 64]×4 denotes 4 cascaded convolution layers with 64 filters of size 3×3, and S2 denotes stride 2. FC1 is the fully connected layer.

Liu W, Wen Y, Yu Z, Li M, Raj B, Song L (2017b) Sphereface: Deep hypersphere embedding for face recognition. arXiv preprint arXiv:170408063

DCRL (Wang et al, 2017c)

- ✔ proposed a Discriminative Covariance oriented Representation Learning (DCRL) framework for face recognition with image sets
- ✔ by learning deep representations which can match the subsequent image set modeling and classification

Wang W, Wang R, Shan S, Chen X (2017c) Discriminative covariance oriented representation learning for face recognition with image sets. In: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, pp 5599–5608

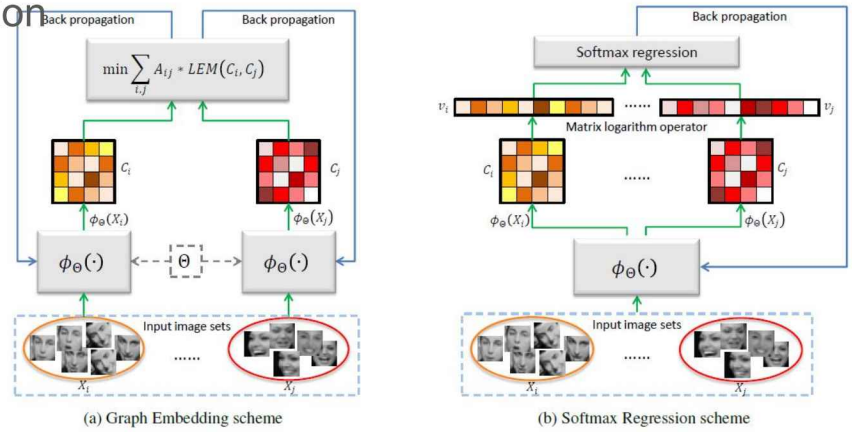


Figure 1: Conceptual illustration. The basic idea is to find a shared mapping $\phi_\Theta(\cdot)$ which projects the images of different sets into a target feature space such that the set model (i.e., set covariance matrix) calculated in this target space has maximum discriminative ability. More precisely, since $\phi_\Theta(\cdot)$ is a CNN which is parameterized by Θ, we seek to find a value of Θ to meet such optimization objective. Note that the green and blue arrows denote feeding forward and back propagation respectively. (a) Given a pair of image sets (X_i, X_j), the **Graph Embedding** scheme optimizes Θ through minimizing the Log-Euclidean metric (LEM) weighted by an affinity matrix A, whose entries correspond to pairs of set covariance matrices calculated in the target image feature space. (b) For the **Softmax Regression** scheme, we seek to find a Θ which ensures that log-covariance vectors corresponding to different sets can be classified by a softmax regression machine.

Doppelganger mining(Smirnov et al, 2017)

- ✔ a method to learn better face representations
- ✔ The main idea:
  - to maintain a list with the most similar identities for each identity in the training set
  - This list is used to generate better mini-batches by sampling pairs of similar-looking identities (”doppelgangers”) together

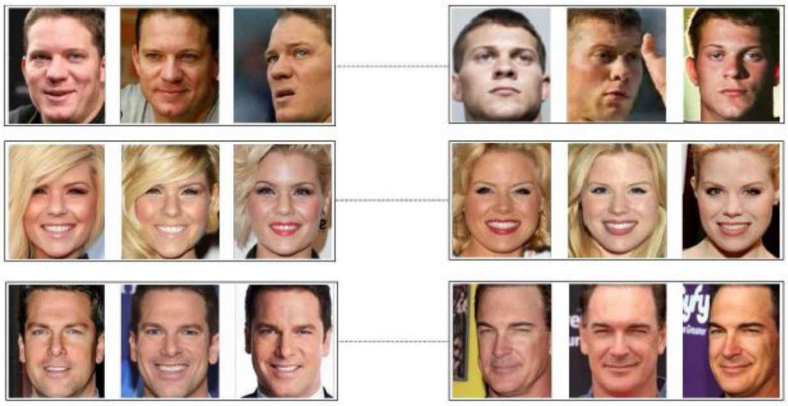


Figure 1. Examples of the doppelganger identities from the Low-shot face recogntion challenge dataset (base set) [9]. Identities at the left and their corresponding doppelgangers at the right.

Smirnov E, Melnikov A, Novoselov S, Luckyanets E, Lavrentyeva G (2017) Doppelganger mining for face representation learning. In: Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition, pp 1916–1923

NR-Network(Ding et al, 2017)

- ✔ a model for noise-robust deep feature representation which can increase inter-personal variations and reduce intra-personal variations at the same time
- ✔ Part 1:
  - contains two convolutional layers
  - each layer is followed by a max pooling layer, respectively
- ✔ part 2:
  - is an inception module which contains one max pooling layer and four convolutional layers with different kernel sizes.
  - the outputs of the three convolutional layers are connected together by a concat layer (Conc1)
- ✔ part 3:
  - with a max pooling layer and a convolutional layer is inserted
  - Following the inception is also a concat layer (Conc2)
- ✔ Part 4:
  - In order to extract both the low-level and high-level features hierarchically, the final fully connected layer is connected to the outputs of all the three parts with 256 hidden neurons
  - The output of this fully connected layer serves as the face representation
  - Followed by the final inner product layer are the normalization and dropout

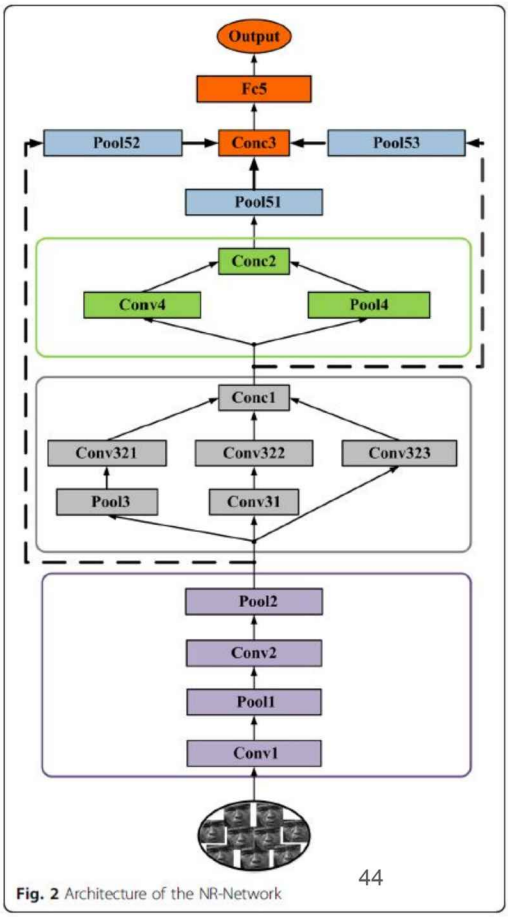


Fig. 2 Architecture of the NR-Network

Ding Y, Cheng Y, Cheng X, Li B, You X, Yuan X (2017) Noise-resistant network: a deep-learning method for face recognition under noise. EURASIP Journal on Image and Video Processing 2017(1):43

Park et al (2017)

- ✔ a method for learning feature representations
- ✔ which directly determine whether two input images are identical
- ✔ using a single model based on DCNN and residual learning

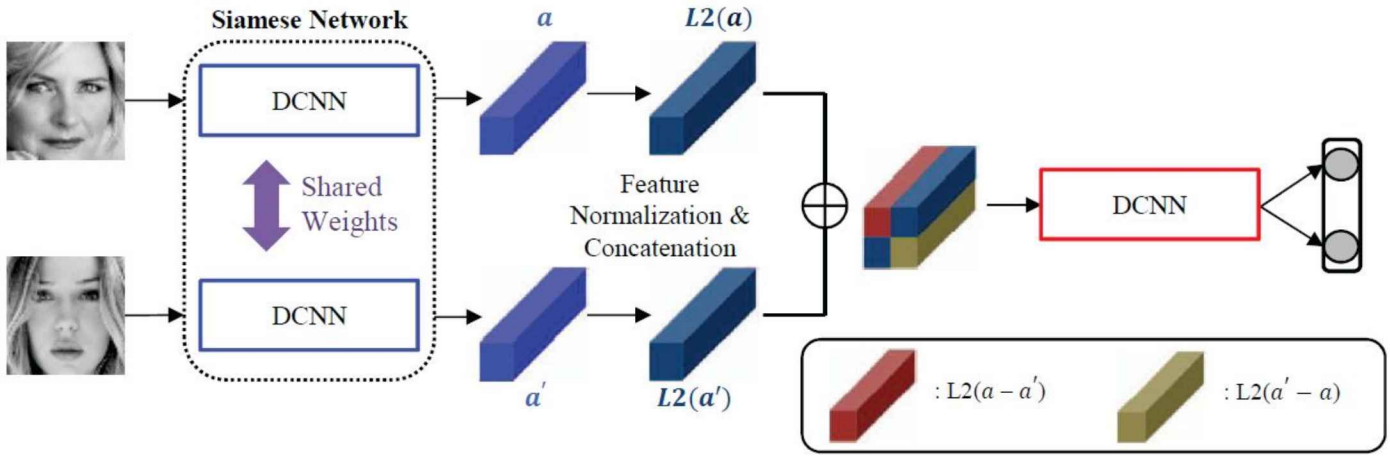


Figure 1. An overall model structure of the proposed method. The ***a*** and ***a'*** denote two extracted features from the siamese network. The L2(***a***) denotes that the vector ***a*** is L2-normalized.

Park S, Yu J, Jeon M (2017) Learning feature representation for face verification. In: Advanced Video and Signal Based Surveillance, Intl. Conf. on, IEEE, pp 1–6

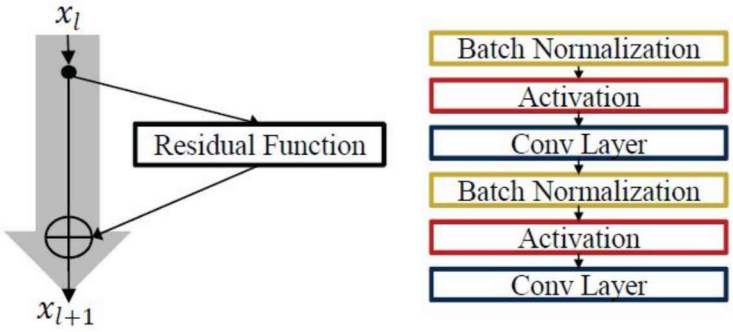


Figure 2. (Left) Residual Learning; (Right) Structure of Residual Function. The output is sum of a results of an identity mapping and residual function for the input.

Name	#Filters / Filter Size / Stride	Output Size	#Params
Conv1	1 / 5×5×1 / 1	96×96×32	1.6K
Pool1	1 / 2×2 / 2	48×48×32	-
Residual Block1	5 / 3×3×32 / 1 1 / 3×3×64 / 1	48×48×32 48×48×64	92K 36K
Pool2	1 / 2×2 / 2	24×24×64	-
Residual Block2	5 / 3×3×64 / 1 1 / 3×3×128 / 1	24×24×64 24×24×128	368K 148K
Pool3	1 / 2×2 / 2	12×12×128	-
Feature Concatenation	-	24×24×128	-
Residual Block3	3 / 3×3×128 / 1 1 / 3×3×256 / 1	24×24×128 24×24×256	442K 295K
Pool4	1 / 2×2 / 2	12×12×256	-
Residual Block4	3 / 3×3×256 / 1 1 / 3×3×384 / 1	12×12×256 12×12×384	1769K 885K
Pool5	1 / 2×2 / 2	6×6×384	-
Global Avg Pool	1 / 6×6 / 1	1×1×384	-
FC1	-	2	0.768K
Loss	-	2	-
Total	-	-	4037K

Table 1. The detailed architecture of the proposed model. The bold indicates layers in the siamese network. Therefore, we compute the number of parameters in the bold layers twice.

We express the residual unit in a general form:

$$\mathbf{x}_{l+1} = h(\mathbf{x}_l) + \mathcal{F}(\mathbf{x}_l, W_l) \quad (1)$$

Here $\mathbf{x}_l$, $\mathbf{x}_{l+1}$, and W_l are an input, output, and set of weights of l-th layer. The function $h(\mathbf{x}_l)$ performs an identity mapping with $h(\mathbf{x}_l) = \mathbf{x}_l$ and $\mathcal{F}(\mathbf{x}_l, W_l)$ denotes a residual function, which is composed of Batch Normalization layers [14], activation functions, and convolution layers.

Hsieh et al (2017)

Hsieh HL, HsuW, Chen YY (2017) Multi-task learning for face identification and attribute estimation. In: Acoustics, Speech and Signal Processing, Intl. Conf. on, IEEE, pp 2981–2985

- ✔ proposed a multi-task learning framework
- ✔ incorporate identity and high-level human attributes (gender, age)
- ✔ make use of multiple loss function by multi-task learning
- ✔ learn more semantic and discriminative face representations
- ✔ these ideas decrease the needed size of dataset and reduce the computation efforts
- ✔ In general, multi-task learning seek to improve the performance of multiple

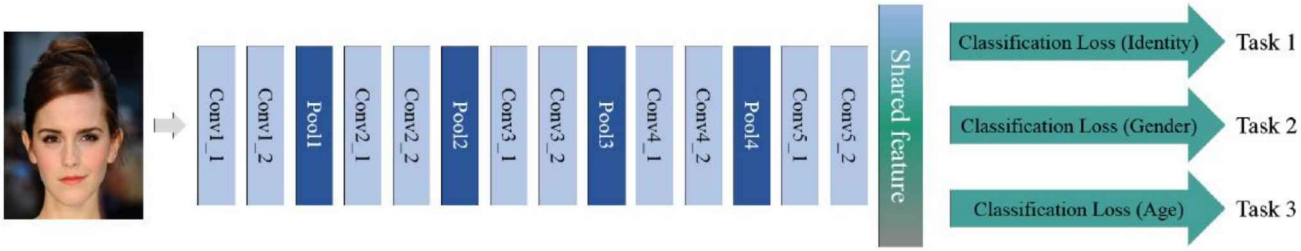


Fig. 2. Illustration of our CNN architecture. The shared features would be the input of multiple loss layers to classify human attributes and human identity. By utilizing the regularized multiple loss function, we can learn better representations for face identification and attribute detection.

Chen et al (2016a)

- ✔ used a joint Bayesian metric learning to assess the similarity between two face representations

✔For training:
- first perform face and landmark detection on the CASIA-WebFace, and the IJB-A datasets to localize and align each face.
- Next, train our DCNN on the CASIA-WebFace
- derive the joint Bayesian metric using the training sets of the IJB-A dataset and the DCNN features.

For testing:
- Then, given a pair of test image sets
- compute the similarity score based on their DCNN features and the learned metric.

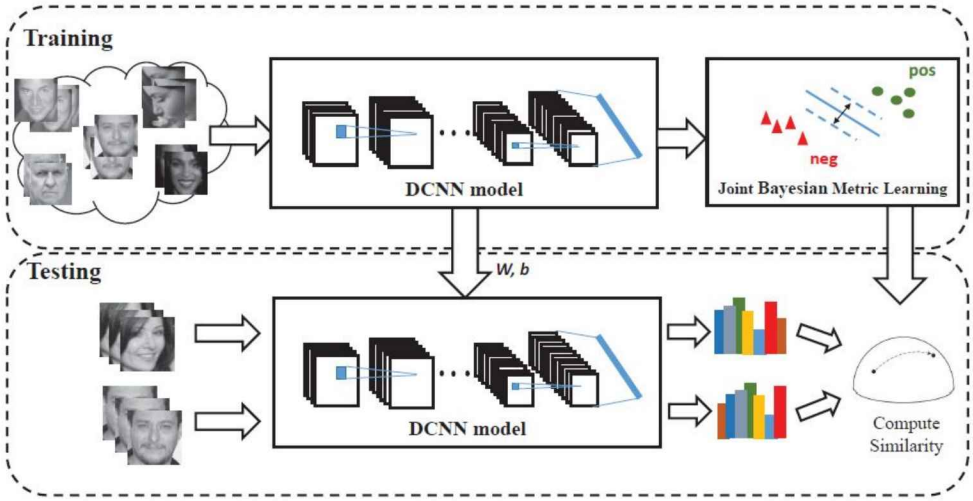


Figure 1. An overview of the proposed DCNN approach for face verification.

Chen JC, Patel VM, Chellappa R (2016a) Unconstrained face verification using deep cnn features. In: Applications of Computer Vision, Winter Conf. on, IEEE, pp 1–9

FV-DCNN (Chen et al, 2016b)

- ✔ combine the deeply learned feature by CNN and Fisher vector representation
- ✔ to generate Fisher vector encoded DCNN features
- ✔ which can capture both local and global variations
- ✔ Training:
  - Each training image is first passed through a pre-trained DCNN model to extract the convolutional features
  - Then, learn the Gaussian mixture model over them and perform FV encoding over these local convolutional features which have already encoded the rich face feature information
  - Finally, learn the metric
- ✔ Testing:
  - extract the DCNN features
  - use the learned GMM to perform FV feature encoding
  - apply the learned metric to compute the similarity scores

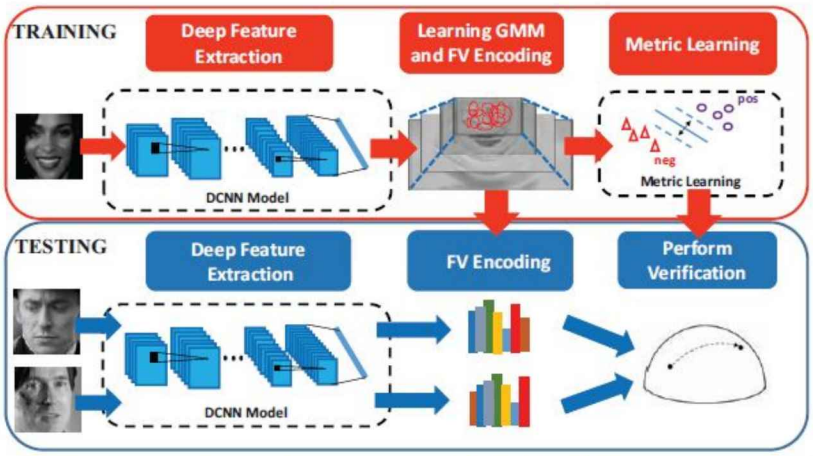


Fig. 1. An overview of the proposed FV-DCNN representation for unconstrained face verification.

Chen JC, Zheng J, Patel VM, Chellappa R (2016b) Fisher vector encoded deep convolutional features for unconstrained face verification. In: Image Processing, Intl. Conf. on, IEEE, pp 2981–2985

NormFace (Wang et al, 2017a)

- ✔ Propose two strategies for training using normalized features
  - use a modification of the softmax loss to optimize cosine similarity instead of inner-product
  - is a reformulation of metric learning by introducing an agent vector for each class

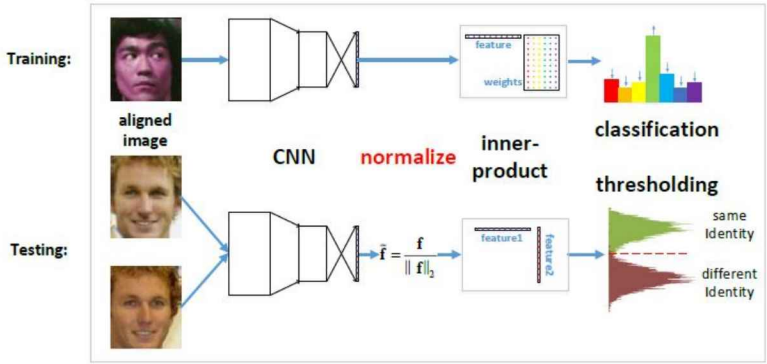


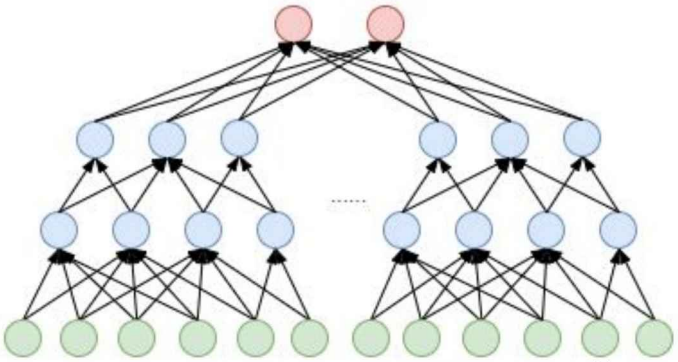
Figure 1: Pipeline of face verification model training and testing using a classification loss function. Previous works did not use the normalization after feature extraction during training. But in the testing phase, all methods used a normalized similarity, e.g. cosine, to compare two features.

Wang F, Xiang X, Cheng J, Yuille AL (2017a) Normface: *l* 2 hypersphere embedding for face verification. arXiv preprint arXiv:1704.06369

❖ Multi-CNN

- Some models use more than one CNN to extract features and concatenate them as the final features for face recognition
- They usually require additional training data to train each CNN
- It is necessary to explore some particular modalities that can contribute to enhance performance

- The use of the above strategies requires significant efforts in terms of data preparation or selection and computing resources



51

Table 2 Overview of deep learning methods based on Multi-CNN

Algorithm	Description/Remark
DeepID (Sun et al, 2014b)	Each CNN takes a face region as input; Features are concatenated from them; All identities are classified simultaneously
DeepID2 (Sun et al, 2014a)	An ensemble of 25 CNNs trained on different local patches; Apply Joint Bayesian to obtain robust embedding space; Use identification and verification signals as supervision
DeepID2+ (Sun et al, 2015b)	Based on DeepID2, further combine verification and identification loss
DeepID3 (Sun et al, 2015a)	Joint identification-verification supervision added in final and a few intermediate layers
Kang et al (2017)	Based on Multi-scale Convolution Layer Blocks (MCLBs); Stack MCLBs to present multi-scale abstraction; Use a deep ensemble; Extract two types of features from each DCNN and combine them to do FR
SIAMESE (Wang et al, 2014)	Trained on different parts and scales of a face using a layer-wise training method; All face representations are concatenated as feature
FR+FCN (Zhu et al, 2014b)	Contain five CNNs; Each takes a pair of whole faces or facial components (forehead, eye, nose and mouth) as input; Five CNNs are concatenated by fully connected layer to learn feature representation; Use a logistic regression layer to predict whether the two face images belong to the same identity
Baidu (Liu et al, 2015)	A two-stage approach combining multi-path deep CNN and deep metric learning; Extract overlapped image patches centered at different landmarks on face region; Concatenates representation together forming a high dimensional feature
MFRS (Zhou et al, 2015)	4 face regions are cropped for feature extraction and PCA for feature reduction
Xiong et al (2017)	Explore complementarity of 2 DCNNs by training with two different large datasets
Bodla et al (2017)	A deep heterogeneous feature fusion network for template-based face recognition
Lu et al (2017b)	2 CNNs; Concatenate features of each CNN after PCA reduction

SIAMESE (Wang et al, 2014)

- ✔ use Siamese Network based on normal convolutional neural network
- ✔ trained on different parts and scales of faces using a layer-wise training method
- ✔ all face representations are concatenated as the feature

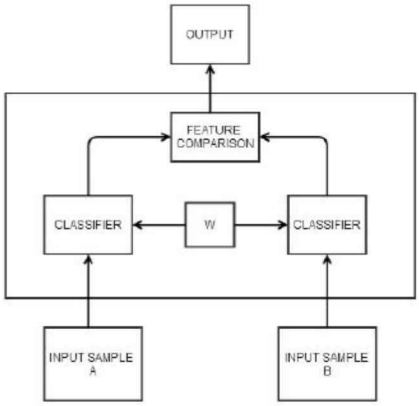


Fig. 1. Siamese Network

Fig.1 is Siamese Network of Probability, which supports the module of $y = f(X_1, X_2)$. X_1, X_2 is a vector of the actual problem, y is their similar probability. Using Siamese Network Module can solve multiple sample input and classification problem.

Wang W, Yang J, Xiao J, Li S, Zhou D (2014) Face recognition based on deep learning. In: Intl. Conf. on Human Centered Computing, Springer, pp 812–820

FR+FCN (Zhu et al, 2014b)

- ✔ can directly recover canonical views of 2D face images using multiple CNNs
- ✔ It first selects canonical view faces by a facial measurement for frontal view based face rank
- ✔ And recovers faces using a 4-layer CNN
- ✔ Then facial component-based CNN is used to train with the recovery faces

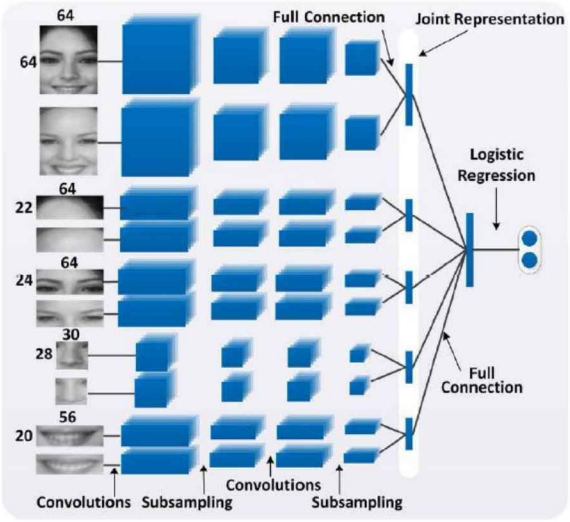


Figure 7: Architecture of the facial component-based network. The network contains five CNNs, each of which takes a pair of whole faces or facial components as input. The sizes of the whole face, forehead, eye, nose, and mouth are 64×64, 22×64, 24×64, 28×30, and 20×56, respectively. First, each CNN learns the joint representation of the pairs of input. A logistic regression layer then concatenates all the joint representations as features to predict whether the two face images belong to the same identity.

Zhu Z, Luo P, Wang X, Tang X (2014b) Recover canonical-view faces in the wild with deep neural networks. arXiv preprint arXiv:14043543 Zou W, Zhu S, Yu K

MFRS (Zhou et al,2015)

✔ four face regions are cropped for the representation extraction

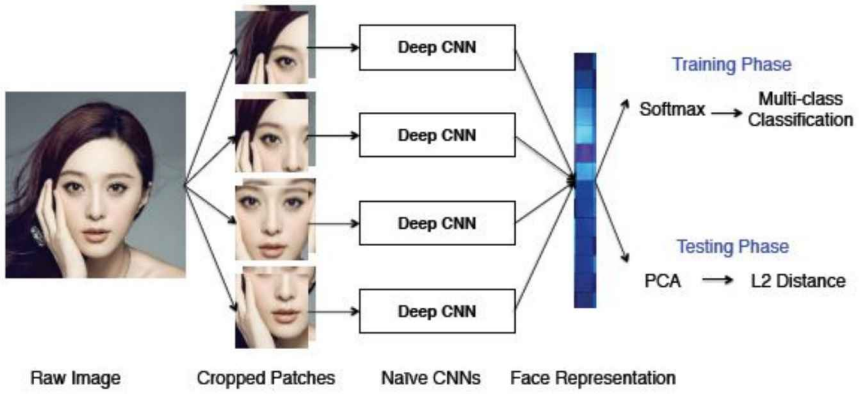


Figure 3. **Overview of Megvii Face Recognition System.** We design a simple 10 layers deep convolutional neural network for recognition. Four face regions are cropped for representation extraction. We train our networks on the MFC database under the traditional multi-class classification framework. In testing phase, a PCA model is applied for feature reduction, and a simple L2 norm is used for measuring the pair of testing faces.

Zhou E, Cao Z, Yin Q (2015) Naive-deep face recognition: Touching the limit of lfw benchmark or not? arXiv preprint arXiv:150104690

DeepID series methods

✔ DeepID, DeepID2, DeepID2+, DeepID3

✔ extract robust features of different local face patches

✔ Combine them together for recognition

✔ Identification and/or verification signals are adopted for supervision

DeepID (Deep hidden Identity features)

- ✔ extract from 60 face patches with ten regions, three scales, and RGB or gray channels
- ✔ form complementary and over-complete representations

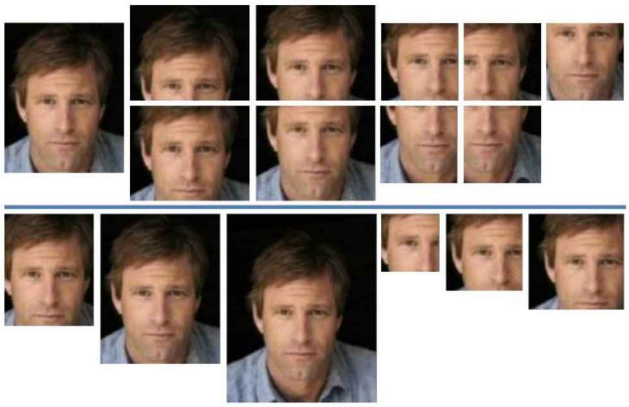


Figure 3. Top: ten face regions of medium scales. The five regions in the top left are global regions taken from the weakly aligned faces, the other five in the top right are local regions centered around the five facial landmarks (two eye centers, nose tip, and two mouse corners). Bottom: three scales of two particular patches.

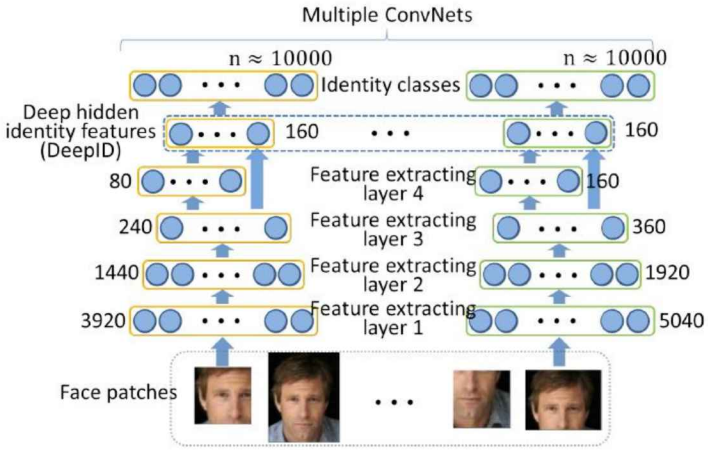


Figure 1. An illustration of the feature extraction process. Arrows indicate forward propagation directions. The number of neurons in each layer of the multiple deep ConvNets are labeled beside each layer. The DeepID features are taken from the last hidden layer of each ConvNet, and predict a large number of identity classes. Feature numbers continue to reduce along the feature extraction cascade till the DeepID layer.

Sun Y, Wang X, Tang X (2014b) Deep learning face representation from predicting 10,000 classes. In: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, pp 1891–1898

Deep ConvNets:

- ✔ contain 4 convolutional layers (with max-pooling) to extract features hierarchically
- ✔ followed by the fully-connected DeepID layer
- ✔ the softmax output layer indicating identity classes
- ✔ The last hidden layer is fully connected to both the 3th and 4th convolutional layers (after maxpooling) such that it sees multi-scale features (features in the 4th convolutional layer are more global than those in the 3th one).

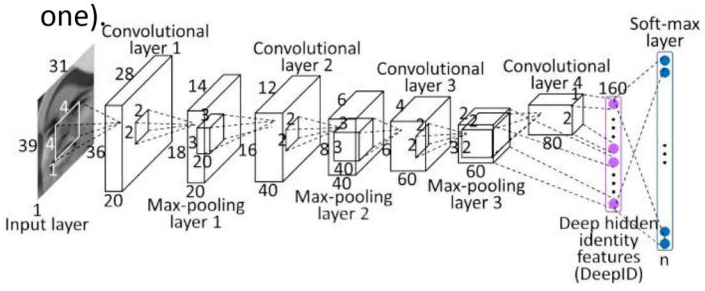


Figure 2. ConvNet structure. The length, width, and height of each cuboid denotes the map number and the dimension of each map for all input, convolutional, and max-pooling layers. The inside small cuboids and squares denote the 3D convolution kernel sizes and the 2D pooling region sizes of convolutional and max-pooling layers, respectively. Neuron numbers of the last two fully-connected layers are marked beside each layer.

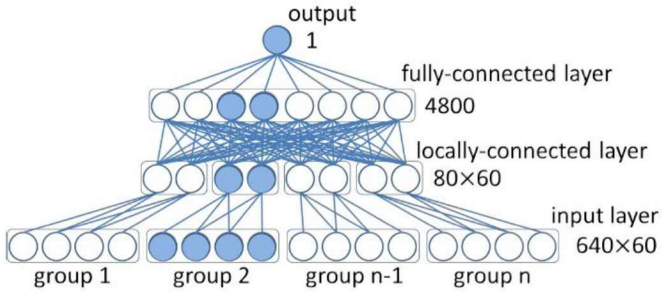


Figure 4. The structure of the neural network used for face verification. The layer type and dimension are labeled beside each layer. The solid neurons form a subnetwork.

- The input features: DeepID
  - divided into 60 groups
  - each contains 640 features extracted from a particular patch pair with a particular ConvNet
  - Features in the same group are highly correlated
- Neurons in the locally-connected layer only connect to a single group of features to learn their local relations and reduce the feature dimension at the same time
- The second hidden layer is fully-connected to the first hidden layer to learn global relations
- The single output neuron is fully connected to the second hidden layer; Output indicates face similarities.

DeepID2 (Deep Identification-verification features)

- Take similar structures as in DeepID
- Get a 160-dimensional DeepID2 feature vector at its DeepID2 layer
- The DeepID2 layer to be learned are fully-connected to both the 3rd and 4th convolutional layers
- However, it use identification and verification signals as supervision

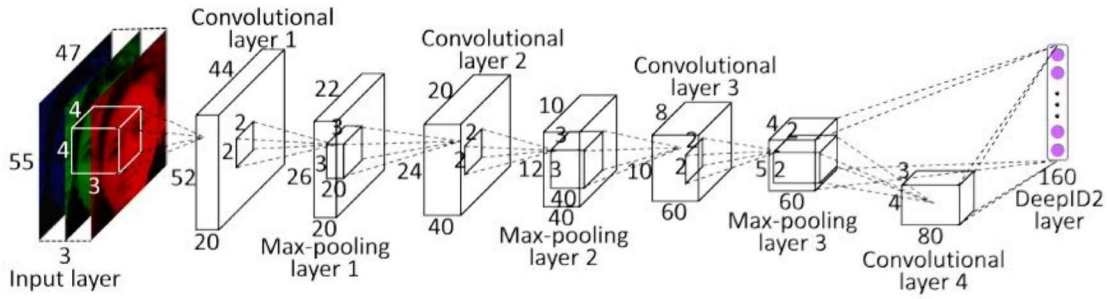


Figure 1: The ConvNet structure for DeepID2 feature extraction.

Sun Y, Chen Y,Wang X, Tang X (2014a) Deep learning face representation by joint identification-verification. In: Advances in neural information processing systems, pp 1988–1996

DeepID2+

- Inherited from DeepID2
- However, make three improvements:
  - ✔ First, DeepID2+ nets are larger
    - with 128 feature maps in each of the 4 convolutional layers
    - The final feature representation is also increased to 512 dimensions
  - ✔ Second, our training data is enlarged
    - trained with around 290, 000 face images from 12, 000 identities
    - DeepID2: 160, 000 images from 8, 000 identities
  - ✔ Third, enhance the supervision
    - connect a 512-dimensional fully-connected layer to each of the four convolutional layers
    - supervise these 4 fully-connected layers with the identification-verification supervisory signals simultaneously

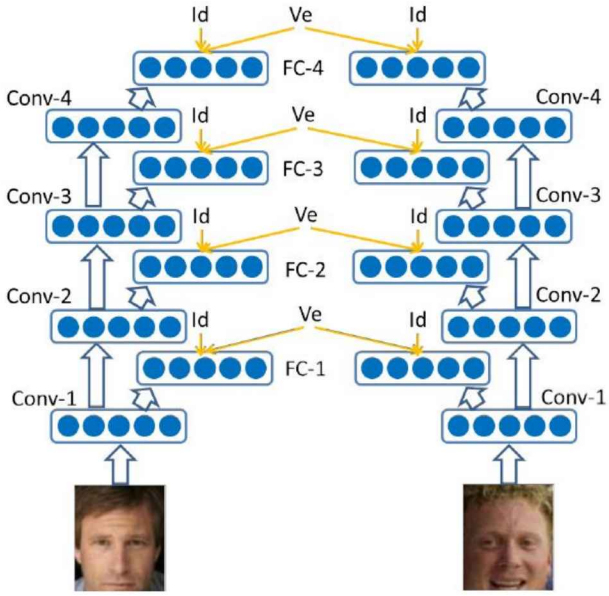
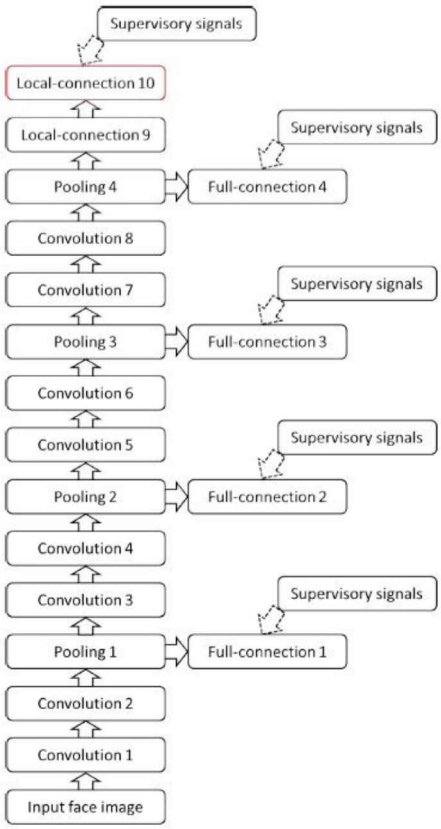


Figure 2: DeepID2+ net and supervisory signals. Conv-n deneotes the n-th convolutional layer (with max-pooling). FC-n denotes the n-th fully connected layer. Id and Ve denote the identification and verification supervisory signals. Blue arrows denote forward-propagation. Yellow arrows denote supervisory signals. Nets in the left and right are the same DeepID2+ net with different input faces.

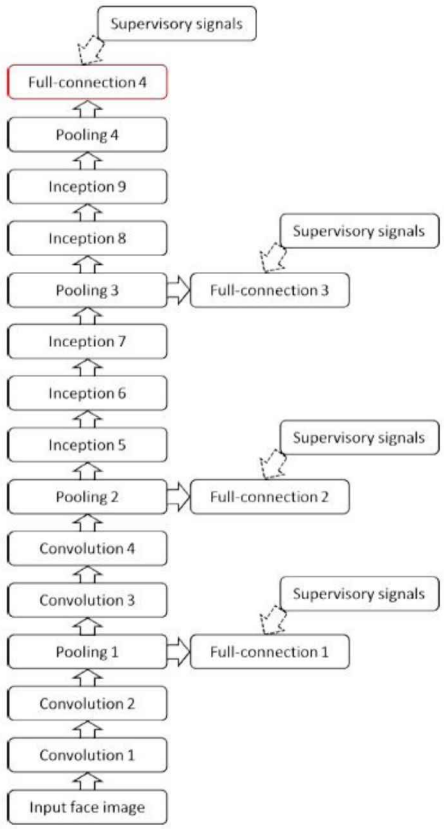
Sun Y, Wang X, Tang X (2015b) Deeply learned face representations are sparse, selective, and robust. In: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, pp 2892–2900

DeepID3

- inherits a few characteristics of the DeepID2+ net
  - ✔ unshared neural weights in the last few feature extraction layers
  - ✔ the way of adding supervisory signals to early layers
- DeepID3 net is significantly deeper
  - ✔ with 10 to 15 non-linear feature extraction layers
  - ✔ DeepID2+: 5
- propose two DeepID3 net architectures
  - ✔ DeepID3 net1
  - ✔ DeepID3 net2
- The depth of DeepID3 net is due to stacking multiple convolution/inception layers before each pooling layer.
- Continuous convolution/inception helps to form features with larger receptive fields and more complex nonlinearity while restricting the number of parameters



DeepID3 net1



DeepID3 net2

Sun Y, Liang D, Wang X, Tang X (2015a) Deepid3: Face recognition with very deep neural networks. arXiv preprint arXiv:150200873

Hu et al (2017a)

- ✔ introduced facial attribute feature (FAF) into face recognition
- ✔ and fused it with face recognition features (FRF)
- ✔ to enhance face recognition performance in various challenging scenarios.

Hu G, Hua Y, Yuan Y, Zhang Z, Lu Z, Mukherjee SS, Hospedales TM, Robertson NM, Yang Y (2017a) Attribute-enhanced face recognition with neural tensor fusion networks. In: Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition, pp 3744–3753

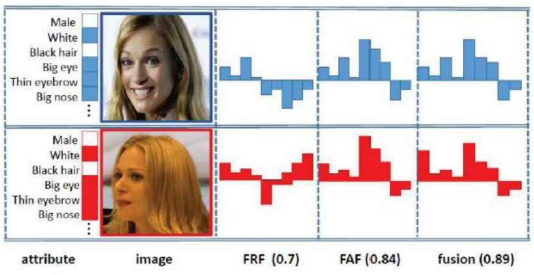
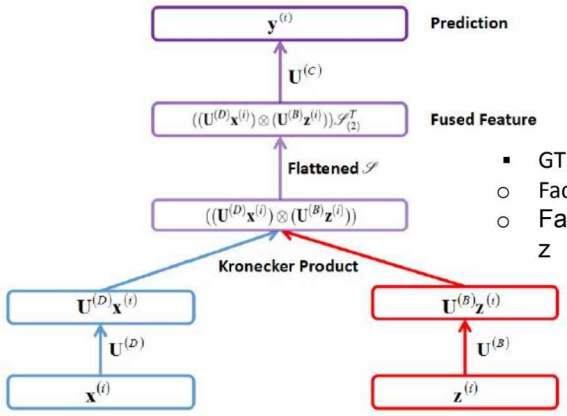


Figure 1: A sample attribute list is given (*col.1*) which pertains to the images of the same individual at different poses (*col.2*). While the similarity scores for each dimension vary in the face recognition feature (FRF) set (*col.3*), the face attribute feature (FAF) set (*col.4*) remains very similar. The fused features (*col.5*) are more similar and a higher similarity score (0.89) is achieved.



- GTNN
  - Face Recognition Feature: x
  - Facial Attribute Feature: z

Figure 2: Gated two-stream neural network to implement low-rank tensor-based fusion. The architecture computes Eq. (7), with the Tucker decomposition in Eq. (4). The network is identity-supervised at train time, and feature in the fusion layer used as representation for verification.

✔ Integration with CNNs for FR: architecture

- LeanFace
  - use a large number of convolutional layers at early stage to capture the subtle low level and mid-level information
  - activation function is maxout
  - Joint supervision of softmax loss and center loss is used for training
- AttNet
  - To detect facial attributes, uses the architecture of Lighten CNN [50] to represent a face.
- Once trained, the features extracted from the penultimate fully-connected layers of LeanFace (256D) and AttNet are extracted as x and z, and input to GTNN for fusion and then face recognition.

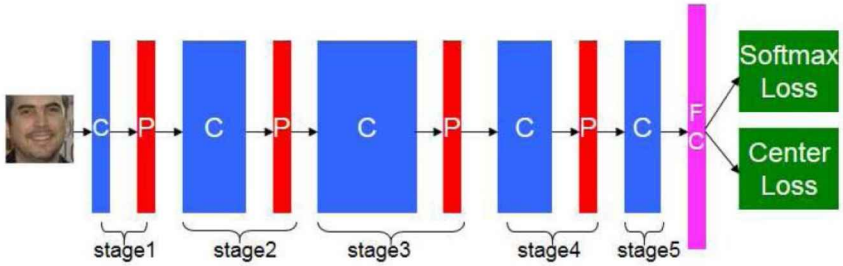


Figure 3: LeanFace. 'C' is a group of convolutional layers. Stage 1: 64 @ 5×5 (64 feature maps are sliced to two groups of 32 ones, which are fed into maxout function.) ; Stage 2: 64 @ 3×3, 64 @ 3×3, 128 @ 3×3, 128 @ 3×3; Stage 3: 196 @ 3×3, 196 @ 3×3, 256 @ 3×3, 256 @ 3×3, 320 @ 3×3, 320 @ 3×3; Stage 4: 512 @ 3×3, 512 @ 3×3, 512 @ 3×3, 512 @ 3×3; Stage 5: 640 @ 5×5, 640@ 5×5. 'P' stands for 2×2 max pooling. The strides for the convolutional and pooling layers are 1 and 2, respectively. 'FC' is a fully-connected layer of 256D.

Kang et al (2017)

✔ designed a face recognition system based on Multi-scale Convolution Layer Blocks (MCLBs)

✔ It stacks MCLBs to present multi-scale abstraction

✔ and uses a deep ensemble for it.

✔ Two types of features, low dimensional but discriminative feature and high-level abstracted feature, are extracted from each deep CNN

✔ and combined together for FR

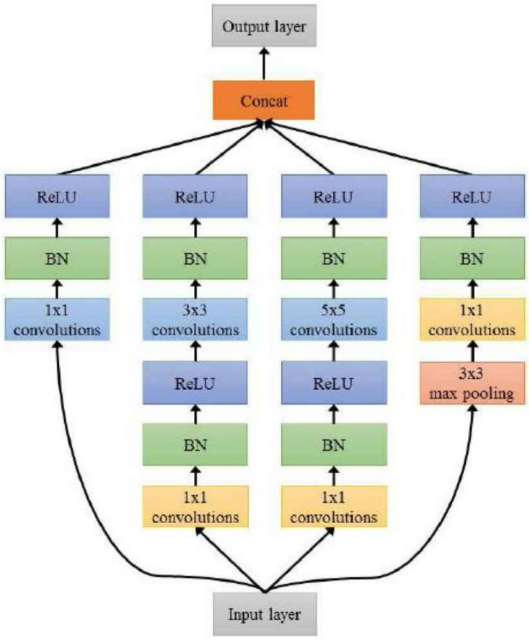


Figure 1: The multi-scale convolution layer block consists of 1×1 convolution, 3×3 convolution, 5×5 convolution, and 3×3 max pooling layers. The function of 1×1 convolution is the dimension reduction; the 3×3 and 5×5 convolutions process at different scales to achieve multi-scale feature abstraction;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

Kang BN, Kim Y, Kim D (2017) Deep convolutional neural network using triplets of faces, deep ensemble, and score-level fusion for face recognition. In: Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition Workshops, pp 109–116

Xiong et al (2017)

✔ Inspired by transfer learning

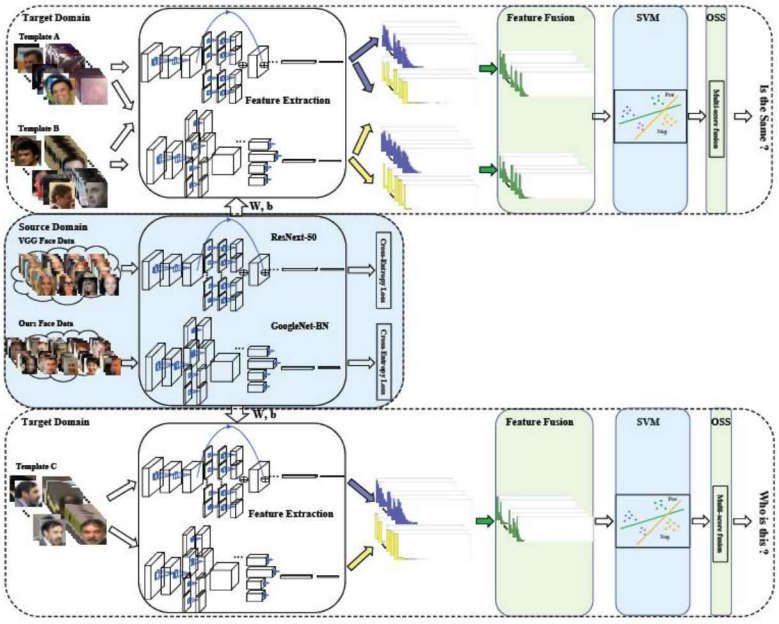


Fig. 2: Framework overview. Our learning framework consists three components: Deep feature learning module locates middle component, Template-based unconstrained face recognition is included in upper and lower components. Training procedures are illustrated with blue blocks, two-stage fusion is depicted in green blocks. Best viewed in color.

✔proposed a unified learning framework, transferred deep feature fusion, to explore the complementarity of two distinct DCNNs

✔by training them with two different large datasets.

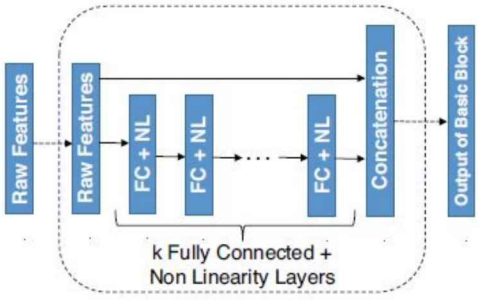
✔When feature extraction is finished, the model will fuse the two types of deep features and adopt specific linear SVMs for classification.

Xiong L, Karlekar J, Zhao J, Feng J, Pranata S, Shen S (2017) A good practice towards top performance of face recognition: Transferred deep feature fusion. arXiv preprint arXiv:1704.00438

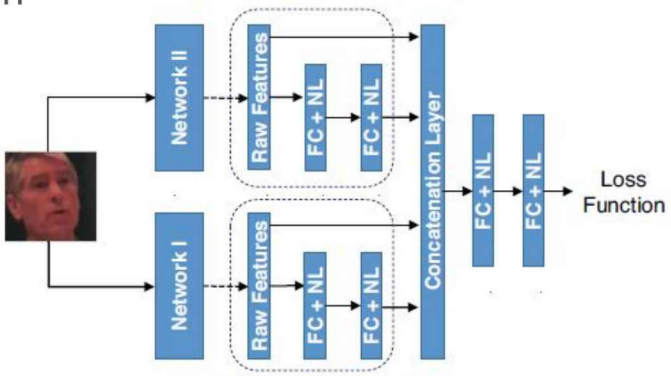
Bodla et al (2017)

✔ proposed a deep heterogeneous fusion network that fuses two deep features generated by different DCNNs

✔ for template-based face recognition by exploiting the complementary information presented in features



(a) Basic block of fusion network



(b) Deep fusion network

Figure 2. The figure on top shows our basic building block of deep fusion network with k hidden layers. The bottom figure is the overview of our deep fusion network that fuses two DCNN features. The boundaries shown in dotted lines are the two basic blocks corresponding to deep networks I [28] and II [29].

Bodla N, Zheng J, Xu H, Chen JC, Castillo C, Chellappa R (2017) Deep heterogeneous feature fusion for template-based face recognition. In: Applications of Computer Vision, Winter Conf. on, IEEE, pp 586–595

❖ Variants of CNN

Based on general CNN, some variants of CNN have been proposed

Table 3 Overview on some variants of CNN

Algorithm	Description/Remark
BCNN (Lin et al, 2015)	To bridge the gap between the texture models and part-based CNN models
Chowdhury et al (2015)	Fine-tune a trained base-model of a symmetric BCNN to extract feature
Chowdhury et al (2016)	Apply BCNN on IJB-A dataset
Pyramid CNN (Fan et al, 2014)	Contain a group of CNNs divided into several levels with different depth and size, and they share some of the layers
c-CNN (Xiong et al, 2015)	The samples in c-CNN are processed with dynamically activated sets of kernels; Kernels are only sparsely activated when a sample is passed through the network
Li et al (2015a)	A tree-structure kernel adaptive CNN; Can heirarchically fuse multiple local adaptive CNN subnets
Guided-CNN (Fu et al, 2017)	Parallel sub-CNN models as guide and learners
PCANet (Chan et al, 2015)	PCA is employed to learn multistage filter banks
SPCANet (Tian et al, 2015a)	Stack multiple output features learned through each stage of the CNN as the input of nonlinear processing layer with hashing method-activation
SRDANet (Tian et al, 2015b)	Use leading eigenvectors from patches in facial image as filter kernels
Weighted-PCANet (Huang and Yuan, 2015)	Combine Linear Regression Classification model and PCANet construction to extract feature
MS-PCANet (Tian et al, 2016)	Multiscaled PCA Network
Simón et al (2016)	Fuse CNN and WNNC
NAN (Yang et al, 2016)	Two modules: CNN based feature embedding and neural aggregation
ABTA (Dong et al, 2017)	Two modules: attention based neural network, template adaptation module
Ranjan et al (2016)	Employ a multi-task learning (MTL) framework to do multi-purpose task
SL-DCNN (Chen and Deng, 2016)	Weakly-supervised self-learning DCNN
LBPNet (Xi et al, 2016)	An unsupervised learning; Trainable kernels are replaced by LBP
JFL (Lu et al, 2015)	Stack an unsupervised feature learning method into a deep CNN
Chen et al (2015b)	An automatic end-to-end FR system: face detection, alignment and verification
Wu et al (2017a)	ReST is introduced into CNN to do face alignment and recognition
Ranjan et al (2016)	Multi-purpose CNN architecture; Can simultaneously perform various tasks

▯ B-CNN (Chowdhury et al, 2016)

- ✔ B-CNN is originally introduced by Lin et al (2015)
- ✔ applied B-CNN to IJB-A
- ✔ consists of two CNNs whose convolutional-layer outputs are multiplied (using outer product) at each location of the image
- ✔ The resulting bilinear feature is pooled across the image resulting in an orderless descriptor for the entire image.
- ✔ This vector can be normalized to provide additional invariances

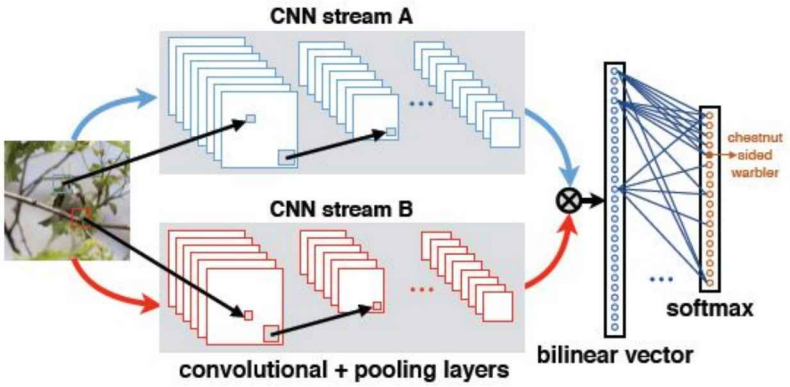


Fig. 1. **Image classification using a B-CNN.** An image is passed through CNNs A and B, and their outputs at each location are combined using the matrix outer product and average pooled to obtain the bilinear feature representation. This is passed through a linear + softmax layer to obtain class predictions.

Chowdhury AR, Lin TY, Maji S, Learned-Miller E (2016) One-tomany face recognition with bilinear cnns. In: Applications of Computer Vision, Winter Conf. on, IEEE, pp 1–9

Pyramid CNN (Fan et al, 2014)

- ✔ presented a structure of DCNN, using image pixels as input and multiscale input patches.
- ✔ There are multiple CNNs.
- ✔ For each CNN, two images are fed and the SIAMESE network is used to train it.
- ✔ The outputs are compared by the output neurons which predict whether the two face images have the same identity.
- ✔ Output is a landmark-based multi-scale feature with a highly compact characteristic

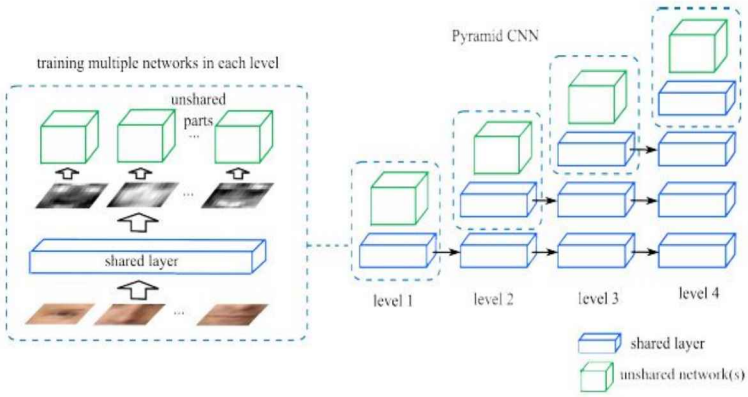


Figure 2: Our Pyramid CNN method. To train an individual network, the "Siamese" network is used. Two images are fed to the same CNN and the outputs are compared by the output neuron which predicts whether the two faces have the same identity. The Pyramid CNN consists of several levels of networks. These networks have different depths and input sizes, and they share some of the layers. The pyramid is trained in a greedy manner. The first network is trained on part of the face, and its first layer is fixed. The fixed layer is used to filter and down-sample input to higher level networks. Higher level networks are trained on the processed images. In this way, the size of the network that is actually trained does not grow as the level increases.

Fan H, Cao Z, Jiang Y, Yin Q, Doudou C (2014) Learning deep face representation. arXiv preprint arXiv:14032802

Conditional CNN (Xiong et al, 2015)

- ✔ c-CNN is a conditional CNN to handle multimodal face recognition.
- ✔ The activations of kernels for each layer are conditioned on the present intermediate representation and the activation status in lower layers.
- ✔ Activated kernels define sample-specific adaptive routes revealing distribution of the underlying modalities.

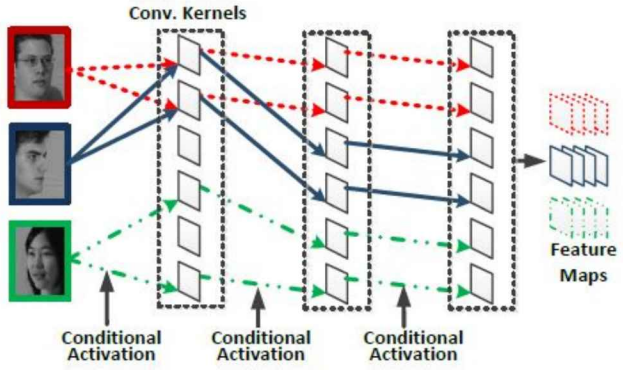


Figure 1. Illustration of c-CNN. Each line type stands for one modality. Each image is passed along with a modality-specific route indicated by the corresponding colored arrows. Only the kernels along the route are activated and utilized to extract features. The passing route defines the splitting w.r.t. inherent modalities in a coarse-to-fine manner: similar modalities, e.g., modality of red dashed line and blue solid line, may share certain kernels at the beginning layers.

Xiong C, Zhao X, Tang D, Jayashree K, Yan S, Kim TK (2015) Conditional convolutional neural network for modality-aware face recognition. In: Proceedings of the IEEE Intl. Conf. on Computer Vision, pp 3667–3675

Li et al (2015a)

- ✔ proposed a tree-structured convolutional architecture to deeply integrate the face representation of local subnetworks
- ✔ convolutional kernels are dynamically determined according to spatial distribution of facial landmarks

Li S, Xing J, Niu Z, Shan S, Yan S (2015a) Shape driven kernel adaptation in convolutional neural network for robust facial traits
recognition. In: Proceedings of the IEEE Conf. on Computer Vision
and Pattern Recognition, pp 222–230

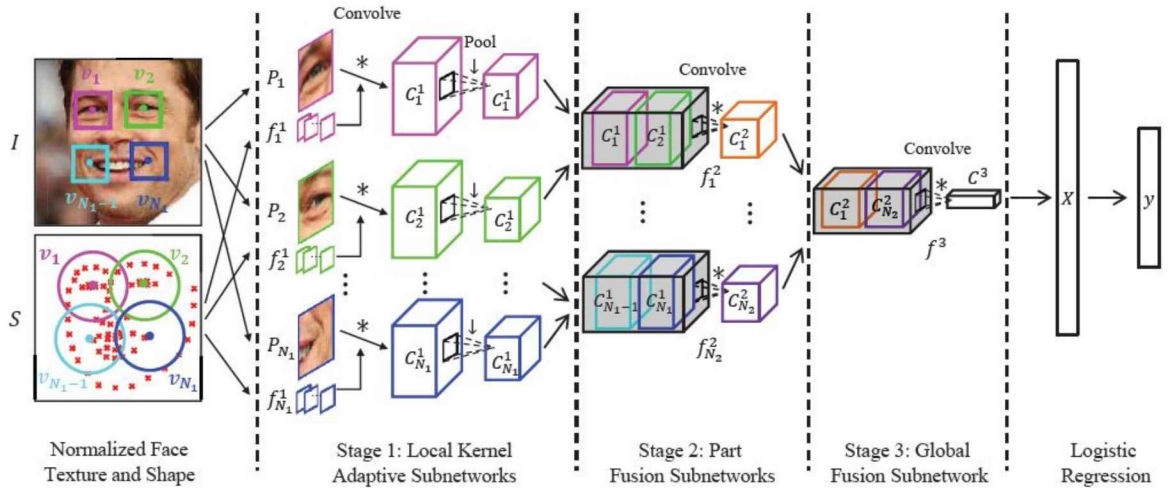


Figure 2. Flowchart of the proposed tree-structured kernel adaptive CNN. Given a normalized face image I and corresponding facial landmarks $S = \{v_i\}_{i=1}^{N_1}$, multiple local kernel adaptive CNN subnetworks $\{C_i^1\}_{i=1}^{N_1}$ are constructed to learn features from multiple local patches $\{P_i\}_{i=1}^{N_1}$. The convolved features learned by multiple local subnetworks are then combined as the middle-level representations to learn high-level features with the fusion subnetworks, i.e. multiple part fusion subnetworks $\{C_i^2\}_{i=1}^{N_2}$ and a global fusion subnetwork C^3. Finally, a logistic regression layer is used to generate the final prediction y from vectorized high level convolved features X.

Guided-CNN (Fu et al, 2017)

- ✔ a deep-learning based architecture of *Guided-CNN*
- ✔ which can be applied for cross-domain FR and beyond
- ✔ By utilizing an existing CNN-based face recognition model as a guide, we adapt and learn a parallel CNN model for dealing with face images with insufficient resolution
- ✔ the proposed Guided-CNN can be viewed as a deep domain adaptation model for relating HR and LR face images with recognition guarantees

Fu TC, Chiu WC, Wang YCF (2017) Learning guided convolutional neural networks for cross-resolution face recognition

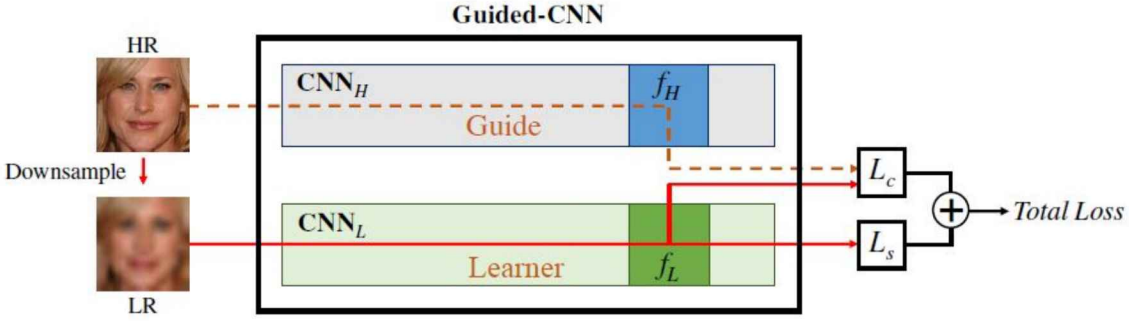


Fig. 2. The architecture of our proposed Guided-CNN for cross-resolution face recognition. During training, we downsample the HR images to be the LR inputs of CNN$_L$. Softmax loss L_s is applied for learning the identification information of LR images and our unique cross-domain loss L_c associates the feature representations of LR images to the corresponding HR ones. In testing phase, we simply input HR/LR to CNN$_H$/CNN$_L$ respectively and calculate the similarities using the cosine distance of the associated features f_H/f_L.

PCANet (Chan et al, 2015)

✔ combined principle component analysis (PCA) with deep neural networks to learn kernels

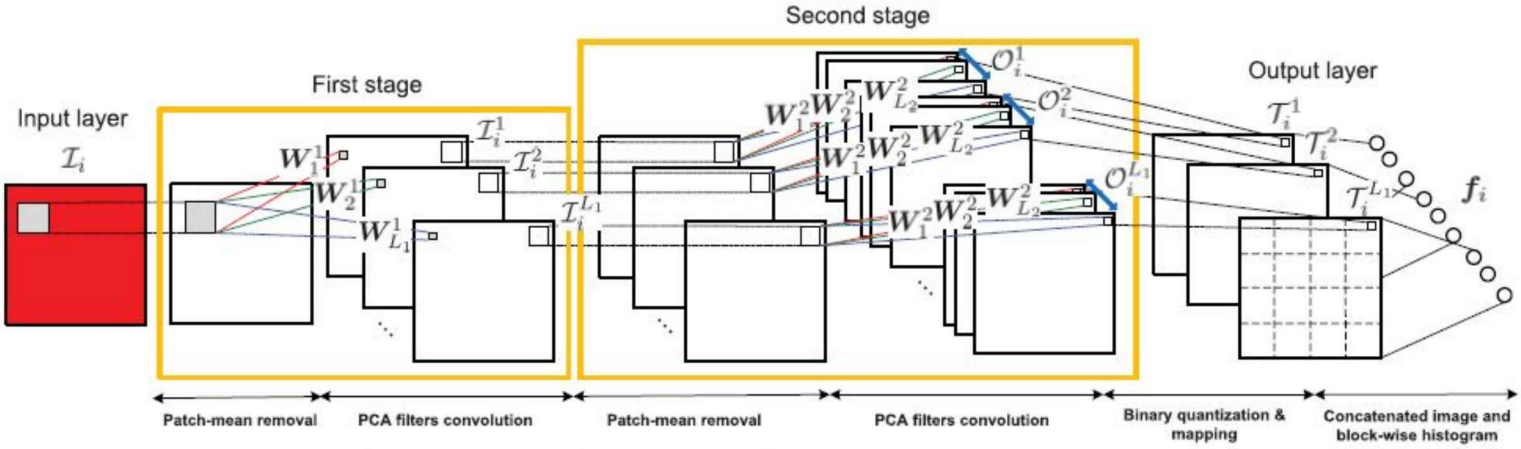


Fig. 2. A detailed block diagram of the proposed (two-stage) PCANet.

Chan TH, Jia K, Gao S, Lu J, Zeng Z, Ma Y (2015) Pcanet: A simple deep learning baseline for image classification? IEEE trans on Image Processing 24(12):5017–5032

SPCANet (Tian et al, 2015a)

✔ a simplified version of CNN, Stacked PCA Neural Network, follows the basic architecture of CNN

✔ Learn filter kernels go through PCA instead of SGD, then perform the nonlinear computation of the output of convolutional layer by hashing method and pooling the decimal-valued image using block-wise histogram technique in each stage

✔stack the output of multiple stages as final feature

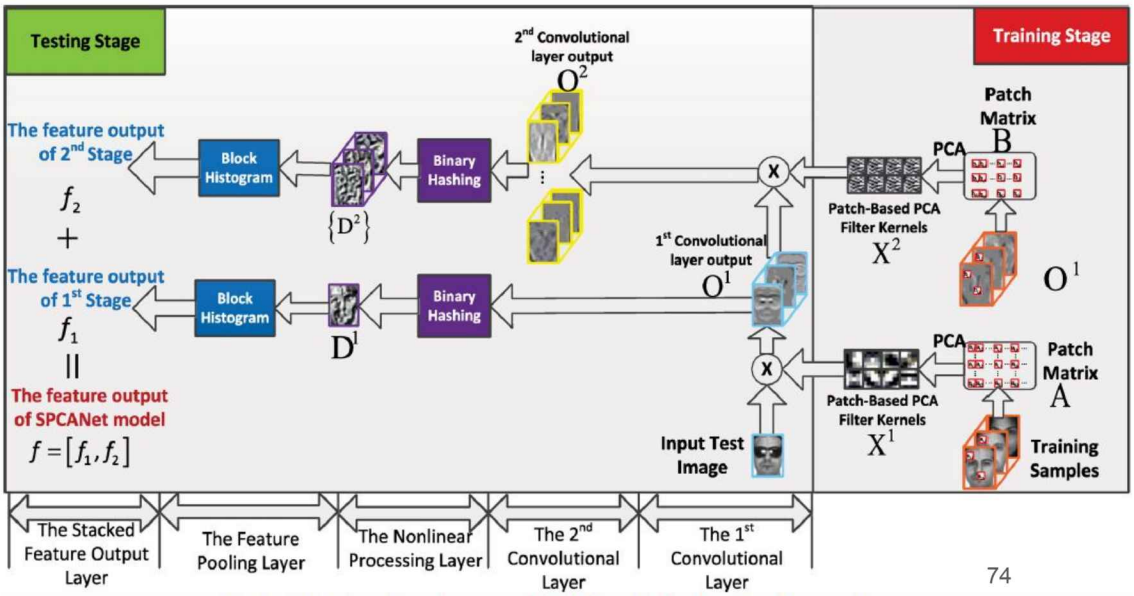


Fig. 1. Illustration of how the proposed SPCANet train filter kernels and extracts feature.

Tian L, Fan C, Ming Y, Jin Y (2015a) Stacked pca network (spcanet): an effective deep learning for face recognition. In: Digital Signal Processing, Intl. Conf. on, IEEE, pp 1039–1043

Weighted-PCANet (Huang and Yuan, 2015)

- ✔ learns features by combining Linear Regression Classification (LRC) model and PCANet construction,
- ✔ shares the main construction characteristics with classical CNNs as a cascaded neural network, including convolution layers and pooling layers
- ✔ The advantage of CNNs is taken in the design of weighted-PCANet, such as the utility of local vision fields
- ✔ However, the traditional back propagation (BP) algorithms in CNNs are replaced with a solved optimization problem.

Huang J, Yuan C (2015) Weighted-pcanet for face recognition. In: Intl. Conf. on Neural Information Processing, Springer, pp 246–254

MS-PCANet (Tian et al, 2016)

- ✔ contains two convolutional layers to extract features hierarchically
- ✔ followed by a nonlinear processing layer by using a simple binary hashing and feature pooling layer
- ✔ It uses PCA to get the prefixed filter kernels

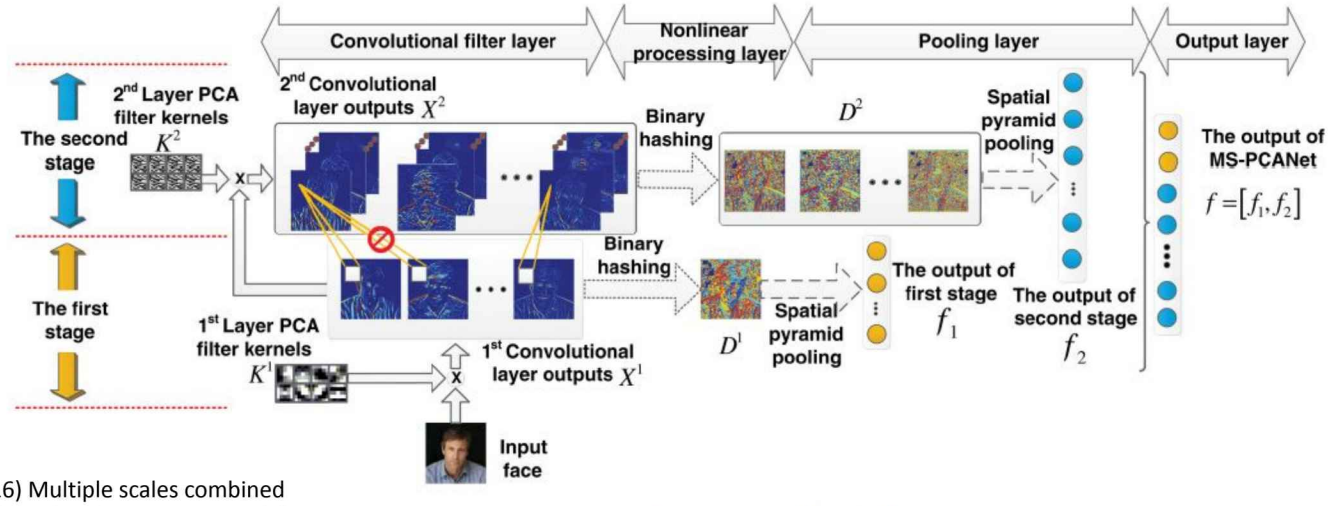


Fig. 1 An illustration of the architecture of our MS-PCANet model.

Tian L, Fan C, Ming Y (2016) Multiple scales combined principle component analysis deep learning network for face recognition. Journal of Electronic Imaging 25(2):023,025–023,025

SRDANet (Tian et al, 2015b)

✔ Spectral Regression Discriminant Analysis Network is similar to SPCANet, but it uses eigenvectors as filter kernels

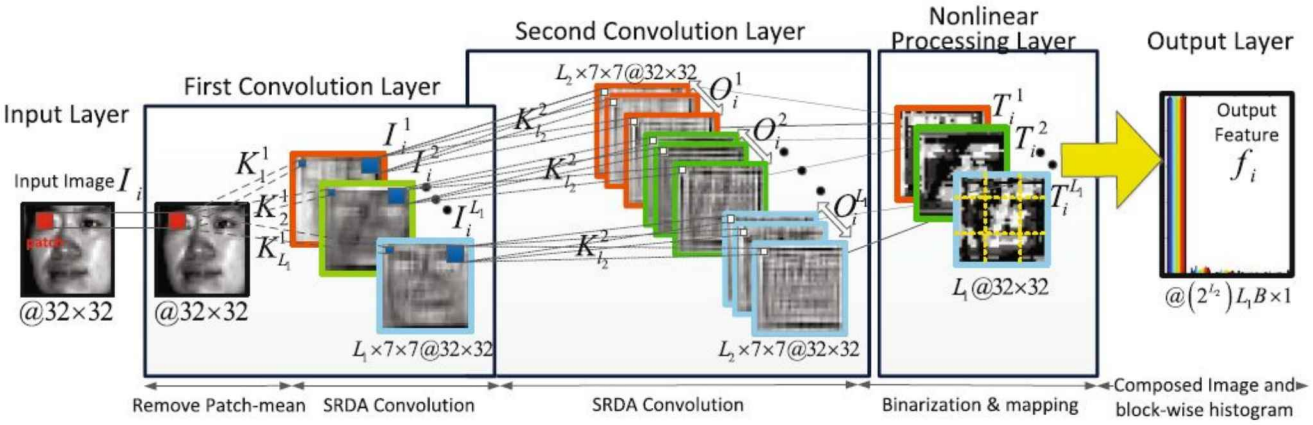


Fig. 1. The detailed layer architecture diagram of the two-stage SRDANet

Tian L, Fan C, Ming Y, Shi J (2015b) Srdanet: an efficient deep learning algorithm for face analysis. In: Intl. Conf. on Intelligent Robotics and Applications, Springer, pp 499–510

fusion of CNN with some other modules

Weighted Nearest Neighbour Classifier (WNNC), Simon et al (2016)

✔RGB, depth and thermal captures of the face are used for training CNNs for a binary classification.

✔The results are then fused with Histograms of Gabor Ordinal Measures (HOGOMs)

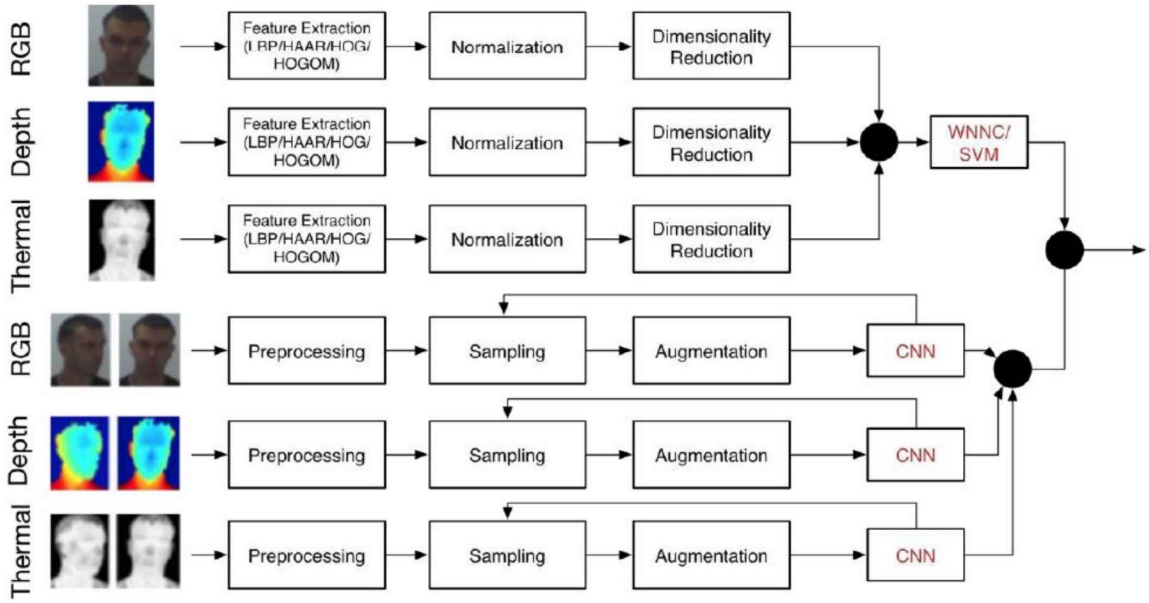


Figure 1: The block diagram of the proposed system. RGB, Depth and Thermal captures of the face are used for training modality specific CNNs for deciding if two samples are from the same person or not. The results are fused with a HOGOM trained WNNC and SVM.

Sim´on MO, Corneanu C, Nasrollahi K, Nikisins O, Escalera S, Sun Y, Li H, Sun Z, Moeslund TB, Greitans M (2016) Improved rgb-dt based face recognition. Iet Biometrics 5(4):297–303

NAN (Yang et al, 2016) ---see VFR

Yang J, Ren P, Chen D, Wen F, Li H, Hua G (2016) Neural aggregation network for video face recognition. arXiv preprint arXiv:160305474

- ✔ Neural Aggregation Network
- ✔ feature embedding module
  - A CNN which maps each face frame into a feature representation
- ✔ neural aggregation module
  - composed of two content based attention blocks which is driven by a memory storing all the features extracted from the face video through the feature embedding module
  - The output of the first attention block adapts the second, whose output is adopted as the aggregated representation of the video faces
  - Due to the attention mechanism, this representation is invariant to the order of the face frames

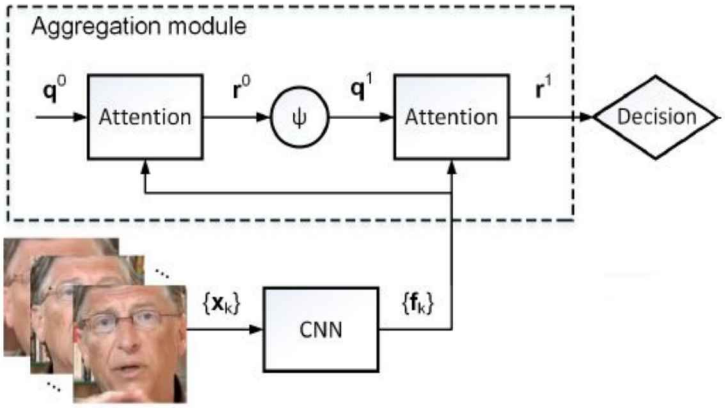


Figure 1. The face recognition framework of our method. All input faces $\{x_k\}$ are processed by a feature embedding module with a CNN, yielding a set of feature representations, $\{f_k\}$. These features are passed to the aggregation module, producing a 128-dimensional representation r^1 for the input video faces. This compact representation can then be used for the decision.

Aggregation Module

- ✔ designed to take benefits from all frames in a video, potentially containing more discriminative information than a single image
- ✔ and handle arbitrary video size in an unified form, producing an order invariant representation

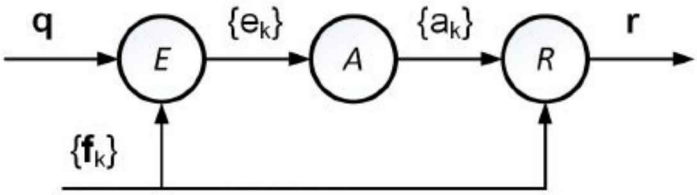


Figure 2. The attention block. It receives a set of feature vectors and filters each of them independently by a kernel q, yielding a set of scalars $\{e_k\}$. There scalars are then passed to a softmax operator, producing a set of weights $\{a_k\}$. Finally, the input feature vectors are fused via Eq. 1.

$$r = \sum_k a_k f_k. \quad (1)$$

ABTA (Dong et al, 2017)

- ✔ Improved from NAN by combining transfer learning
- ✔ Attention-Based Template Adaptation also contains two modules
  - attention based neural network (feature extractor) to integrate the template features of various lengths to a single fixed length feature representation, according to the attention mechanism
  - template adaptation module to transfer the knowledge of a hold-out dataset to the test templates to improve the performance via transfer learning

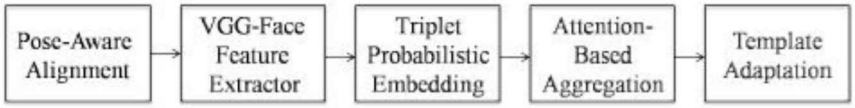


Fig. 2. Overall procedure. Step 1, align the faces of similar pose using their visible landmarks; step 2, extract the features of the aligned faces using VGG-FACE model; step 3, learn the triplet probabilistic embedding using the training data; step 4, attention-based feature aggregation using neural aggregation network; step 5, template adaptation using one-shot-similarity. Step 1 to step 4 constitute the feature extractor module, and step 5 is the transfer module.

Dong B, An Z, Lin J, Deng W (2017) Attention-based template adaptation for face verification. In: Automatic Face & Gesture Recognition, Intl. Conf. on, IEEE, pp 941-946

Ranjan et al (2016)

- ✔ a multi-purpose CNN architecture
- ✔ can simultaneously perform multiple tasks:
  - face identification and verification
  - face detection, landmarks localization, pose estimation, gender recognition, smile detection, and age estimation
- ✔ employs a multi-task learning (MTL) framework to regularize the shared parameters of the network

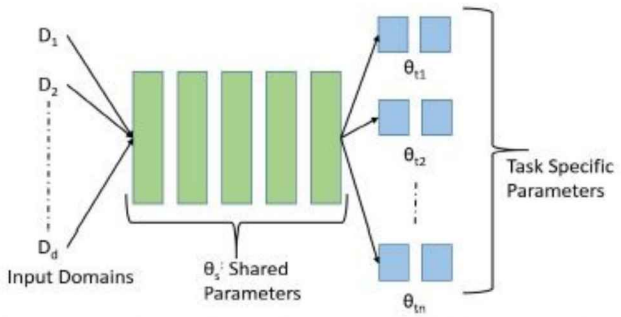


Fig. 2: A general multitask learning framework for deep CNN architecture. The lower layers are shared among all the tasks and input domains.

Ranjan R, Sankaranarayanan S, Castillo CD, Chellappa R (2016) An all-in-one convolutional neural network for face analysis. arXiv preprint arXiv:161100851

Besides the supervised CNN, there are also some weakly-supervised or unsupervised CNN models.

- SL-DCNN (Chen and Deng, 2016)
  - ✔ is a weakly-supervised self-learning DCNN for face recognition

- JFL (Lu et al, 2015)
  - ✔ stacks an unsupervised feature learning method into a DCNN
  - ✔ to learn a hierarchical feature representation
  - ✔ It uses different feature dictionaries to represent the physical characteristics of various face regions
  - ✔ and learns multiple related feature projection matrices for these regions

- Chen B, Deng W (2016) Weakly-supervised deep self-learning for face recognition. In: Multimedia and Expo, Intl. Conf. on, IEEE, pp 1–6
- Lu J, Liong VE, Wang G, Moulin P (2015) Joint feature learning for face recognition. IEEE trans on Information Forensics and Security 10(7):1371–1383

- LBPNet (Xi et al, 2016)
  - ✔ Local Binary Pattern Network
  - ✔ a simplified deep network with handcrafted filters for unsupervised learning
  - ✔ It keeps the same topology of CNN whereas the trainable kernels are replaced by LBP

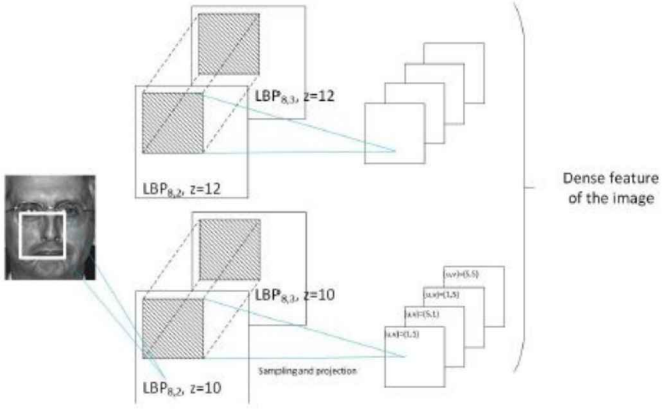


Fig. 1. The deep network part of LBPNet for feature extraction.

Xi M, Chen L, Polajnar D, Tong W (2016) Local binary pattern
network: a deep learning approach for face recognition. In: Image Processing, Intl. Conf. on, IEEE, pp 3224–3228

Some CNN-based methods are proposed in an end-to-end fashion

- Chen et al (2015b)
  - ✔ proposed an automatic end-to-end face verification system with a complete pipeline

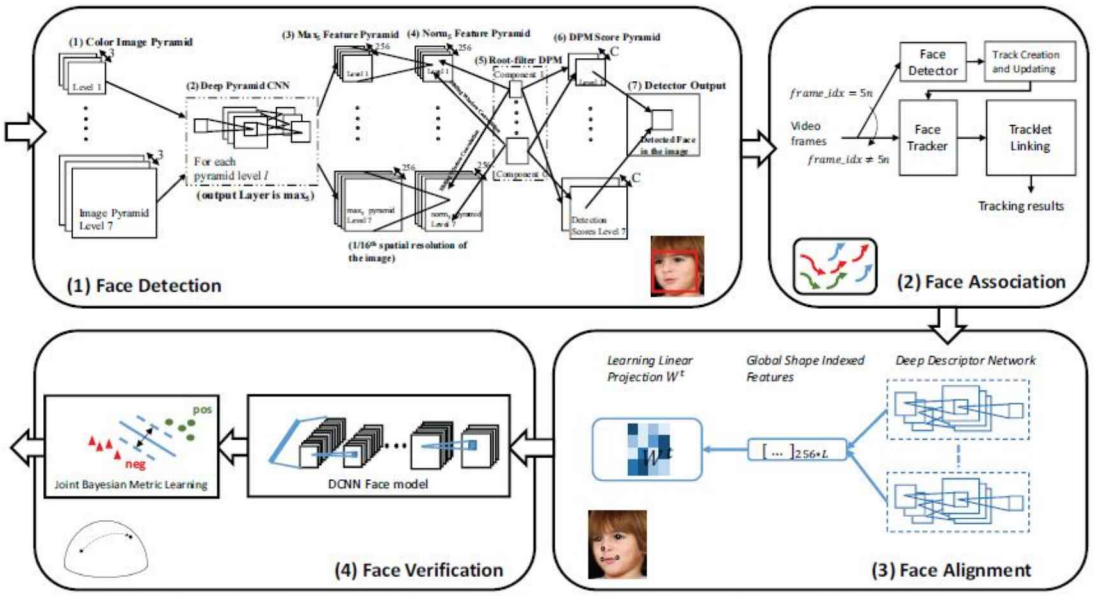


Figure 1. An overview of the proposed end-to-end DCNN-based face verification system.

Chen JC, Ranjan R, Kumar A, Chen CH, Patel VM, Chellappa R (2015b) An end-to-end system for unconstrained face verification with deep convolutional neural networks. In: Proceedings of IEEE Intl. Conf. on Computer Vision Workshops, pp 118–126

- Wu et al (2017a)
  - ✔ Inspired by the spatial transformer
  - ✔ introduced a Recursive Spatial Transformer (ReST) module into CNN
  - ✔ to jointly optimize face alignment and recognition in an end-to-end fashion
  - ✔ The ReST can align faces to the canonical view in a progressive way, which can be considered as an alignment-free face recognition system

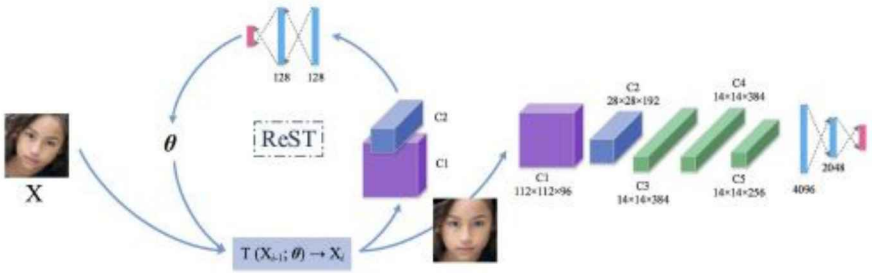


Figure 3. Overview of the proposed ReST integrated in a CNN for alignment-free face recognition.

Wu W, Kan M, Liu X, Yang Y, Shan S, Chen X (2017a) Recursive spatial transformer (rest) for alignment-free face recognition. In: Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition, pp 3772–3780

❖ Other CNN Related Issues

- Phillips (2017)
  - ✔ did a cross-benchmark assessment of VGGFace for FR
  - ✔ on eight National Institute of Standards and Technology (NIST) benchmarks
- Reale et al (2017)
  - ✔ presented a method to remove unnecessary hidden nodes from a deep neural network
  - ✔ by using the group lasso penalty (Meier et al, 2008) to select the appropriate number of hidden nodes for each convolutional and fully connected layer
- Franc and Cech (2017)
  - ✔ discussed the problem of learning CNNs for FR
  - ✔ from weakly annotated images assigned with a pair of attribute and identity labels to a set of faces that are automatically detected from each image
  - ✔ and proposed a heuristic for assigning the annotations to faces
- Bansal et al (2017)
  - ✔ tried to explore some issues that are critical to FR, including:
    - which dataset is better in training CNN, deeper and wider?
    - Is there a need to do face alignment? etc.

- Ferrari et al (2017)
  - ✔ made an evaluation on AlexNet and VGGFace
  - ✔ to explore how the performance of a DCNN is influenced by the characteristics of raw image data
    - such as bounding box size, alignment, positioning and data source
  - ✔ draw some interesting conclusions including that:
    - image normalization operations are less useful for feature extraction in DCNN
    - training and test data are dependent on each other
    - performance is affected by data source combinations (images or video frames)
- Parde et al (2017)
  - ✔ made an analysis on the nature of the face code in the top level of DCNNs
  - ✔ It shows that:
    - DCNN features retain a surprising amount of information about the original input image
    - the tendency to develop a view-dependent code was a characteristic of the identities rather than the features
    - an image distance from the origin of the DCNNs top-level feature space could be used to index the quality of an image

Some deep models were designed to be applicable to real life scenarios

- Jiang and Wang (2017)
  - ✔ an effective face detection and recognition system based on end-to-end deep CNN
  - ✔ composed of two parts
    - the detection network: follows the structure of the faster R-CNN network, which is further composed of two modules: the deep fully convolutional network module for region proposal and the fast R-CNN detector that uses the proposed regions.
    - the recognition network: follows the structure of the deep CNN based FaceNet

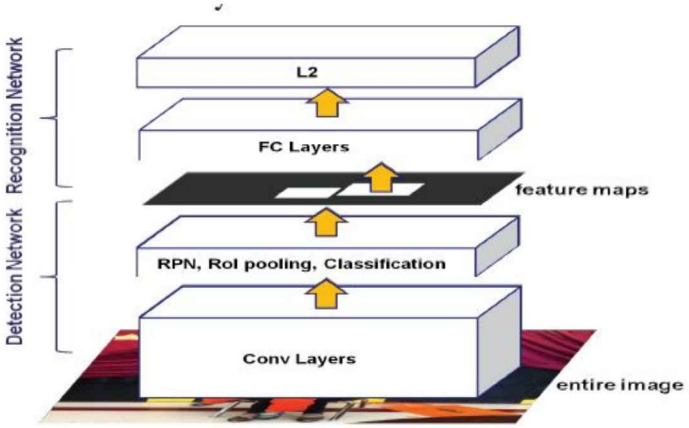


Fig. 1. The unified network structure for end-to-end face detection and recognition.

Jiang W, Wang W (2017) Face detection and recognition for home service robots with end-to-end deep neural networks. In: Acoustics, Speech and Signal Processing, Intl. Conf. on, IEEE, pp 2232–2236

- Chen et al (2017)
  - ✔ designed a DCNN-based complete pipeline for face verification, which can automatically perform automatic face detection, association and alignment in still and video faces

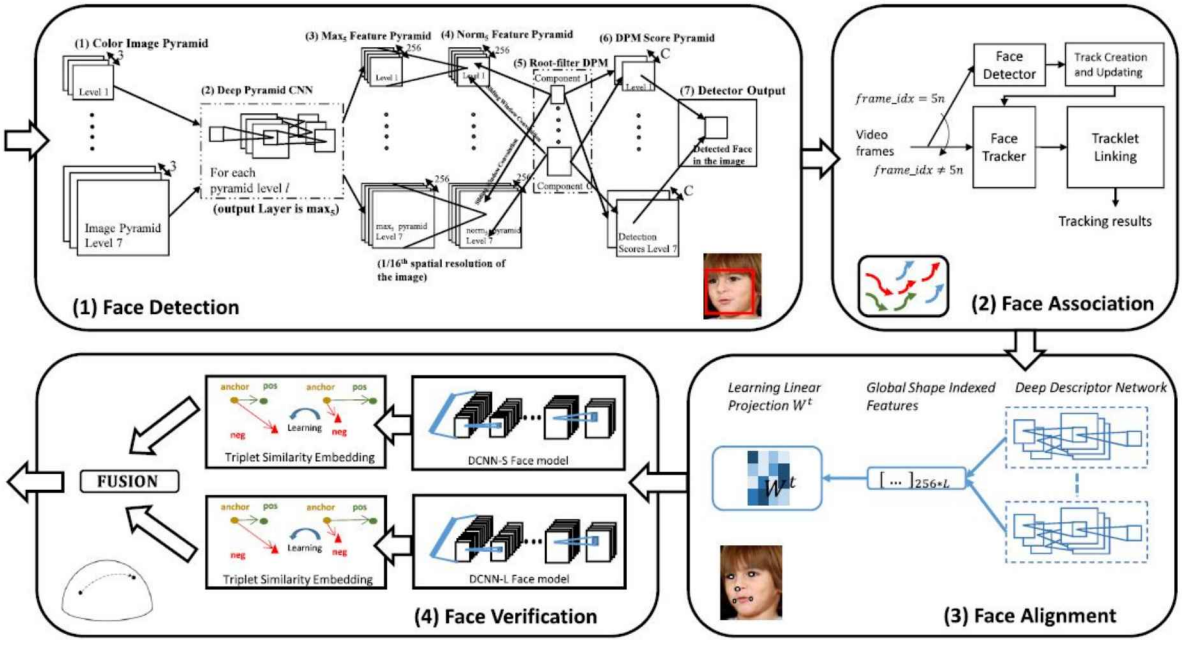


Fig. 1 An overview of the proposed DCNN-based face verification system

- first perform face detection to localize faces in each image and video frame
- Then, associate the detected faces with the common identity across video frames and align the faces into canonical coordinates using the detected landmarks
- Finally, perform face verification to compute the similarity between a pair of images/videos

Chen JC, Ranjan R, Sankaranarayanan S, Kumar A, Chen CH, Patel VM, Castillo CD, Chellappa R (2017) Unconstrained still/videobased face verification with deep convolutional neural networks. Intl Journal of Computer Vision pp 1–20

- VIPLFaceNet (Liu et al, 2017c)
  - ✔ provided an open source deep face recognition SDK based on DCNN for various face recognition applications.

- Dam et al (2015)
  - ✔ developed a light weight CNN that can be deployed on regular commodity computers for real time FR.