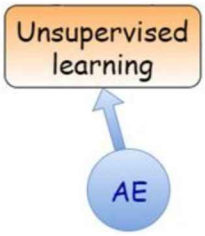
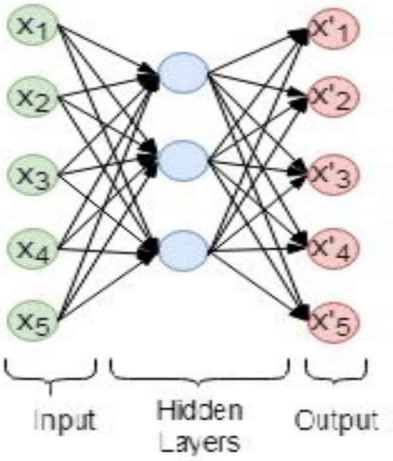
A Survey on Deep Learning based Face Recognition

Na Zhang

Part II: AE, GAN and Other Networks

AE: Autoencoder





- A feedforward, non-recurrent neural network
- Similar to the multilayer perceptron (MLP)
- Contains an input layer, an output layer, one or more hidden layers
- Hidden layers:
  - Reconstruct their own inputs, which forces hidden layers to try to learn good representations of the inputs
  - Instead of predicting the target value Y, given inputs X

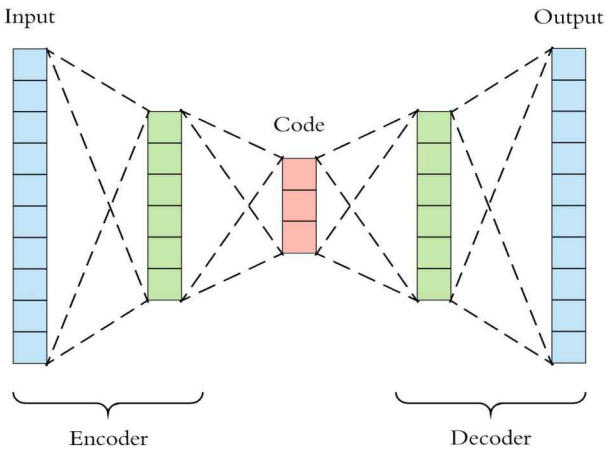


Figure from:
https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

- Encoder
  - maps the input *x* onto *z* which is usually referred to as code, latent variable, or latent representation
- Decoder
  - maps *z* to the reconstruction *x* of the same shape as *x*
- Goal: to minimize reconstruction errors
- Trained layer-by-layer
- Can be used for efficient coding

Variations of AE

- DAE: Denoising Autoencoder
  - Enhances its generalization by training with locally corrupted inputs
  - Does two things:
    - ✔ encode the input
    - ✔ undo the effect of a corruption process
- SAE: Stacked Autoencoder
  - Stacked to form a deep network by feeding the latent representation of an autoencoder as input to the next autoencoder
- CAE: Contractive autoencoder
- VAE: Variational autoencoder

- AE is one of the commonly used building blocks in deep neural networks
- A number of deep methods based on it have been proposed recently

Table 4 Overview of deep methods based on AE and its variants

Algorithm	Description/Remark
CpAEs (Riggan et al, 2015)	Coupled autoencoder for learning a target-to-source image representation for HFR
Shao et al (2015)	A framework integrating multiple deep AEs with bagging strategy to deal with classification with missing modality problem
Liu et al (2016a)	7-layer deep neural network; First 6 layers can be seen as an autoencoder network
DDA (Pathirage et al, 2016)	Deep autoencoder for pose, expression
CAN (Xu et al, 2017a)	Coupled AE networks to handle age-invariant FR and retrieval problem
ADSNT (Huang et al, 2016)	Supervised autoencoder
SPAE (Kan et al, 2014)	Stacked progressive autoencoder; Learn pose-robust features
SFDAE (Pathirage et al, 2015)	Stacked face DAEs; A multiple-encoder single-decoder color fusion model
Gao et al (2015)	Stack the supervised autoencoders (SSAE) to form deep architecture to extract features
Zhu et al (2013)	Encoder: a 3-layer CNN; Decoder: reconstruction layer
RF-SME (Zhang et al, 2013)	Encoder: a single-hidden-layer neural network (S-NN)

CpAEs (Riggan et al, 2015): ---see HFR

- ✔ A coupled AEs for learning a target-to-source image representation
- ✔ A cross-modal transformation is learned by:
  - forcing the hidden units (latent features) of two neural networks to be as similar as possible
  - while simultaneously preserving information from the input.

Riggan BS, Reale C, Nasrabadi NM (2015) Coupled auto-associative neural networks for heterogeneous face recognition. IEEE Access 3:1620–1632

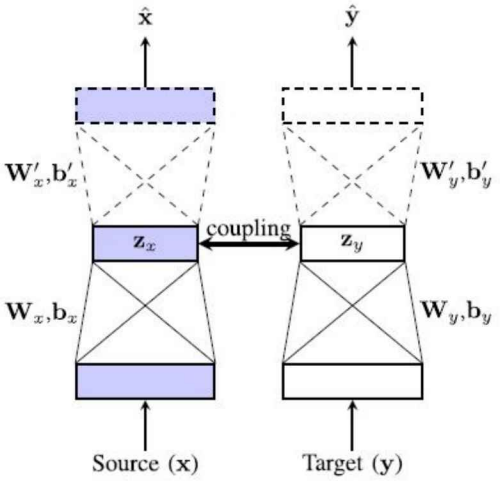


FIGURE 5. A CpAE is a pair of AEs where the hidden units (latent features) are coupled. The latent features, $\mathbf{z}_x$ and $\mathbf{z}_y$, are computed from the source and domain inputs, $\mathbf{x}$ and $\mathbf{y}$, and the encoder parameters: $\mathbf{W}_x$, $\mathbf{b}_x$ and $\mathbf{W}_y$, $\mathbf{b}_y$. Additionally, source and domain reconstructions, $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$, are computed using the latent features and decoder parameters: $\mathbf{W}'_x$, $\mathbf{b}'_x$ and $\mathbf{W}'_y$, $\mathbf{b}'_y$.

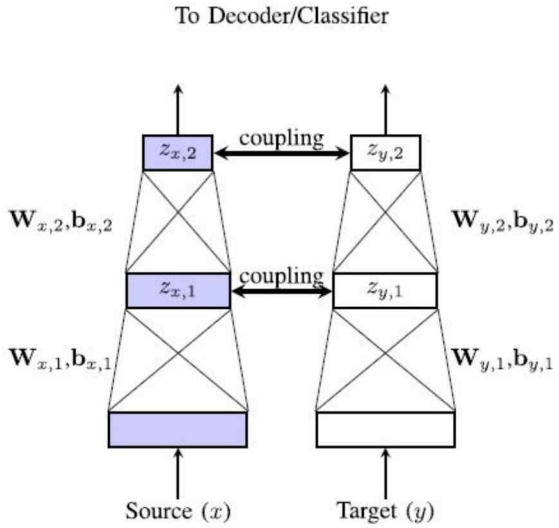


FIGURE 6. A stacked CpAE is a pair of stacked AEs with one (or more) coupled layers of hidden units. As shown, a subsequent CpAE is trained using the hidden units from the previous CpAE. For convenience, we have dropped the decoders.

Shao et al (2015)

Shao M, Ding Z, Fu Y (2015) Sparse low-rank fusion based deep features for missing modality face recognition. In: Automatic Face and Gesture Recognition, Intl. Conf. and Workshops on, IEEE, vol 1, pp 1–6

- ✔ integrated multiple deep AEs
- ✔ each AE generates input by randomly sampling data from another modality and the auxiliary database
- ✔ and enforces the output to lie in a common feature space through Robust PCA.
- ✔ Finally, a sparse, low-rank feature fusion approach is proposed in the test phase to integrate multiple features learned from different AEs, followed by a decision voting

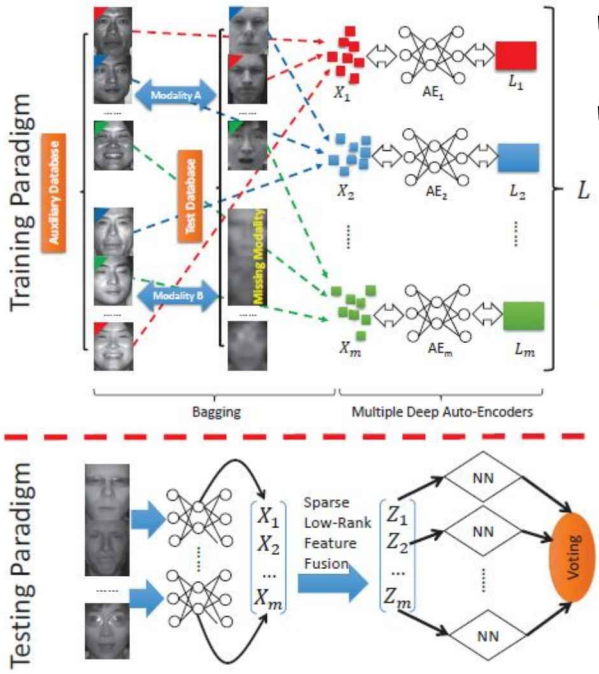


Fig. 1: Framework of the proposed method, which contains two modalities: "Modality-A" as VIS, "Modality-B" as NIR. It uses two databases: an auxiliary database with complete multi-modal data, a test/target database with missing modality. We first use the bagging strategy to sample data from both auxiliary and test databases, where a color (on the up-left corner of each face) represents a sampling. Then these sampled data can train m autoencoders $AE_{1\sim m}$ and yield m decisions which will be fused by a voting scheme. Note X_i and L_i represent a sampled dataset, and its low-rank recovery, respectively in the training phase. X_i and Z_i are deep features, and the new representation of X_i after sparse low-rank feature fusion in the test phase, respectively.

- Liu et al (2016a) ---see facial expression
  - ✔ A fusion based face recognition method
  - ✔ using AE to reduce the dimension of fusion features
  - ✔ Use softmax regression to get identification decision

Liu J, Fang C,Wu C (2016a) A fusion face recognition approach based on 7-layer deep learning neural network. Journal of Electrical and Computer Engineering 2016

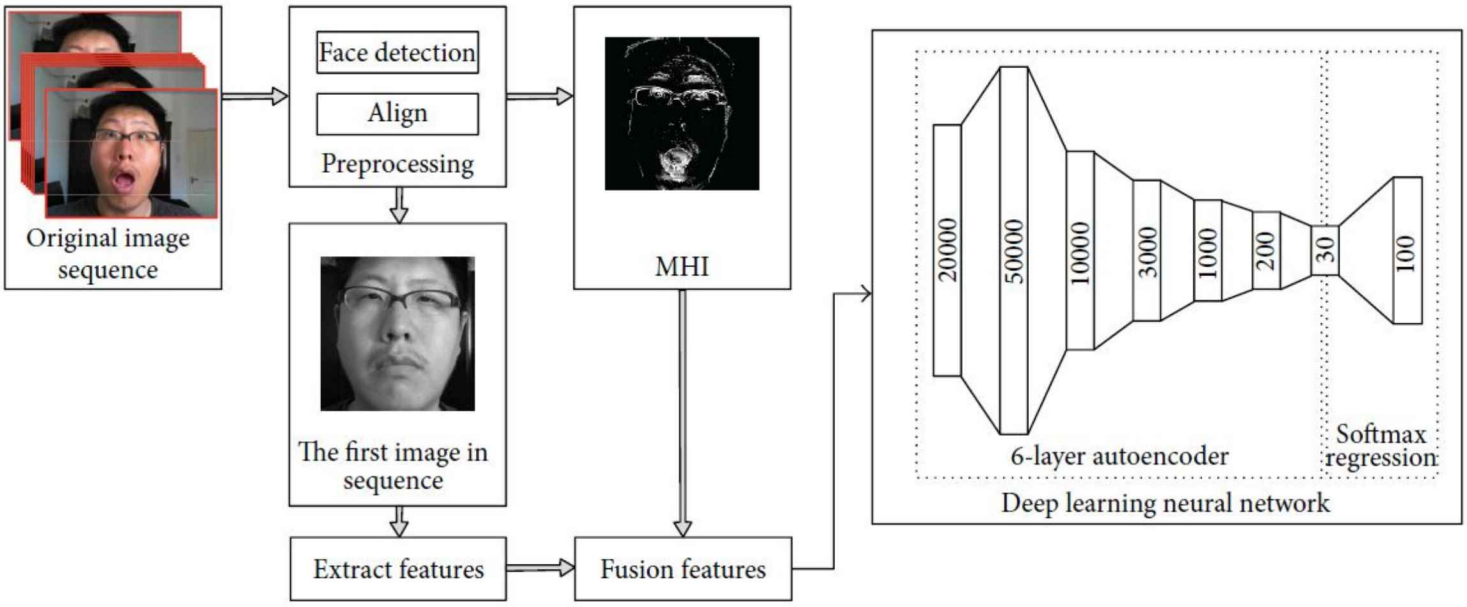


FIGURE 3: Structure of our network.

- DDA (Pathirage et al, 2016) ---see pose&expression
  - ✔ Deep Discriminant Analysis Nets
  - ✔ can learn dynamic data adaptive features used for various problems such as face pose and expressions
  - ✔ consists of 3 interconnected learning processes:
    - the progressive non-linear dimension reduction process:
      L1, L2; yield a low dimensional feature whose effective dimension is half the dimension of the original RGB features
    - de-noising process
      L3; based on a strong supervisory signal which is the neutral frontal face
    - Discrimination process
      L5; based on a single representative face image thus ensures the features observed in the reconstruction layer are highly discriminative

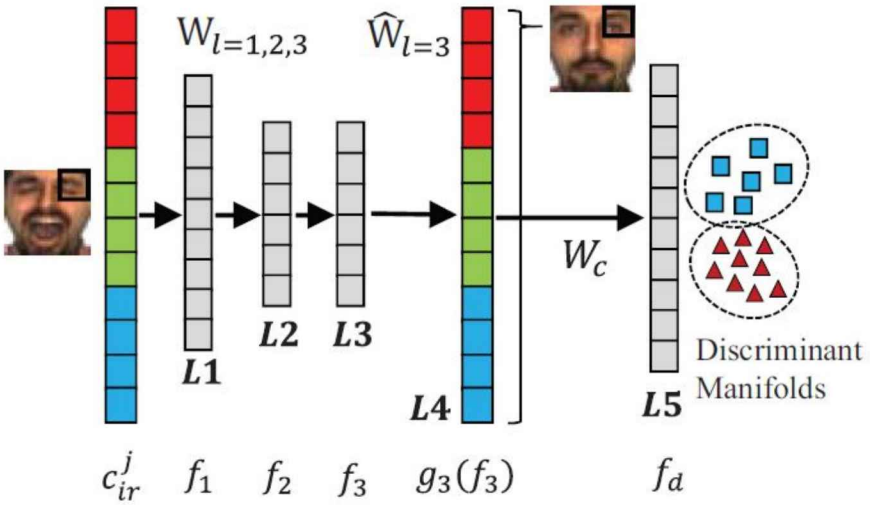


Fig. 1. DDA Net where $c_{ir}^{j} \in \mathbb{R}^{(36*3)}$, $f_1 \in \mathbb{R}^{75}$, $f_2 \in \mathbb{R}^{50}$ denote the combined patch feature and the low dimensional noisy feature learned at Layer 1 (L1) and Layer 2 (L2) respectively while $f_3 \in \mathbb{R}^{50}$ denotes the noise-less feature learned at Layer 3 (de-noising layer) in the observed low dimensional space. $g_3(.)$ represents the decoder function. Hence the discriminant layer where $f_d \in \mathbb{R}^{class\ count-1}$ is shown as the right most layer.

Pathirage CSN, Li L, Liu W (2016) Discriminant auto encoders for face recognition with expression and pose variations. In: Pattern Recognition, Intl.

- ✔ In DDA, each shallow AE is trained to achieve simple but tractable goals required to address the global non-linear objective as a whole
- ✔ The framework follows a patch based approach to further refine the global non-linear objective into simpler tasks
- ✔ choose non-overlapping patches of the face image of size and stride 6x6 respectively
- ✔ it limits the number of parameters of the model that need to be learnt while training each DDA Net in a parallel environment

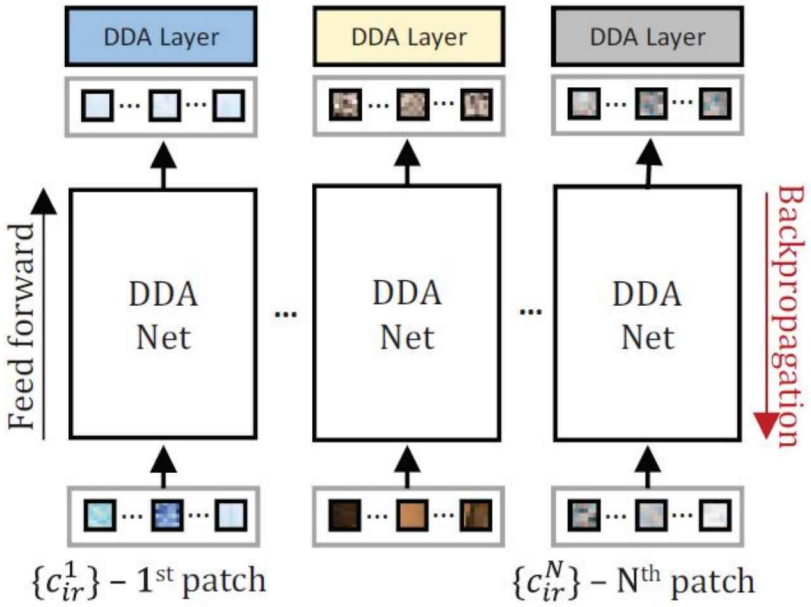


Fig. 2. Patch based DDA Framework that converts each patch of a face image to its corresponding frontal face patch followed by the non-linear discriminant analysis process (DDA layer).

CAN (Xu et al, 2017a)

- ✔ Coupled Autoencoder Networks
- ✔ used AE to handle the cross-age face recognition and retrieval problem

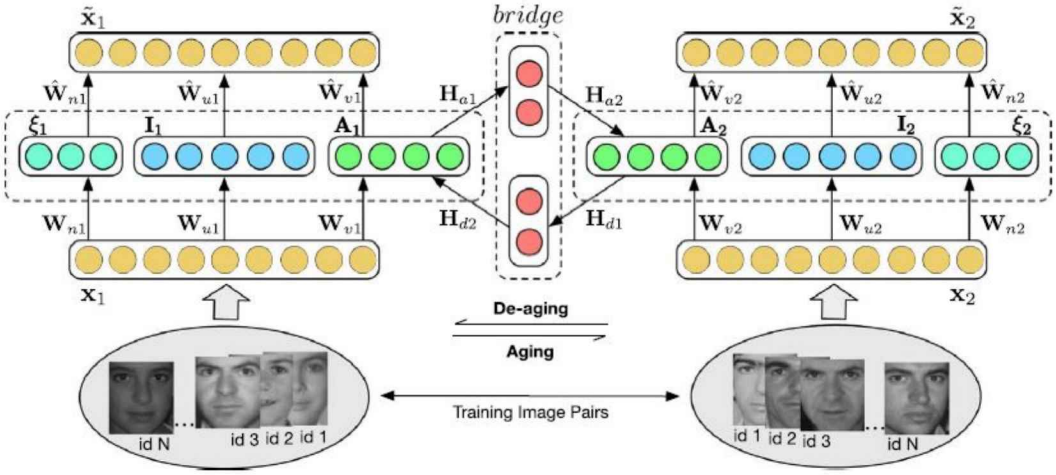


Fig. 2. The overview of CAN. CAN is composed of two identical auto-encoders and a *bridge* network. Given a pair of input images (**x1**, **x2**) of one person, first we leverage auto-encoders to reconstruct inputs to project them into a high-dimensional feature space in hidden layers. Second, we add constraints in the above feature space to decompose it into three components where ($\mathbf{I}_1$, $\mathbf{I}_2$) as identity features can be used as age-invariant representations for recognition and retrieval. Note here different id can refer to the same person.

Xu C, Liu Q, Ye M (2017a) Age invariant face recognition and retrieval by coupled auto-encoder networks. Neurocomputing

Besides, some AE based methods are designed in a supervised manner

- ADSNT (Huang et al, 2016)
  - ✔ an Adaptive Deep Supervised Network Template with a supervised AE
  - ✔ trained to extract characteristic features from corrupted/clean facial images
  - ✔ and reconstruct the corresponding similar facial images.

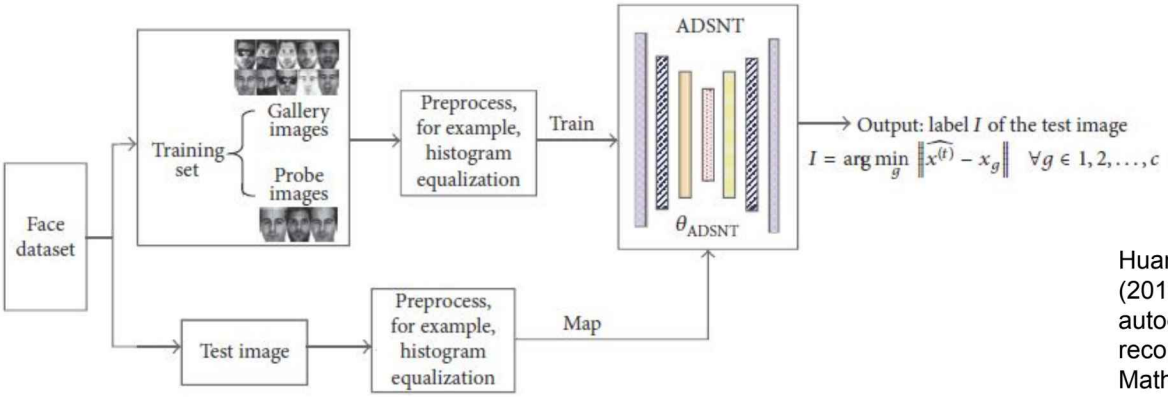
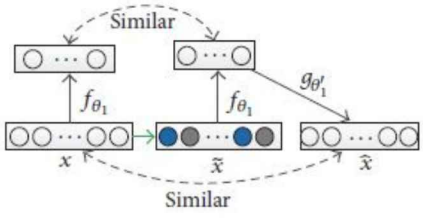
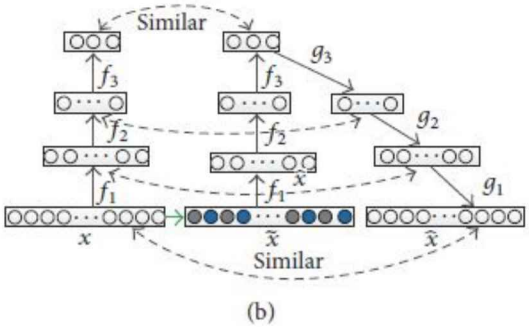


FIGURE 2: Flowchart of the proposed ADSNT image reconstruction for face recognition.

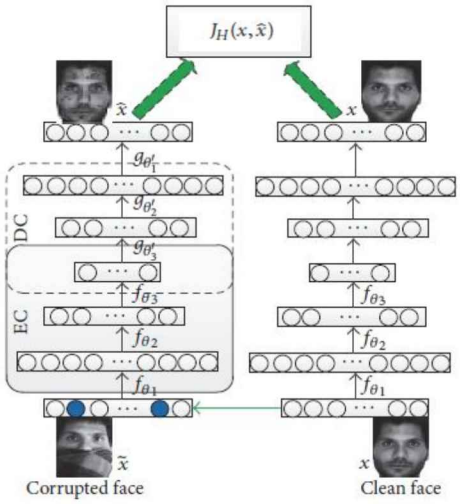
Huang R, Liu C, Li G, Zhou J (2016) Adaptive deep supervised autoencoder based image reconstruction for face recognition. Mathematical Problems in Engineering 2016



(a) Supervised autoencoder (SAE) which is comprised of clean/"corrupted" datum, one hidden layer, and one reconstruction layer by using the "corrupted datum";



(b) stacked supervised autoencoder (SSAE);



(c) architecture of the ADSNT

- ✔ A deep supervised autoencoder (DSAE)
- ✔ Consists of two parts:
  - an encoder (EC)
  - a decoder (DC)
- ✔ Each of them has three hidden layers
- ✔ share the third layer, the central hidden layer
- ✔ The features learned from the hidden layer and the reconstructed clean face are obtained by using the "corrupted" data

Some variations of the AE are also adopted in FR

- SPAE (Kan et al, 2014)
  - ✔ proposed a stacked progressive autoencoder to learn pose-robust features
  - ✔ by modeling a complex non-linear transform from non-frontal face images to frontal ones in a progressive way
  - ✔ SPAE contains multiple progressive AEs, and each maps face at large poses to a virtual view at smaller pose angles
  - ✔ The output contains very small pose variations

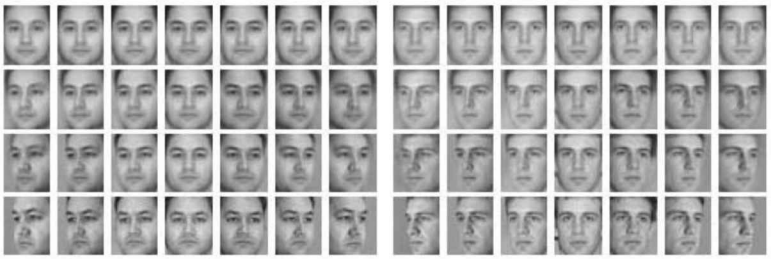


Figure 4. The output from each decoder in the SPAE network for the input images in the bottom row. (a) output of exemplar training images from MultiPIE. (b) output of exemplar testing images from MultiPIE.

Kan M, Shan S, Chang H, Chen X (2014) Stacked progressive autoencoders (spae) for face recognition across poses. In: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, pp 1883–1890

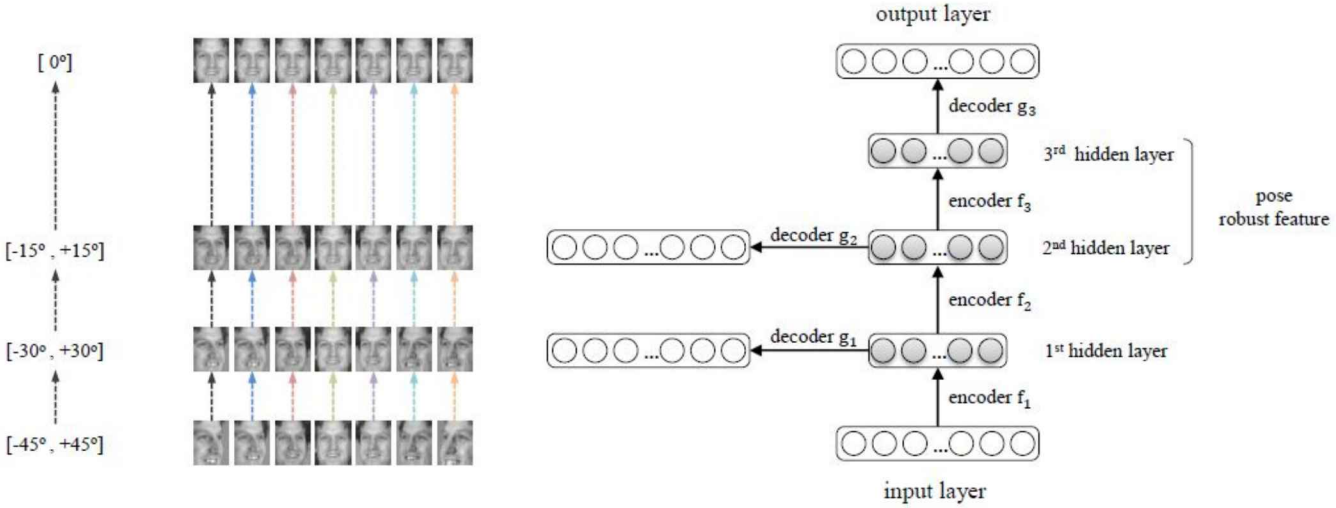


Figure 1. The schema of the proposed Stacked Progressive Auto-Encoders (SPAE) network for pose-robust face recognition. We illustrate an exemplar architecture of the stacked network with $L = 3$ hidden layers, which can deal with poses in yaw rotation within [-45°, +45°]. In training stage of our SPAE, each progressive auto-encoder aims at converting the face images at large poses to a virtual view at a smaller pose (*i.e.*, closer to frontal), and meanwhile keeping the face images with smaller poses unchanged. For instance, for the first progressive AE demonstrated in this figure, only images with yaw rotation larger than 30° are converted to 30°, while other face images with yaw rotation smaller than 30° are mapped to themselves. Such a progressive mode endows each progressive AE a limited goal matching its capacity. In the testing stage, given an image, it is fed into the SPAE network, and the outputs of the topmost hidden layers with very small pose variations are used as the pose-robust features for face recognition.

SFDAE (Pathirage et al, 2015)

- ✔ Inspired by SPAE
- ✔ stacked face denoising autoencoders
- ✔ proposed for expression-robust feature acquisition
- ✔ exploits contributions of different color components in different local face regions by recovering neutral expression from various ones
- ✔ and denoises the face with dynamic expressions in a progressive way

--- see expression

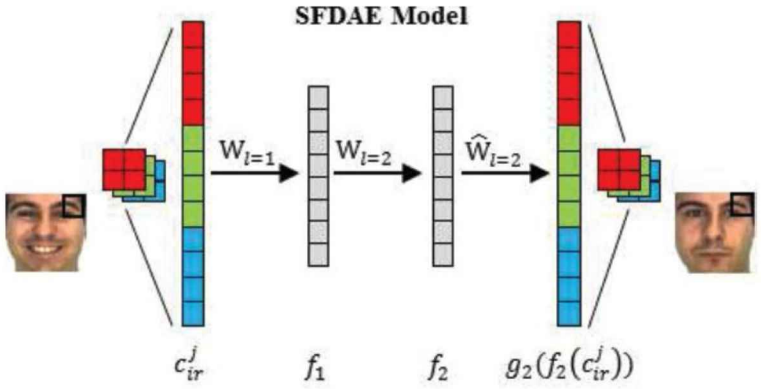


Fig. 4. The proposed SFDAE model where $f_1 \in \mathbb{R}^{50}$ denotes low dimensional noisy feature learnt at layer 1, while $f_2 \in \mathbb{R}^{50}$ denotes the noiseless feature learnt at layer 2 in the observed low dimensional space. We halves the image space by 50% to constraint the model to learn an effective low dimensional feature.

Pathirage CSN, Li L, Liu W, Zhang M (2015) Stacked face de-noising auto encoders for expression-robust face recognition. In: Digital Image Computing: Techniques and

Gao et al (2015)

Gao S, Zhang Y, Jia K, Lu J, Zhang Y (2015) Single sample face recognition via learning deep supervised autoencoders. trans on Information Forensics and Security 10(10):2108–2118

- ✔ Motivated by Denoising AE
- ✔ a supervised autoencoder to learn a robust image representation for the single training sample per person (SSPP) problem
- ✔ It enforces faces with variations mapped to the canonical face
- ✔ and enforces features of the same person to be similar
- ✔ and then it stacks the supervised autoencoders to form a deep architecture to extract features

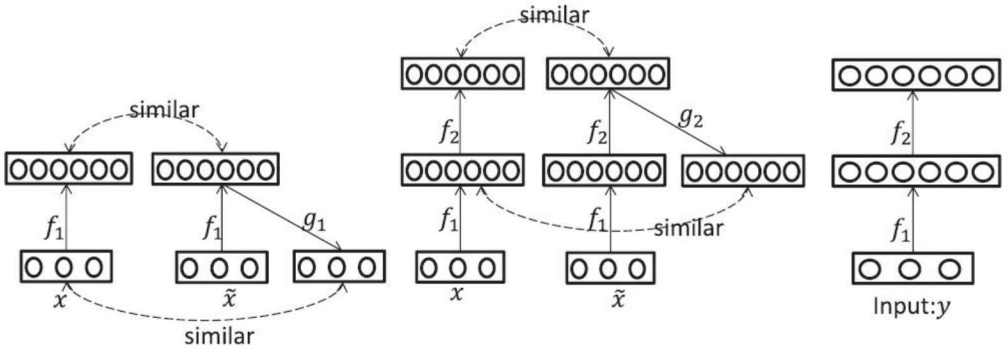
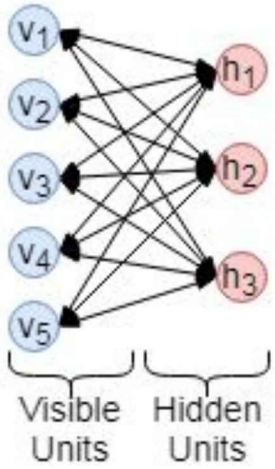


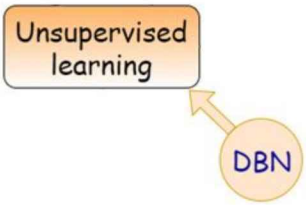
Fig. 2. Architecture of Staked Supervised Auto-Encoders. The left figure: The basic supervised auto-encoder, which is comprised of the clean/"corrupted" faces, there features (hidden layer), as well as the reconstructed clean face by using the "corrupted face". The middle figure: The output of previous hidden layer is used as the input to train the next supervised auto-encoder. We repeat such training several times until the desired number of hidden layers is reached. In this paper, only two hidden layers are used. The right figure: Once the network is trained, given any input face, the output of the last hidden layer is used as the feature for image representation.

RBM: Restricted Boltzmann Machine

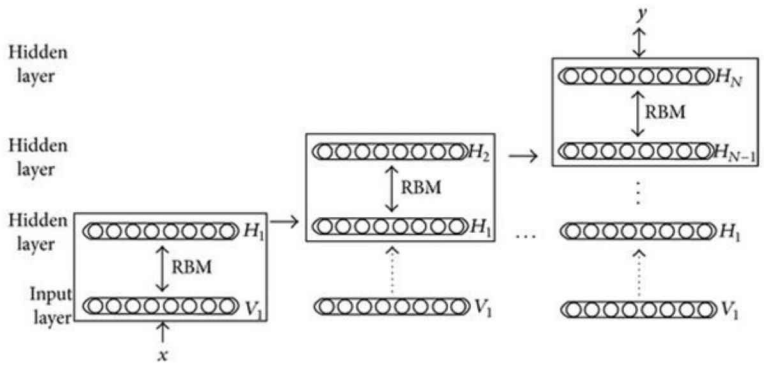
- Boltzmann Machine (BM) is a particular form of log linear Markov Random Field (MRF)
- RBM is a variant of BM with the restriction that:
  - Its neurons must form a bipartite graph
  - A pair of nodes from each of the two groups of units (visible, hidden units) ma have a symmetric connection between them
  - and there are no connections between nodes within a group
- RBM is a shallow, two-layer neural network



DBN: Deep Belief Network

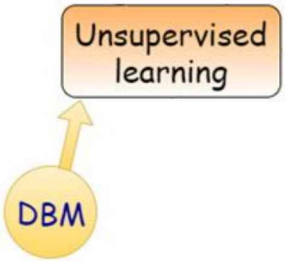


- Can be formed by stacking RBMs
- The learning procedure can be divided into two stages:
  - generative learning to abstract information layer by layer with unlabeled samples, and then
  - discriminative learning to fine-tune the whole deep network with labeled samples to the ultimate learning target

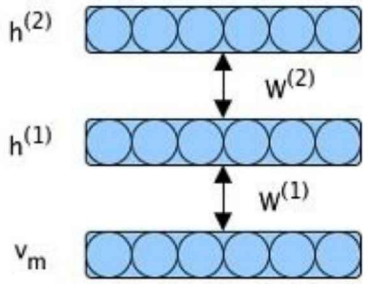


Wang H, Cai Y, Chen L (2014a) A vehicle detection algorithm based on deep belief network. The scientific world journal 2014

DBM: Deep Boltzmann Machines



- Gained significant attentions in learning of:
  - higher level and more complex representation of data
  - distribution of observations
- Nonlinear latent variables in DBM are organized in multiple connected layers in a way that:
  - variables in one layer can simultaneously contribute to the probabilities or states of variables in the next layer
- Each layer learns a different factor to represent the variations in the given data.



21

- In FR, some methods were introduced using DBN, DBM and/or RBM

Table 5 Overview of deep methods based on DBN, DBM, RBM

Algorithm	Description/Remark
Chen et al (2013b)	A feature learning method by stacking the RBM networks
Yi et al (2015)	A local to global learning framework based on RBM for heterogeneous face recognition
CDBN (Huang et al, 2012b)	Convolutional deep belief networks to learn features in high-resolution face images
Jhuang et al (2016)	Use DBN to train identification model using features with depth information of 3D data
Wu et al (2013)	Use DBM to track facial feature under varying expressions and poses
DAMs (Duong et al, 2015)	2 DBMs capture variations of facial shapes and appearances respectively

- Jhuang et al (2016):
  - ✔ built a DBN based network to learn features
  - ✔ three-dimensional face verification approach that includes three phases:
    - point cloud library is applied to estimate features
    - adopt deep belief networks to train the identification model using extracted features
    - face verification is accomplished
  - ✔ DBN:
    - one visible layer comprising n neurons
    - two hidden layers having n/2 neurons
    - one regression layer comprising two output labels

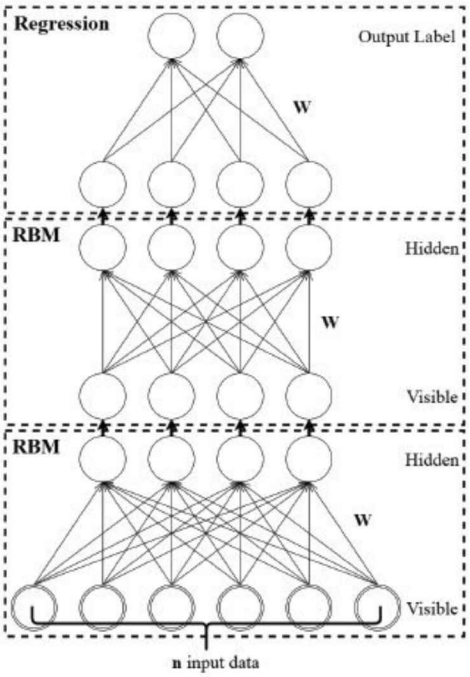
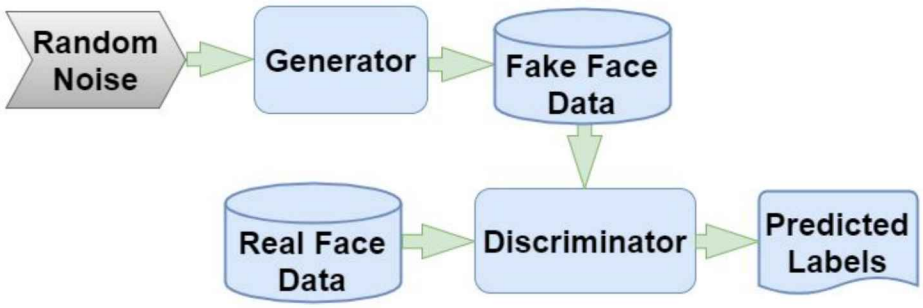


Fig. 8. Deep belief networks architecture.

Jhuang DH, Lin DT, Tsai CH (2016) Face verification with three-dimensional point cloud by using deep belief networks. In: Pattern Recognition, Intl. Conf. on, IEEE, pp 1430–1435

GAN: Generative Adversarial Network

- Gained much attention in recent two years
- Adopted to handle more complicated recognition tasks
- General idea:
  - to build two competing neural network models
- Two independent networks, which work separately and act as adversaries



- generative model (generator):
  - takes noise as input and generates samples
- discriminative model (discriminator):
  - receives samples from both generator and training data,
  - has to be able to distinguish between the two sources

- The two models play a continuous game
  - the generator
    - learns to produce more and more realistic samples
  - the discriminator
    - learns to get better and better at distinguishing the generated data from real data
- The two models are trained simultaneously
- The goal is that:
  - the competition will drive the generated samples to be indistinguishable from real data

- GAN can be viewed as an architecture able to achieve far better performance compared to the traditional networks

Table 6 Overview of deep methods based on GAN

Algorithm	Description/Remark
Age-cGAN (Antipov et al, 2017b)	A aging/rejuvenation method to synthesize more plausible and realistic faces
AgecGAN+LMA (Antipov et al, 2017a)	A generative aging/rejuvenation method
GAN-VFS (Zhang et al, 2017a)	Visible Face Synthesis method to synthesize photo realistic visible face images
DR-GAN (Tran et al, 2017)	GAN based framework for pose-invariant face recognition and face synthesis
DAN (Rao et al, 2017a)	A discriminative aggregation network for video face recognition
BLAN (Li et al, 2017)	A bi-level adversarial network for makeup-invariant face verification

- Zhang et al (2017a): a GAN based Visible Face Synthesis (GAN-VFS) method
  - ✔ Synthesize visible faces from their corresponding polarimetric thermal images
  - ✔ The whole network contains an encoder-decoder structure
  - ✔ Learned visible features: outputs of the encoder; input for the decoder
  - ✔ Guidance sub-network: to guarantee the reconstructability of the encoded features and to make sure that the leaned features

Zhang H, Patel VM, Riggan BS, Hu S (2017a) Generative adversarial network-based synthesis of visible faces from polarimetric thermal faces. arXiv preprint arXiv:170802681

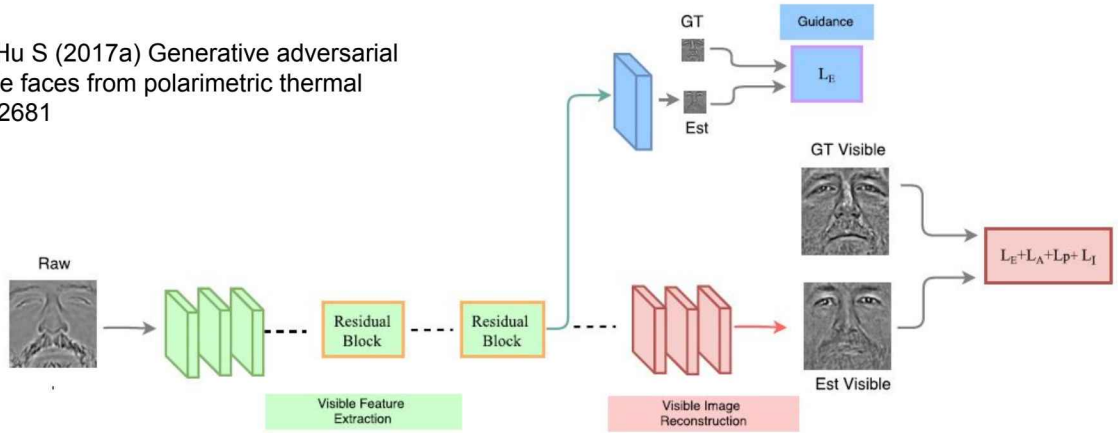


Figure 2: Overview of the proposed GAN-VFS method. It contains three modules. (a) Visible feature extraction module, (b) Guidance sub-network and (c) Visible image reconstruction module. Firstly, the visible feature is extracted from the raw polarimetric image. Then, to make sure that the learned feature can better reconstruct the visible image, a guidance sub-network is involved into the optimization. Finally, the guided feature is used to reconstruct the photo realistic visible image using the combination of different losses.

DR-GAN (Tran et al, 2017)

✔ Disentangled Representation Learning GAN () for pose-invariant face recognition and face synthesis

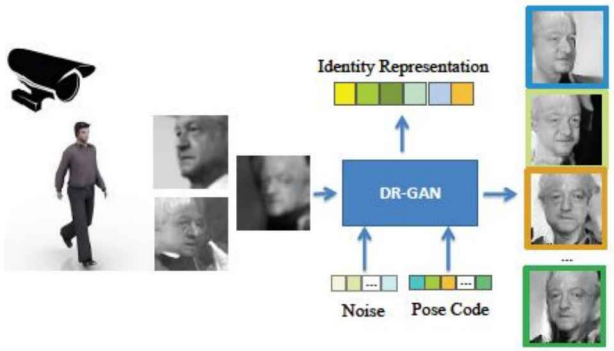


Figure 1: With one or multiple face images as the input, DR-GAN can produce an identity representation that is both discriminative and generative, i.e., the representation demonstrates superior PIFR performance, and can synthesize identity-preserving faces at target poses specified by the pose code.

Tran L, Yin X, Liu X (2017) Disentangled representation learning gan for pose-invariant face recognition. In: CVPR, vol 4, p 7

✔ Generator: serves as a face rotator

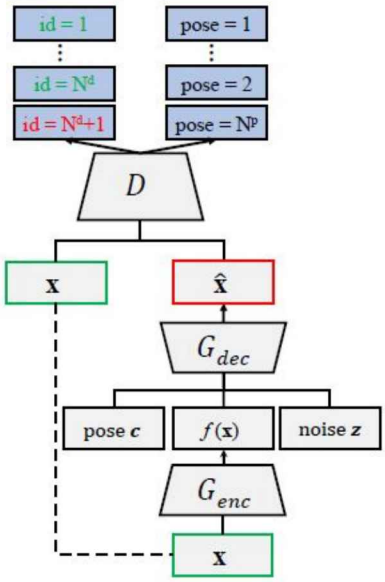
- an encoder-decoder structured generator
- input X to the encoder G_{enc} is a face image of any pose
- output $\hat{X}$ of the decoder G_{dec} is a synthetic face at a target pose
- the learnt representation $f(x)$ bridges G_{enc} and G_{dec}

✔ Discriminator: do pose classification

- distinguish real X vs. synthetic images $\hat{X}$
- predict the identity and pose of a face
- strives for the rotated face to have the same identity as the input real face, which has two effects on G:

1) The rotated face looks more like the input subject in terms of identity

2) The learnt representation is more inclusive or generative for synthesizing an identity-preserving face



DR-GAN

DAN (Rao et al, 2017a)

- ✔ a discriminative aggregation network
- ✔ for video FR
- ✔ combine the idea of adversarial learning with metric learning
- ✔ aggregate the useful information of an input video into one or few more discriminative images in the feature space

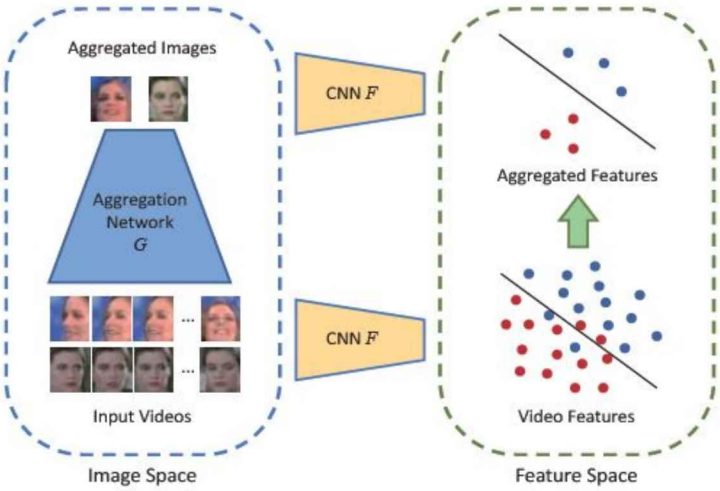
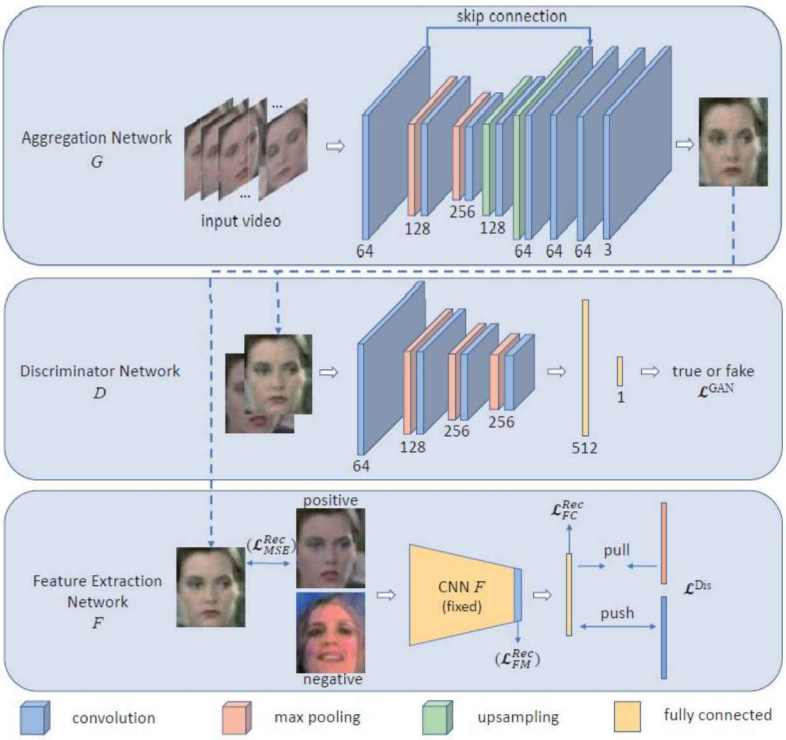


Figure 1. The basic idea of our proposed frames aggregation method. For each video clip, we integrate the information of videos to produce few synthesized images with discriminative aggregation network (DAN). The supervision signal of our proposed framework makes the synthesized images more discriminative than original frames in the feature space. Besides, we only need to pass the few aggregated images into feature extraction network and thus greatly speed up the overall system.

Rao Y, Lin J, Lu J, Zhou J (2017a) Learning discriminative aggregation network for video-based face recognition. In: Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition, pp 3781–3790



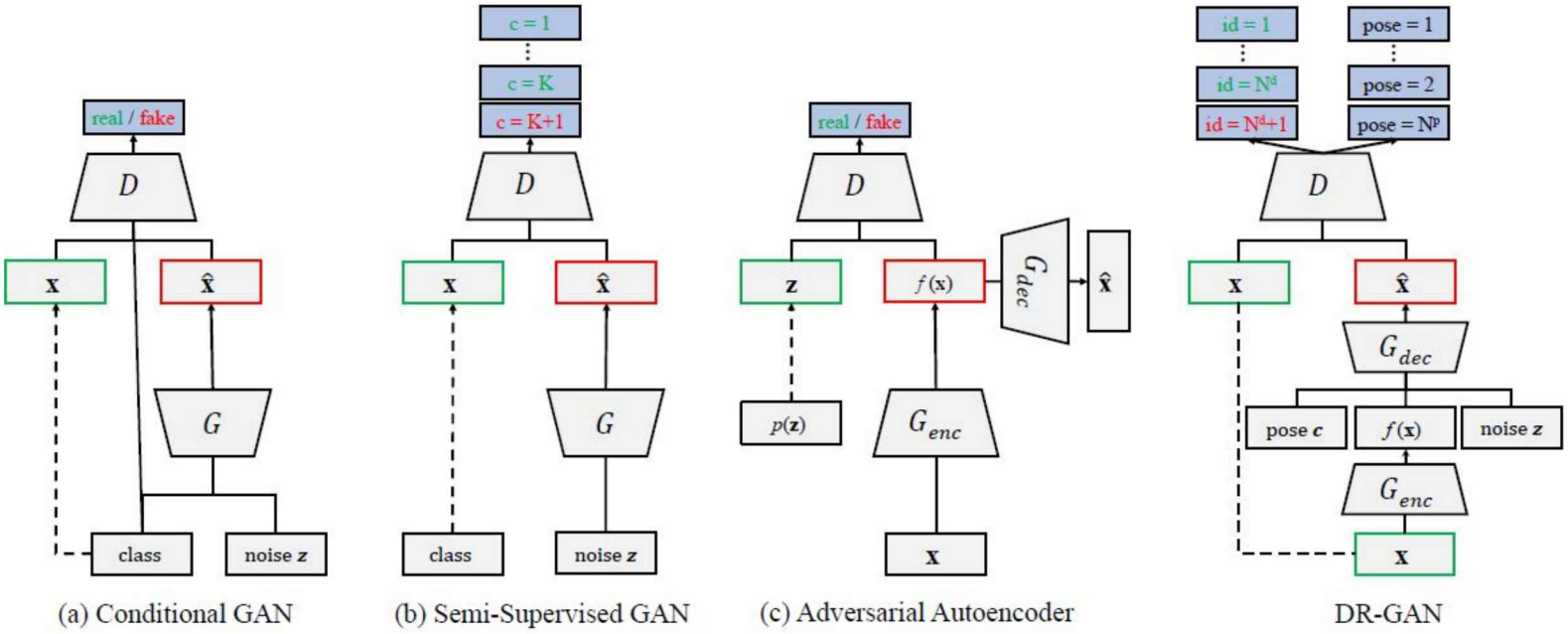
- 3 sub-networks
  - ✔ aggregation (generator) network G, D and F
- can aggregate video clip into single image while at the same time gain more discriminative power
- Loss function

$$\mathcal{L} = \lambda\mathcal{L}^{Dis} + \eta\mathcal{L}^{Rec} + 0.01\mathcal{L}^{GAN}$$

$\mathcal{L}^{Dis}$ is the discriminative loss
$\mathcal{L}^{Rec}$ is the reconstruction loss
$\mathcal{L}^{GAN}$ is the adversarial loss

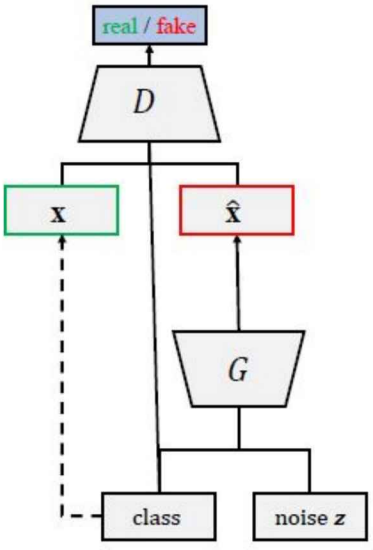
Figure 2. Detailed architecture of our proposed framework. The numbers are either the feature map channel for convolutional blocks or feature dimension for fully connected layers. The output of aggregation network is then fed into discriminative network for adversarial learning, and the feature extraction network to increase discrimination. Different losses are applied at different places as illustrated in the figure.

Some GAN variants



(a) Conditional GAN (b) Semi-Supervised GAN (c) Adversarial Autoencoder DR-GAN

- # Conditional GAN
  - extends the GAN by feeding the labels to both G and D to generate images conditioned on the label, which can be the class label, modality information, or even partial data for inpainting
  - In conditional GAN, D is trained to classify a real image with mismatched conditions to a fake class

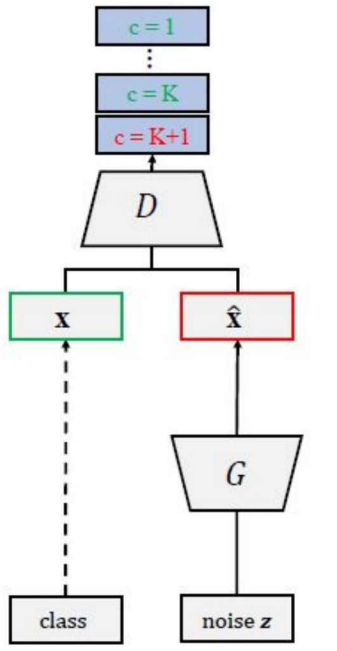


(a) Conditional GAN

Mirza M, Osindero S (2014) Conditional generative adversarial nets. arXiv preprint

Semi-Supervised GAN

- Generalizes GAN to learn a discriminative classifier
- D is trained to not only distinguish between real and fake images, but also classify real images into K classes
- D outputs a (K+1)-dim vector with the last dimension for the real/fake decision
- The trained D is used for image classification

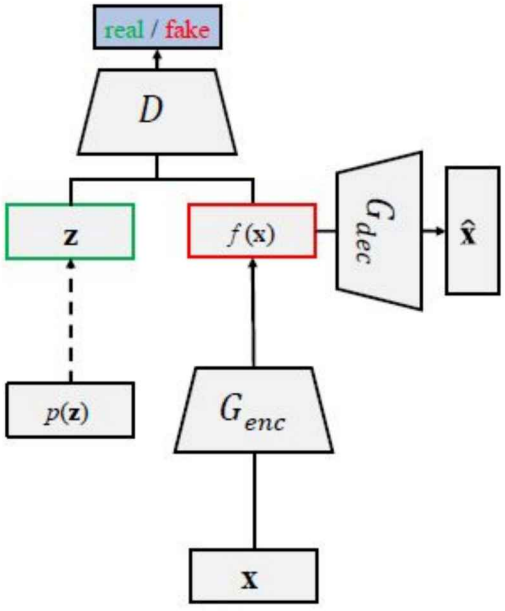


(b) Semi-Supervised GAN

Odena A (2016) Semi-supervised learning with generative adversarial networks. arXiv preprint

Adversarial Autoencoder (AAE)

- G is the encoder of an autoencoder
- has two objectives in order to turn an autoencoder into a generative model:
  - ✔ the autoencoder reconstructs the input image
  - ✔ the latent vector generated by the encoder matches an arbitrary prior distribution by training D

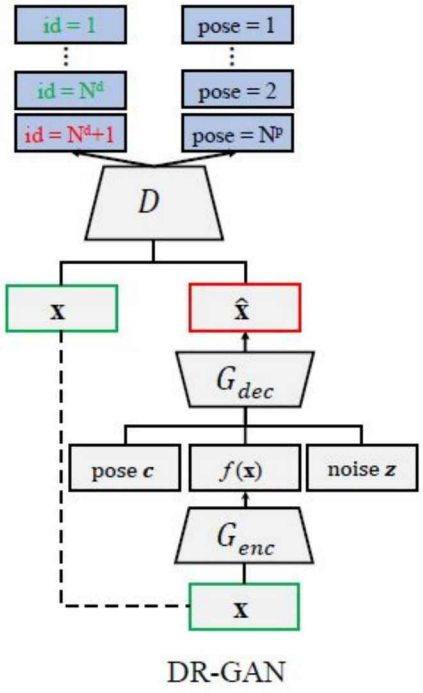


(c) Adversarial Autoencoder

Makhzani A, Shlens J, Jaitly N, Goodfellow I, Frey B (2015) Adversarial autoencoders. arXiv preprint

DR-GAN

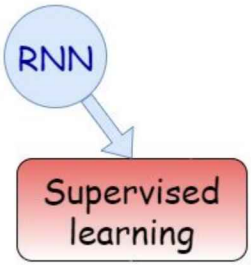
- D classifies a real image to the corresponding class based on the label
- DR-GAN differs to AAE in two aspects:
  - ✔ First, the autoencoder in AAE is trained to learn a latent representation similar to an imposed prior distribution, while DR-GAN encoder-decoder learns discriminative identity representations
  - ✔ Second, D in AAE is trained to distinguish real/fake distributions while D in DR-GAN is trained to classify real/fake images, the identity and pose of the images



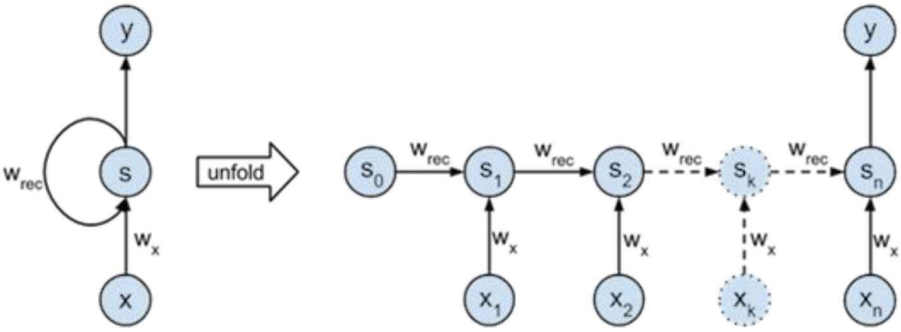
DR-GAN

Tran L, Yin X, Liu X (2017) Disentangled representation learning gan for pose-invariant face recognition. In:

RNN: Recurrent Neural Network



- A special network to deal with sequences of inputs
- The connections between units form a directed cycle, allowing it to exhibit dynamic temporal behavior



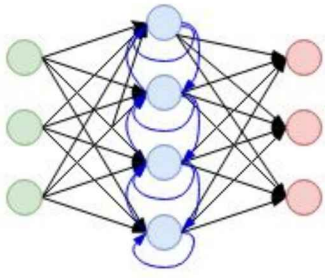
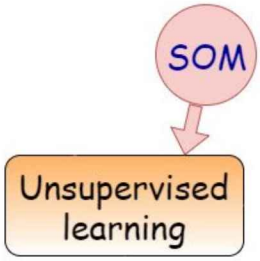


Figure from: http://peterroelants.github.io/posts/rnn_implementation_part01/

- Can be used for mapping inputs to outputs of varying types, lengths and is fairly general in its applications
- Has been applied to some tasks, e.g.
  - Unsegmented, connected handwriting recognition
  - speech recognition
  - …, etc.
- but not much to face recognition

SOM: Self-Organizing Map



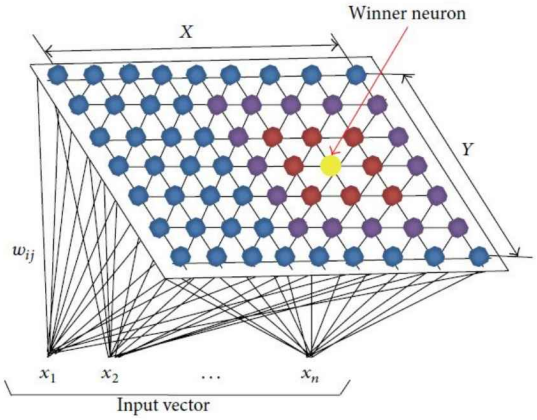


FIGURE 2: SOM input and output layers.

- Provides a data visualization technique which helps to understand high dimensional data by reducing the dimensions of data to a map
- Based on competitive learning, in which
  - the output neurons compete amongst themselves to be activated, with the result that only one is activated at one time

Figure from: Yorek N, Ugulu I, Aydin H (2016) Using self-organizing neural network map combined with ward's clustering algorithm for visualization of students' cognitive structural models about aliveness concept. Computational intelligence and neuroscience 2016:6

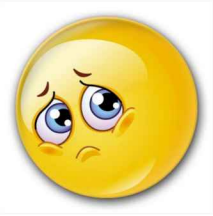
- Such competition can be induced/implemented by having lateral inhibition connections (negative feedback paths) between the neurons

- The result is that the neurons are forced to organize themselves

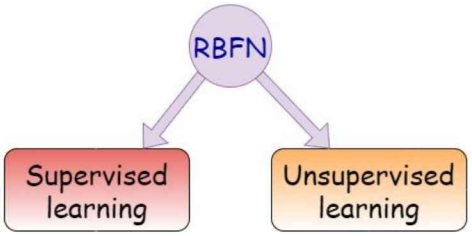
-

- SOM was used in face recognition (Alqudah and Al-Zoubi, 2015; Anggraini, 2014)

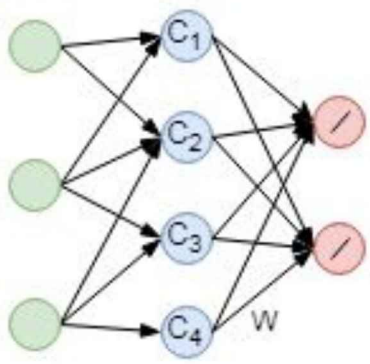
- However, they are not treated as a deep learning method



RBFN: Radial Basis Function Network



- Built upon function approximation theory in mathematics
- Consists of 3 layers: input layer, hidden layer, output layer



- The hidden units known as radial centers provide a set of functions that constitute an arbitrary basis for the input patterns
- The mapping from:
  - input to a high dimension hidden space is nonlinear
  - hidden to output space is linear

- With a sufficient number of radial basis function units, RBFN can also be a universal approximator
- The hidden units can use different radial functions:

Gaussian	$\Phi(z) = e^{-\frac{z^2}{2\sigma^2}}$
Thin Plate Spline	$\Phi(z) = z^2 logz$
Quadratic	$\Phi(z) = (z^2 + r^2)^{1/2}$
Inverse Quadratic	$\Phi(z) = \frac{1}{(z^2+r^2)^{1/2}}, z = \|\|x - c_j\|\|$

- Since RBFN exhibits several advantages:
  - global optimal approximation
  - classification capabilities
- it has been found to be very attractive for many engineering problems, including face recognition
- However, they are not considered as deep learning methods for face recognition

Hybrid Architectures

- Combine two or more types of neural networks
  - AE+DBM
  - CNN+AE
  - GAN+CNN, etc.

Table 7 Overview of deep methods using hybrid architectures

Algorithm	Description/Remark
Goswami et al (2017)	SDAE; DBM; For crossmodality learning
Nagpal et al (2015)	SDAE; DBM; Learn weight invariant facial representations
MM-DFR (Ding and Tao, 2015)	CNNs: extracts complementary facial features; SAE: compress dimension
Convnet-RBM (Sun et al, 2013)	CNN: characterize face similarities; RBM: perform inference
MDLFace (Goswami et al, 2014)	SDAE: robust to noise; RBM: learn internal complex representation; DNN
McDFR (Chen et al, 2015c)	Deep AE: extract generic feature of each facial regions; DNN: get discriminative feature; DNN: classification
Zhang et al (2017c)	GAN:generative capacity; CNN: discriminative feature extraction
Gan et al (2014)	multi-layer network architecture; graph embedding framework

❑ Stacked denoising sparse autoencoder + DBM

- Goswami et al (2017)
  - ✔ built a deep learning framework for video based FR

----See video section

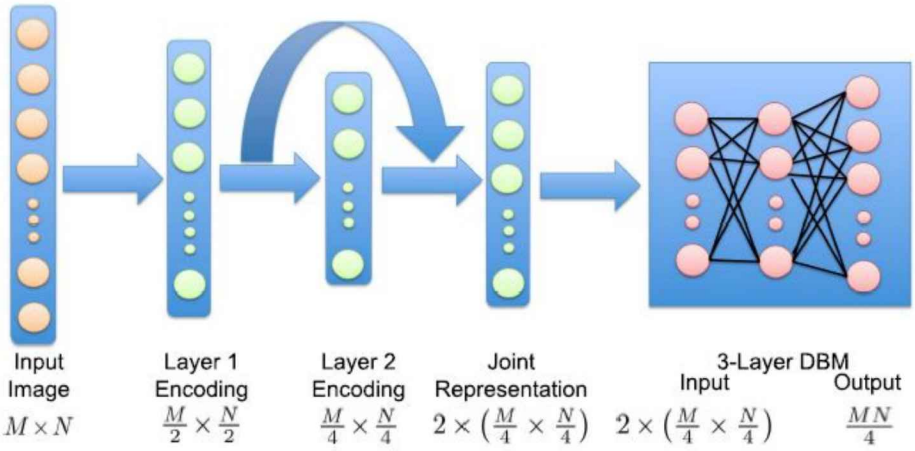


Fig. 5. Proposed deep learning architecture for facial representation: from input layer (image), two hidden layer representations are computed using SDAE encoding function. A joint representation is then obtained which combines the information from two SDAE encoding layers. Using joint representation as input, a DBM is used for computing a final feature vector.

Goswami G, Vatsa M, Singh R (2017) Face verification via learned representation on feature-rich video frames. trans on Information Forensics and Security 12(7):1686–1698

❑ GAN + CNN

- Zhang et al (2017c) combined
  - ✔ the generative capacity of conditional GAN (cGAN)
  - ✔ and the discriminative feature extraction of CNN
  - ✔ for cross-modality learning

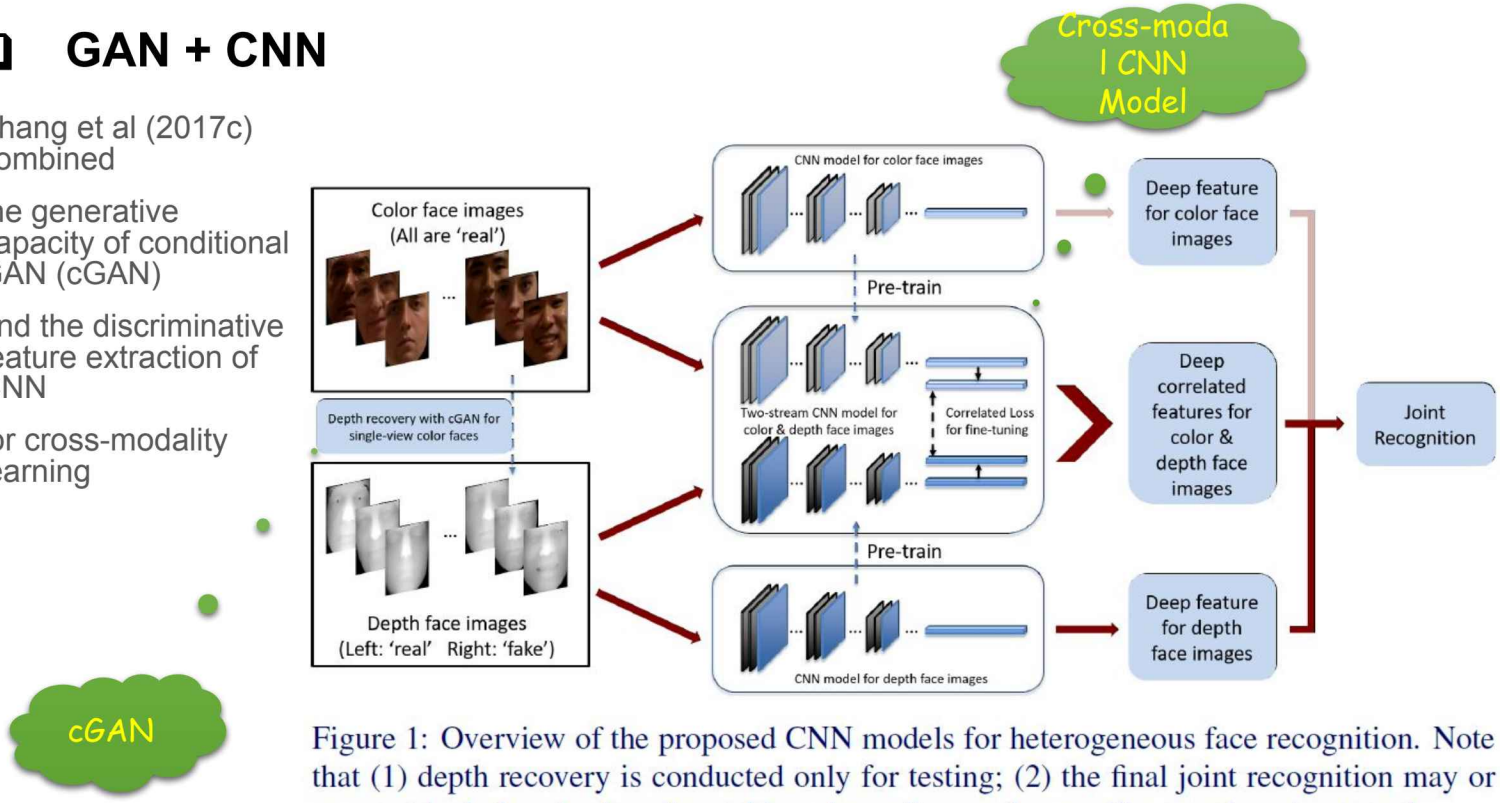
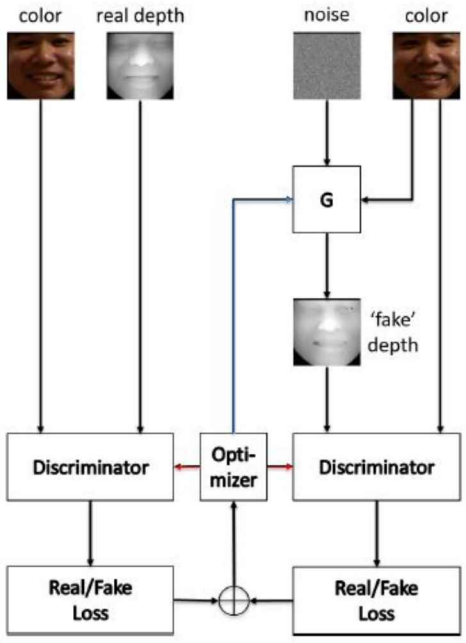


Figure 1: Overview of the proposed CNN models for heterogeneous face recognition. Note that (1) depth recovery is conducted only for testing; (2) the final joint recognition may or may not include color based matching, depending on the specific experiment protocol.

Zhang W, Shu Z, Samaras D, Chen L (2017c) Improving heterogeneous face recognition with conditional adversarial networks. arXiv preprint

cGAN



(a) Workflow of cGAN

- Training data contains image pairs {x,y}, where x and y refer to the depth and color faces respectively with a one-to-one correspondence between them
- y (color faces) can be involved in the model as a prior for generative task
- The optimization for cGAN:
  - the mini-batch SGD and the Adam solver are applied to optimize G and D alternately

Mirza M, Osindero S (2014) Conditional generative adversarial nets. arXiv preprint arXiv:14111784

Cross-modal CNN Model

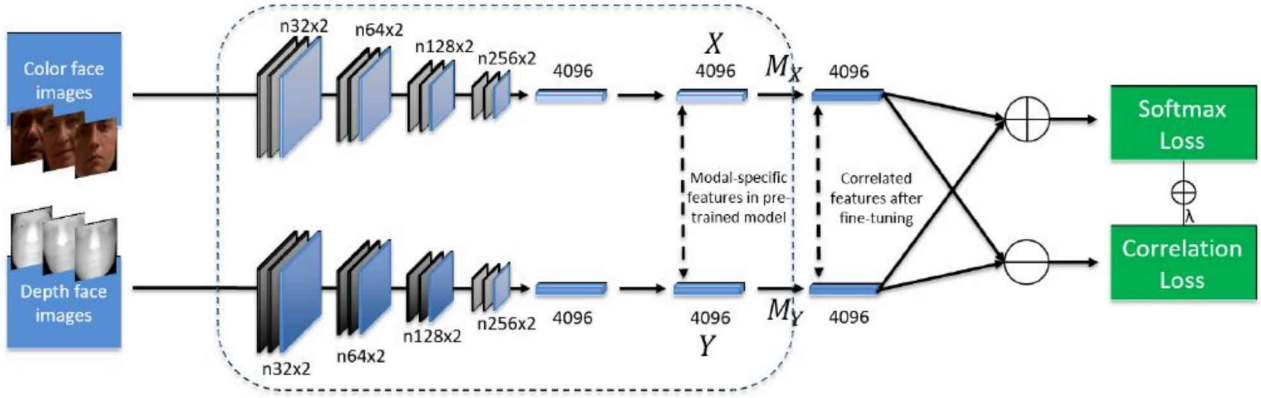


Figure 3: Training procedure of the cross-modal CNN model. Models in the dashed box are pre-trained using 2D and 2.5D face images individually.

- Once a pair of unimodal models for both views (depth and color) are trained, the modal-specific representations, {X,Y}, can be obtained after the last fully connected layers
- a joint supervision is required to enforce both correlation and distinctiveness simultaneously

❑ Auto-Encoders + DNNs

- McDFR (Chen et al, 2015c)
  - ✔ produce a generically descriptive yet class-specific deep multi-channel representation
  - ✔ use unsupervised and supervised learning in a cascaded fashion

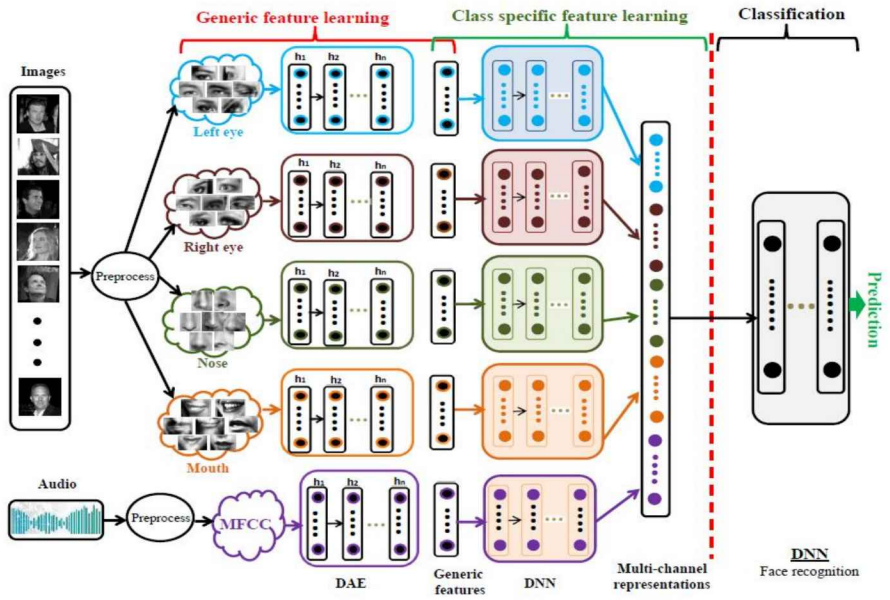
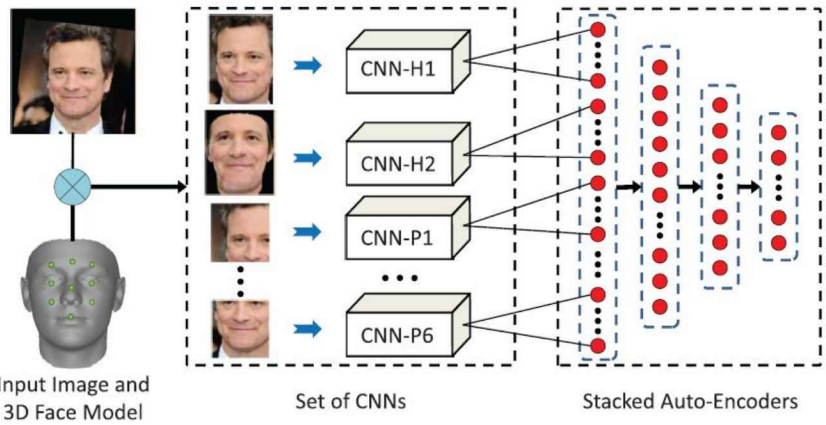


Figure 1: Outline of the proposed multi-channel deep feature representations for face recognition. The input data (images or other multimedia data) is first preprocessed. In each channel, generic features are learned on a (unlabeled) facial region or audio data through a deep autoencoder (DAE), and then class specific features are learned under supervision by feeding generic features into a DNN. The learned features from multiple channels are fused together as the final representation which is used as input to another DNN for classification.

❑ CNN + Stacked Auto-Encoder

- MM-DFR (Ding and Tao, 2015)
  - ✔ integrated a set of elaborately designed CNNs and a three-layer SAE
  - ✔ The CNNs extract complementary facial features from multimodal data
  - ✔ the extracted features are concatenated to form a high-dimensional feature vector, whose dimension is compressed by the SAE



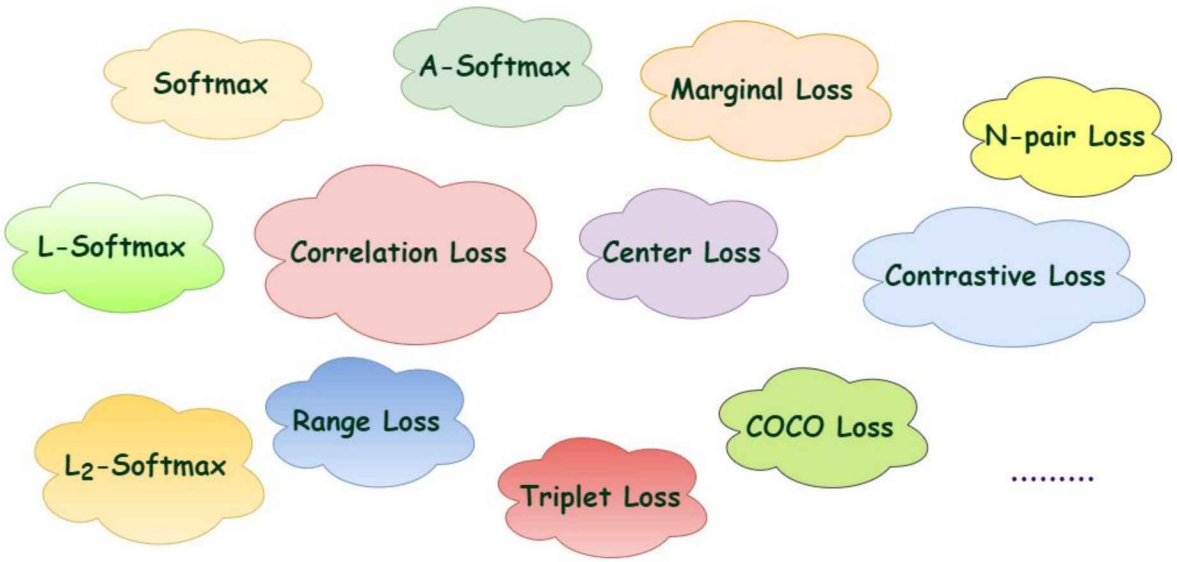
Ding C, Tao D (2015) Robust face recognition via multimodal deep face representation. IEEE trans on Multimedia 17(11):2049–2058

Fig. 2. Flowchart of the proposed multimodal deep face representation (MM-DFR) framework. MM-DFR is essentially composed of two steps: multimodal feature extraction using a set of CNNs and feature-level fusion of the set of CNN features using SAE.

Loss Functions

- In various deep neural networks, usually there is a loss layer, normally the final layer, which specifies how to penalize the deviation between the predicted and true labels in training
- An effective loss function:
  - can improve the discriminative power of the deeply learned features
- Intuitively, the learning should:
  - minimize the intra-class variations and maximize the extra-class differences

- Various loss functions have been proposed



Softmax Loss

- ✔ often used for predicting a single class of K mutually exclusive classes

$$L_s = -\sum_{i=1}^{m} log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}}$$

m: classes, W: weights, b: bias

Angular Softmax (A-Softmax)

- ✔ Adds an angular margin to softmax loss
- ✔ Renders a geometric interpretation by constraining learned features to be discriminative on a hypersphere manifold, which intrinsically matches the prior that faces also lie on a nonlinear manifold

$$L_{ang} = \frac{1}{N}\sum_{i} -log\frac{e^{\|x_i\|\varphi(\theta_{y_i,i})}}{e^{\|x_i\|\varphi(\theta_{y_i,i})} + \sum_{j\neq y_i} e^{\|x_i\|\cos(\theta_{j,i})}}$$

$\varphi(\theta_{y_i,i}) = (-1)^k \cos(m\theta_{y_i,i}) - 2k$

$\theta_{y_i,i} \in [\frac{k\pi}{m}, \frac{(k+1)\pi}{m}], k \in [0, m-1],$

m(≥ 1): an integer controling the size of angular margin

Large-margin Softmax (L-Softmax)

- ✔ To explicitly encourage intra-class compactness and inter-class separability for the learned features
- ✔ It can not only adjust the desired margin but also avoid overfitting

$$L_i = -log\left(\frac{e^{\|W_{y_i}\|\|x_i\|\varphi(\theta_{y_i})}}{e^{\|W_{y_i}\|\|x_i\|\varphi(\theta_{y_i})} + \sum_{j\neq y_i} e^{\|W_j\|\|x_i\|\cos(\theta_j)}}\right)$$

$$\varphi(\theta) = \begin{cases} \cos(m\theta), & \text{if } 0 \leq \theta \leq \frac{\pi}{m} \\ \mathcal{D}(\theta), & \text{if } \frac{\pi}{m} \leq \theta \leq \pi \end{cases}$$

where m is a integer that is closely related to the classification margin, $\mathcal{D}(\theta)$ is a monotonically decreasing function and $\mathcal{D}(\frac{\pi}{m})$ should equal $\cos(\frac{\pi}{m})$

L2-Softmax

- ✔ add an L2-constraint to softmax loss
- ✔ restricts the features to lie on a hypersphere of a fixed radius

$$\mathcal{L}_{L_2} = -\frac{1}{M}\sum_{i=1}^{M} log\frac{e^{W_{y_i}^T f(X_i) + b_{y_i}}}{\sum_{j=1}^{C} e^{W_j^T f(X_i) + b_j}}$$

minimizes $\mathcal{L}_{L_2}$ subject to $\|f(X_i)\|_2 = \alpha, \forall$ i=1,2,....M,

X_i:input in a mini-batch of size M, y_i:class label, C:# classes

$f(X_i)$: feature descriptor obtained from the penultimate layer

W,b: weights, bias for the last layer which acts as a classifier

Correlation Loss

- ✔ encourage the large correlation between the deep feature vectors and their corresponding weight vectors in softmax loss
- ✔ applies a weight vector in softmax loss as the prototype of each class

$$\mathcal{L}_C = -\sum_i \cos(\theta_{y_i}) = -\sum_i \frac{W_{y_i}^T x_i}{\|W_{y_i}\|\|x_i\|}$$

W_{y_i}: weight vector

Contrastive Loss

- ✔ runs over pairs of samples

$$L(W, Y, \overrightarrow{X_1}, \overrightarrow{X_2}) = (1-Y)\tfrac{1}{2}(D_W)^2 + (Y)\tfrac{1}{2}max(0, m - D_W)^2$$

Let $X_1, X_2 \in I$ be a pair of input vectors
Let Y be a binary label assigned to this pair
Y = 0 if X_1 and X_2 are deemd similar
Y = 1 if they are deemed dissimilar
Define the parameterized distance function to be learned D_W between X_1, X_2 as the euclidean distance between the outputs of G_W
m (> 0): a margin

Range Loss

- ✔ Inspired by contrastive loss
- ✔ to utilize the tailed data in training
- ✔ can reduce the overall intra-personal variations and enlarge inter-personal differences simultaneously
- ✔ unlike the contrastive loss defined on individual positive and negative pairs, range loss is defined on the overall distances between all sample pairs within one minibatch

$$\mathcal{L}_R = \alpha\mathcal{L}_{R_{intra}} + \beta\mathcal{L}_{R_{inter}}$$

where α and β are two weights
$\mathcal{L}_{R_{intra}}$ denotes the intra-class loss
$\mathcal{L}_{R_{inter}}$ represents the inter-class loss

Triplet Loss

- ✔ aims at ensuring a face image of a specific person (anchor) is closer to other images of the same person (positive) than to images of any other persons (negative)

$$\|x_i^a - x_i^p\|_2^2 + \alpha < \|x_i^a - x_i^n\|_2^2, \forall (x_i^a, x_i^p, x_i^n) \in \tau$$

α: a margin, τ: set of all possible triplets

$$L = \sum_i^N [\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha]_+$$

Multi-class N-pair Loss

- ✔ Since contrastive loss and triplet loss often leads to a slow convergence
- ✔ address this problem
- ✔ can significantly improves upon the triplet loss by pushing away multiple negative examples jointly at each update

$$\mathcal{L}_{N-pair-mc}(\{(x_i, x_i^+)\}_{i=1}^N; f) = \frac{1}{N}\sum_{i=1}^{N} log(1 + \sum_{j \neq i} exp(f_i^\top f_i^+ - f_i^\top f_i^+))$$

Let x be an input data,
x^+ and x^- be positive and negative examples of x,
f be kernal taking x and generating an embedding vector f(x)

Marginal Loss

- ✔ to minimize intra-class differences and maximize interclass distances by focusing on the marginal samples

$$L_m = \frac{1}{m^2 - m}\sum_{i,j,i \neq j}^{m} (\xi - y_{ij}(\theta - \| \frac{x_i}{\|x_i\|} - \frac{x_j}{\|x_j\|} \|_2^2))$$

where x_i, x_j are two face samples,
θ is a threshold of distance,
$y_{ij} \in \pm$ shows whether faces x_i and x_j are from same or different classes,
ξ is error margin besides the classification hyperplane.

Center Loss

- ✔ 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

$$L_C = \frac{1}{2}\sum_{i=1}^{m} \|x_i - c_{y_i}\|_2^2$$

$c_{y_i} \in \mathcal{R}^d$: y_ith class center,
x_i: input vector, m: # classes

Contrastive-Center Loss

- ✔ center loss only considers intra-class compactness
- ✔ consider both intra-class compactness and inter-class separability
- ✔ by penalizing two contrastive values, i.e.,
  - distances of input to its corresponding class centers
  - the sum of the distances of input to its non-corresponding class centers

$$L_{ct-c} = \frac{1}{2}\sum_{i=1}^{m} \frac{\|x_i - c_{y_i}\|_2^2}{(\sum_{j=1, j \neq y_i}^{k} \|x_i - c_j\|_2^2) + \delta}$$

δ: constant for preventing denominator equal to 0

Congenerous Cosine (COCO) Loss

- ✔ consider both feature discrimination and polymerization by directly optimizing and comparing the cosine distance (similarity) between features
- ✔ has the softmax property to make features discriminative and keeps the idea of class centroid

$$\mathcal{L}^{COCO}(f^{(i)}, c_k) = -\sum_{i\in\mathcal{B},k} t_k^{(i)} logp_k^{(i)} = -\sum_{i\in\mathcal{B}} logp_{l_i}^{(i)}$$

$f^{(i)}$: feature vector of i-th sample;
$\mathcal{B}$: mini-batch; c_k: centroid of class k;
k: index along the class dimension in $\mathcal{R}^K$;
$t_k^{(i)} \in \{0, 1\}$: binary mapping of sample i based on its label l_i

There are some other loss functions:

- verification loss and classification loss used in DeepID2, DeepID2+
- Sigmoid Cross-entropy loss is used for predicting K independent probability values in [0,1]

In unsupervised deep learning, there are also some loss functions:

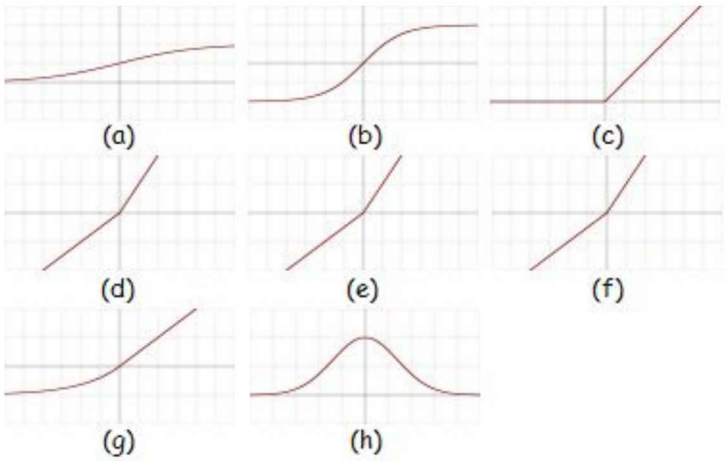
- Reconstruction error used in AE and its variants
- Square-loss function
- coupling error
- etc.

Activation Functions

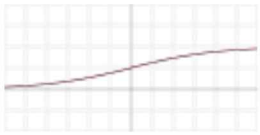
- Decides whether a neuron should be activated or not
  - If activated, it means the information that the neuron is receiving is relevant for the given information
  - otherwise the information will be ignored
- Nonlinear transformation
- The transformed output is sent to next layer of neurons as their input

Table 9 Description of common activation functions

Activation function	Definition
Sigmoid	$f(x) = (1 + e^{-x})^{-1}$
Tanh	$f(x) = tanh(x) = \frac{2}{1+e^{-2x}} - 1$
ReLU	$f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases}$
LReLU	$f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.01 * x, & \text{if } x \leq 0 \end{cases}$
PReLU	$f(a, x) = \begin{cases} x, & \text{if } x > 0 \\ a * x, & \text{if } x \leq 0 \end{cases}$
RReLU	$f(a_i, y_i) = \begin{cases} y_i, & \text{if } y_i > 0 \\ a_i * y_i, & \text{if } y_i \leq 0 \end{cases}$
ELU	$f(a, x) = \begin{cases} x, & \text{if } x > 0 \\ a(e^x - 1), & \text{if } x \leq 0 \end{cases}$
Maxout	$max(w_1^T x + b_1, w_2^T x + b_2)$
Gaussian	$\Phi(z) = e^{-\frac{z^2}{2\sigma^2}}$
Thin Plate Spline	$\Phi(z) = z^2 logz$
Quadratic	$\Phi(z) = (z^2 + r^2)^{1/2}$
Inverse Quadratic	$\Phi(z) = \frac{1}{(z^2+r^2)^{1/2}}, z = \|x - c_j\|$



Activation Function. (a) Sigmoid (b) Tanh (c) ReLU (d) LReLU (e) PReLU (f) RReLU (g) ELU (h) Gaussian



$$f(x) = (1 + e^{-x})^{-1}$$

Sigmoid

- ✔ Squashes real-valued number into the range between 0 and 1
- ✔ However, the sigmoid is rarely used in deep networks:
  - when the activation of a neuron saturates at either tail of 0 or 1, the gradient there is almost zero, resulting in almost no signal flowing through the neuron to its weights, and recursively to its data
  - the sigmoid outputs are not zero-centered

$$f(x) = tanh(x) = \frac{2}{1+e^{-2x}} - 1$$

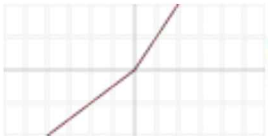
Tanh:

- ✔ Squashes a real-valued number to the range of [-1, 1]
- ✔ Like sigmoid neuron, its activations saturate
- ✔ but unlike the sigmoid neuron, its output is zero-centered
- ✔ Therefore, in practice the tanh nonlinearity is preferable than the sigmoid

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases}$$

ReLU: Rectified Linear Units

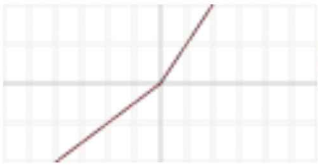
- ✔ It increases the nonlinear properties of the decision function and overall network without affecting the receptive fields of the convolution layer
- ✔ It trains the neural network faster without a significant penalty to generalization capability
- ✔ Compared to tanh and sigmoid neurons that involve expensive operations, ReLU can be implemented by simply thresholding a matrix of activations at zero
- ✔ Offers a way to separate noisy data from informative signals
- ✔ If a neuron is not activated, its output value will be 0
- ✔ However, this thresholding might lead to the loss of some information, especially for the first several convolution layers.
- ✔ LReLU, PReLU and ELU are proposed to alleviate this problem



$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.01 * x, & \text{if } x \leq 0 \end{cases}$$

LReLU: Leaky rectified linear unit (Leaky ReLU)

- ✔ The motivation is to avoid zero gradients
- ✔ Experiments carried out by Maas et al (2013) showed that the LReLU has negligible impact on accuracy compared with ReLU
- ✔ Instead of the function being zero when x<0, a leaky ReLU will instead have a small negative slope (of 0.01, or so)
- ✔ Some researchers reported success with this form of activation function, but the results are not always consistent



$$f(a, x) = \begin{cases} x, & \text{if } x > 0 \\ a * x, & \text{if } x \leq 0 \end{cases}$$

PReLU: Parametric Rectified Linear Units

- ✔ *a* is a coefficient controlling the slope of the negative part
- ✔ When *a* = 0, it becomes ReLU
- ✔ When *a* is a learnable parameter, it is referred to PReLU
- ✔ Equivalent to $f(x) = max(0, x) + a \cdot min(0, x)$
- ✔ If *a* is small and fixed, PReLU becomes Leaky ReLU (LReLU) (*a* = 0.01)
- ✔ PReLU can be trained using backpropagation and optimized simultaneously with other layers

Maxout

- ✔ Generalizes ReLU and its leaky version
- ✔ It has the benefits of a ReLU unit (linear regime of operation, no saturation), while does not have its drawbacks
- ✔ Unlike ReLU, it doubles the number of parameters for every single neuron, leading to a higher number of parameters in total

Max-Feature-Map

- ✔ Proposed with the Light CNN (Wu et al, 2015)
- ✔ It can be treated as an extension of Maxout activation
- ✔ Different from Maxout activation that uses enough hidden neurons to approximate an arbitrary convex function, MFM suppresses only a small number of neurons to make the CNN models light and robust

- In RBFN, the hidden units often use as the activation function
  - Gaussian radial function
  - thin plate spline
  - quadratic
  - inverse quadratic