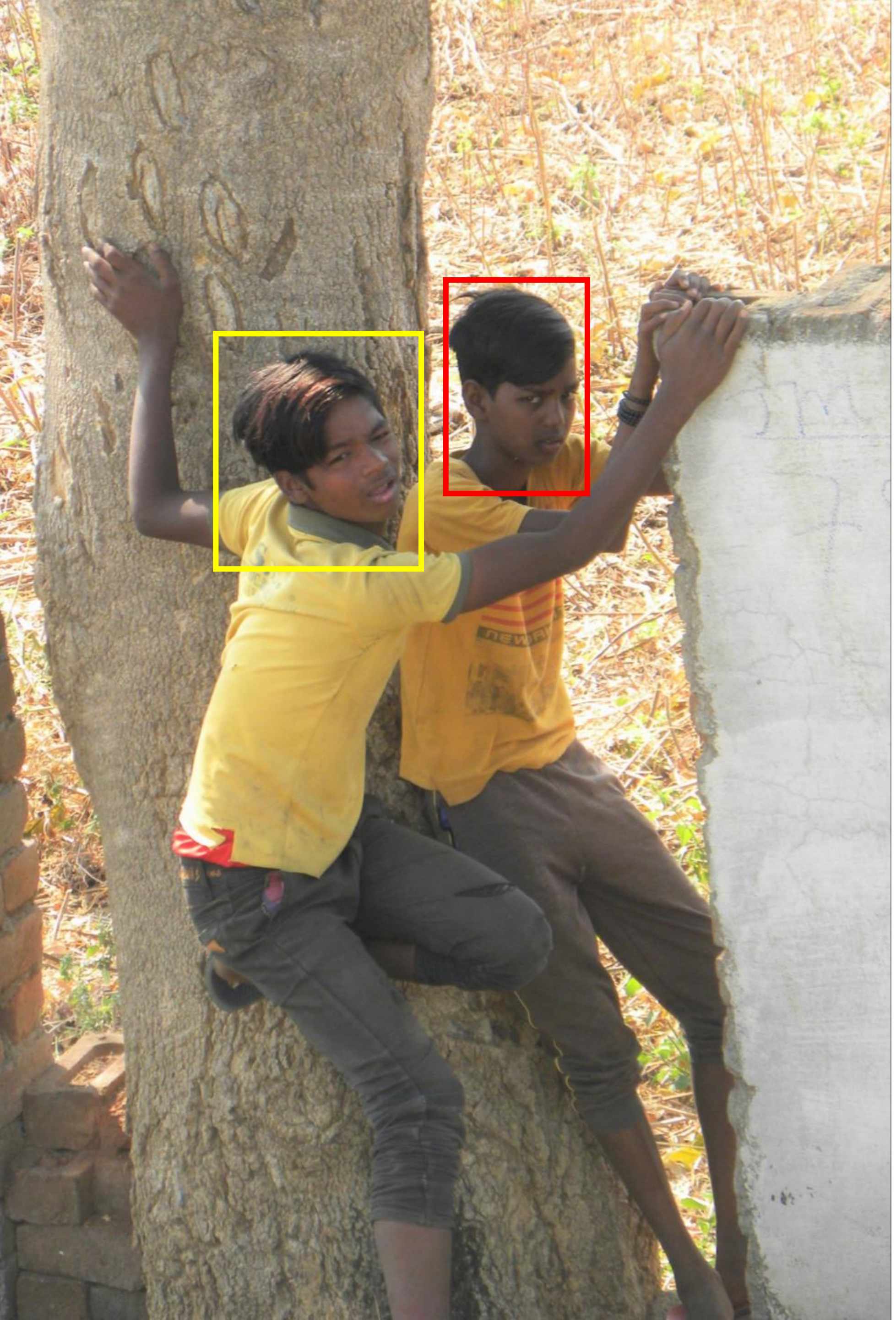
Face Recognition Using Deep Neural Networks

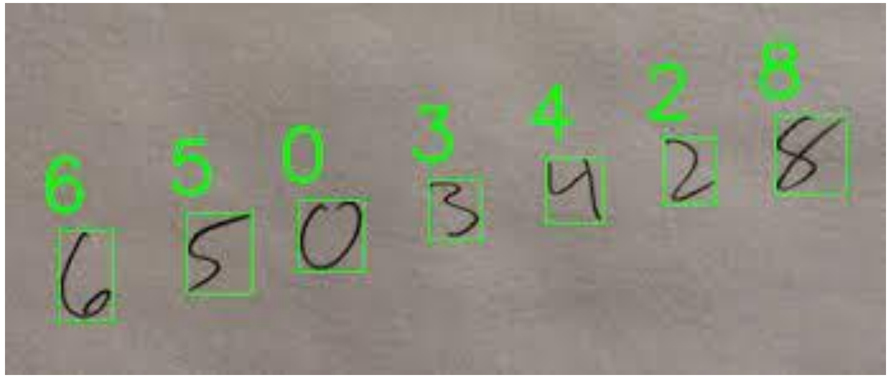


One Shot Learning

- Learning from one (or a few) examples to recognize a person.
- The idea is to understand the similarity between the detected object to a known object.

Learning Similarity

- d(img1,img2) = degree of difference between images



How can we measure similarity?

- We will find a function that quantifies a “distance” between every pair of elements in a set

  Non-negativity: $f(x, y) \geq 0$

  Identity of Discernible: $f(x, y) = 0 \Leftrightarrow x = y$

  Symmetry: $f(x, y) = f(y, x)$

  Triangle Inequality: $f(x, z) \leq f(x, y) + f(y, z)$

Distance Selection \Learning

- **Pre-defined Metrics**

Metrics which are fully specified without the knowledge of data.

E.g. Euclidian Distance:

$$\mathbf{f}(\vec{x}, \vec{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

Distance Selection \Learning

- **Learned Metrics**

Metrics which can only be defined with the **knowledge** of the **data**:

- Un-Supervised Learning

Or

- Supervised Learning

Un-supervised distance metric: :

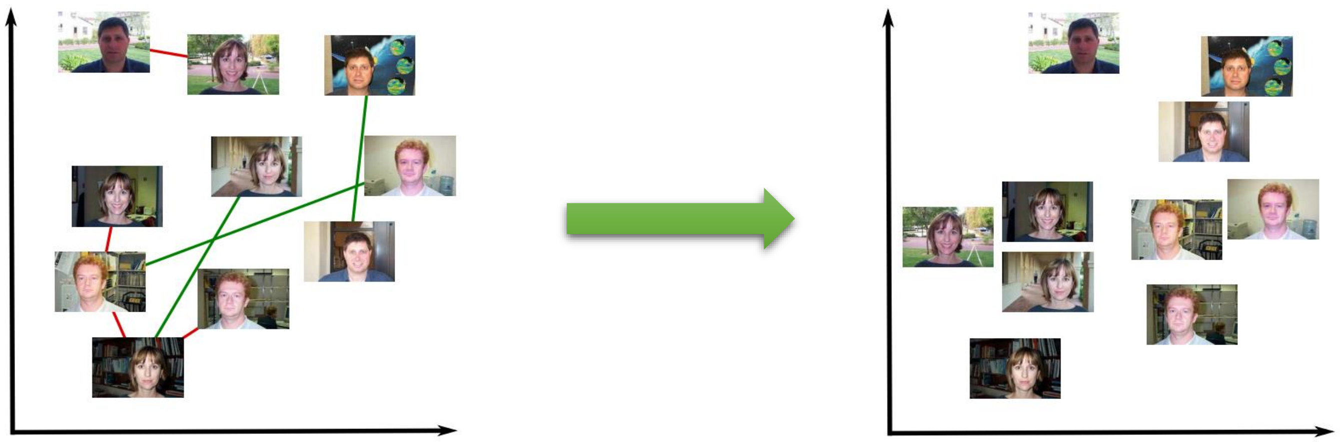
- Example - Mahalanobis Distance :

- $f(x, y) = (x - y)^T S^{-1}(x - y)$

  where S is the mean-subtracted covariance matrix of all data points.

Un-supervised distance metric:

- 2-step procedure:
  - Apply some **supervised** domain transform:



  - Then use one of the un-supervised metrics for performing the mapping.

Bellet, A., Habrard, A. and Sebban, A survey on metric learning for feature vectors and structured data, 2013

The One-Shot learning challenge

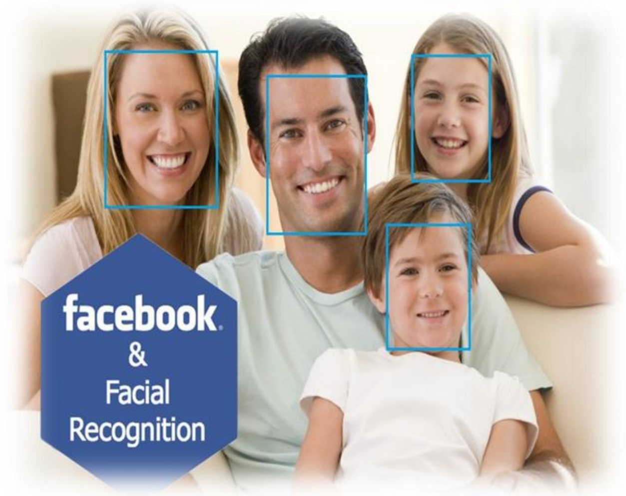
- There are lot of categories
- The Number of categories is not always known
- The number of samples in each category is small

▪ One shot learning is relevant in the field of **computer vision:** recognize objects in images from a single example.

Face Recognition challenges

Verification

Recognition

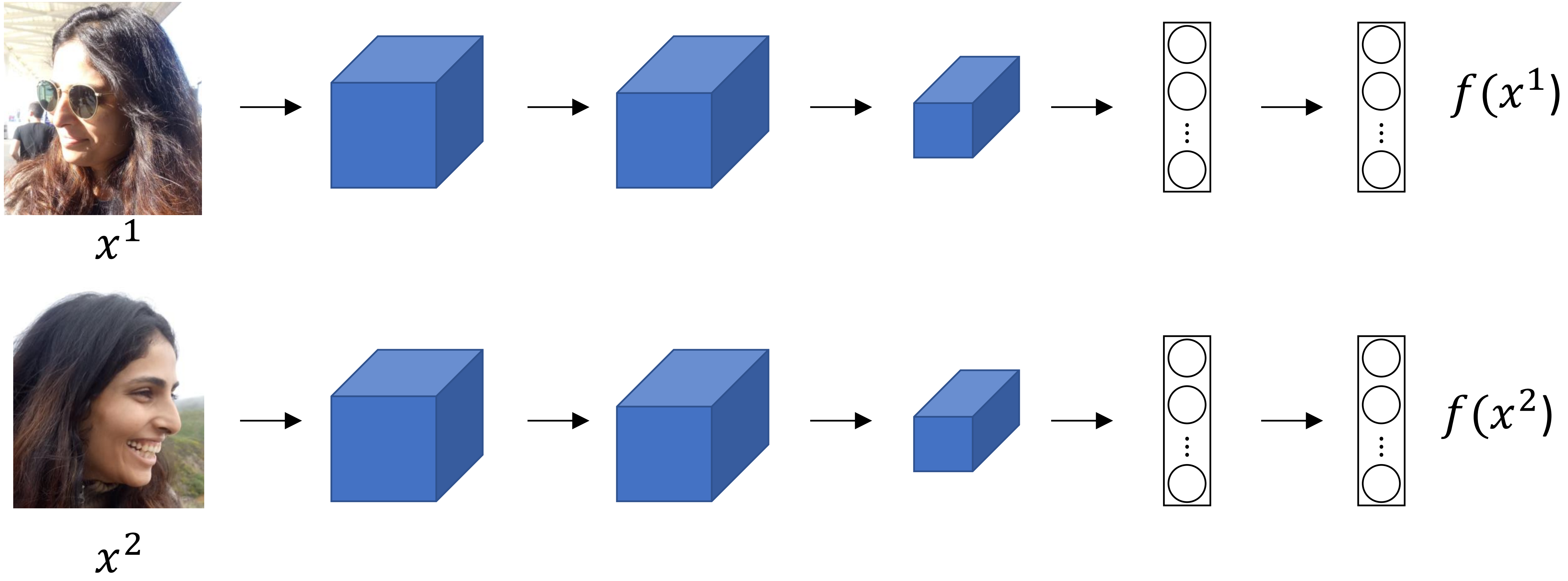


Clustering

Using CNN for Face Recognition

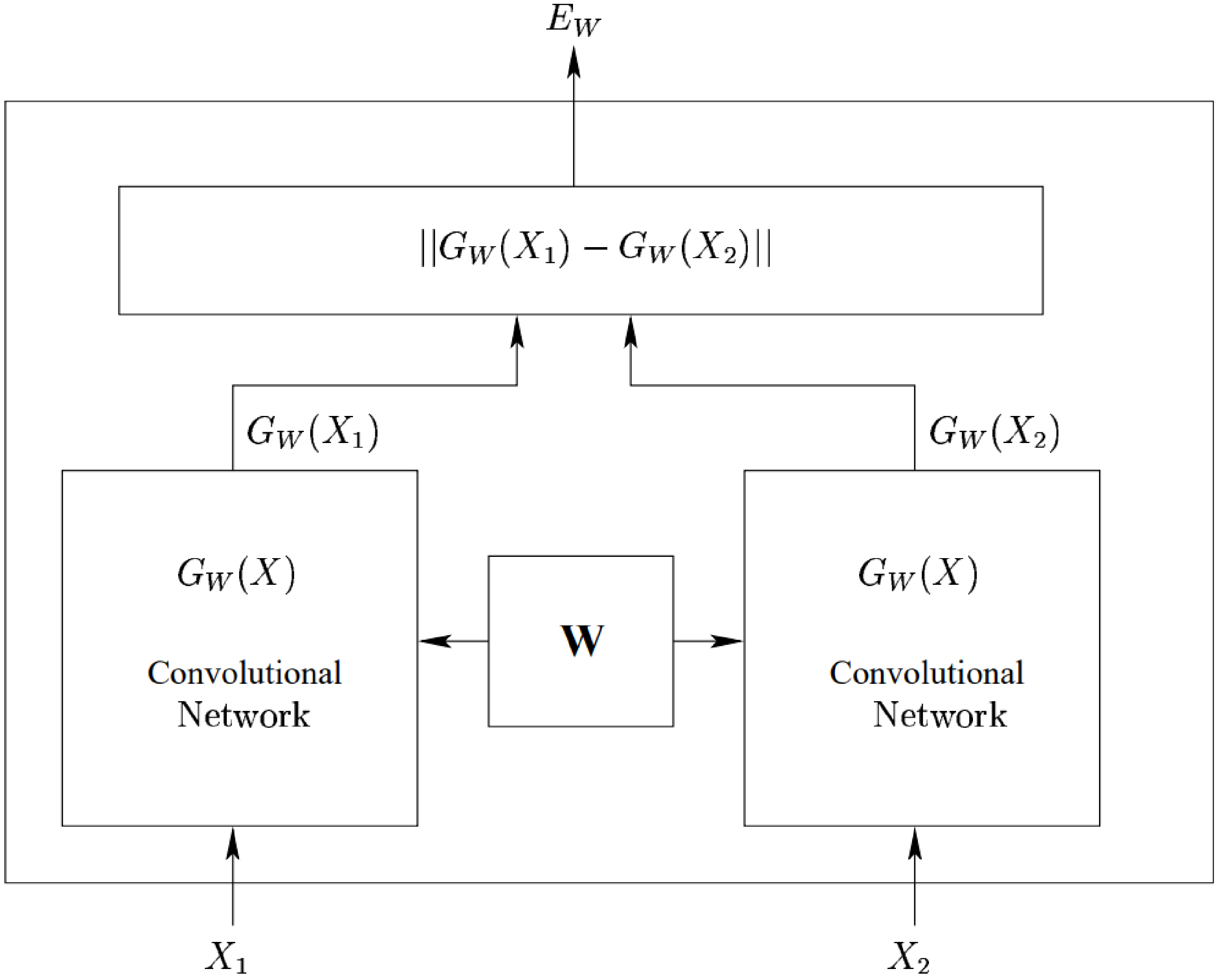
- The Idea is **to learn a function** that maps input patterns to target space.
- **non-linear** mapping that can map any input vector to its corresponding low-dimensional version.
- The distance in the target space approximates the “semantic” distance in the input space.
- Extract information about the problem from the available data, without requiring specific information about the categories.
- The training will be on pairs of samples.

Siamese network



[Taigman et. al., 2014. DeepFace closing the gap to human level performance, from lecture slides of Andrew Ng, Convolution Neural Network]

Siamese Network Architecture



Koch, Zemel, Salakhutdinov, Siamese Neural Networks for One-shot Image Recognition, 2015

Similarity Metric

$$E_W(X_1, X_2) = ||G_W(X_1) - G_W(X_2)||$$

We seek to find a value of the parameter W such that:

X_1, X_2 - Same Category

Genuine Pair

Minimizes $E_W(X_1, X_2)$

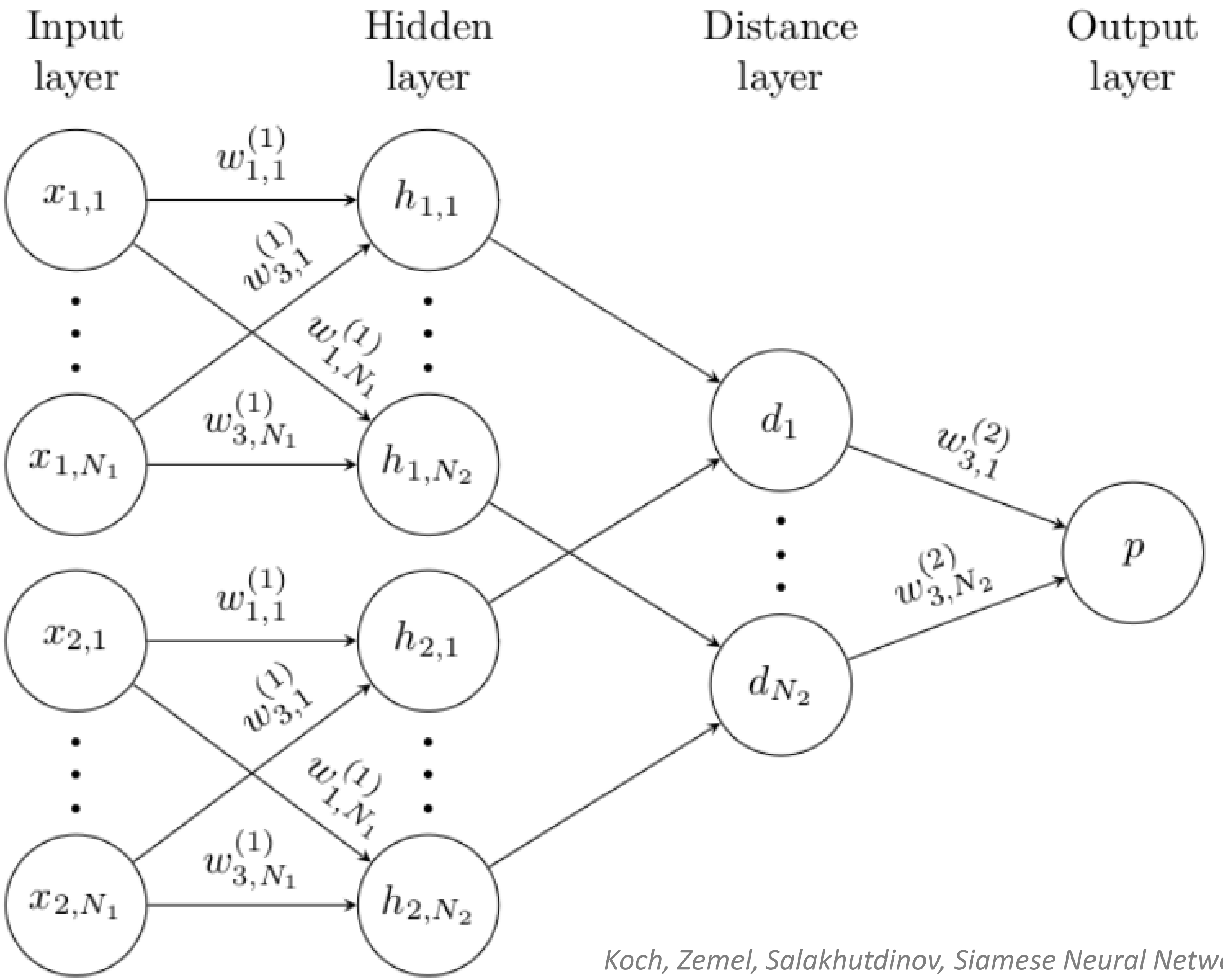
X_1, X_2 - Different Categories

Impostor Pair

Maximizes $E_W(X_1, X_2)$

- Contrastive term is needed to ensure: not only that the energy for a pair of inputs from the same category is low, but also that the energy for a pair from different categories is large.

Siamese Neural Networks



Koch, Zemel, Salakhutdinov, Siamese Neural Networks for One-shot Image Recognition, 2015

Contrastive Loss Function

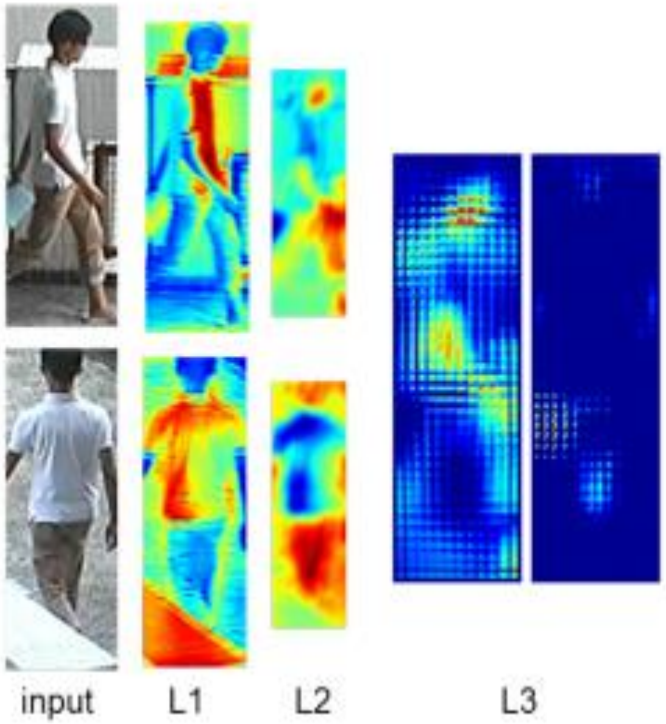
- Contrastive loss between a pair of samples X^i, X^j is defined as

$$L(X^i, X^j) = Y^{i,j} D\left(X^i, X^j\right) + (1 - Y^{i,j})\max(0, m - D\left(X^i, X^j\right))$$

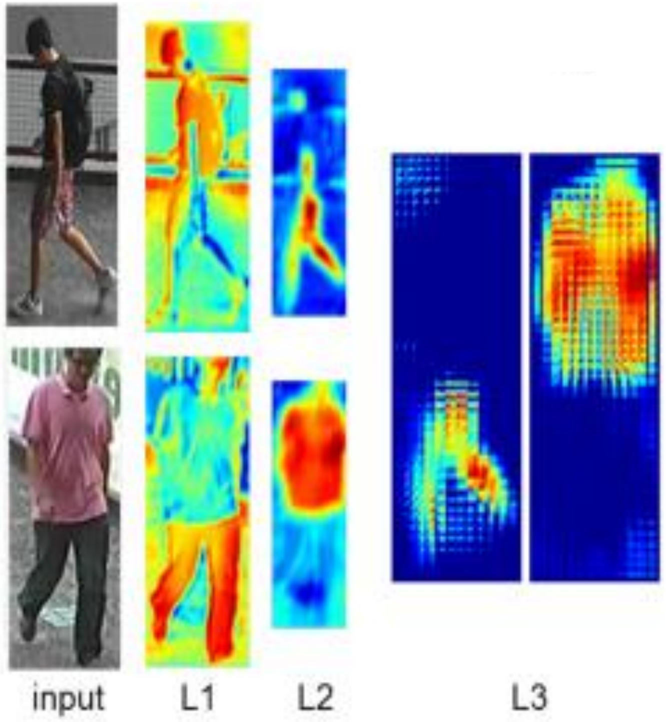
- Here $D\left(X^i, X^j\right)$ is the Square of the Euclidean distance and m is the margin for the distance that we set for dissimilar pair of samples.
- For a batch of input samples, we take it as the average contrastive loss.

Visualization of Learned Features

Positive Pair



Negative Pair



Ahmed, Jones, Marks, An improved deep learning architecture for person re-identification, 2015

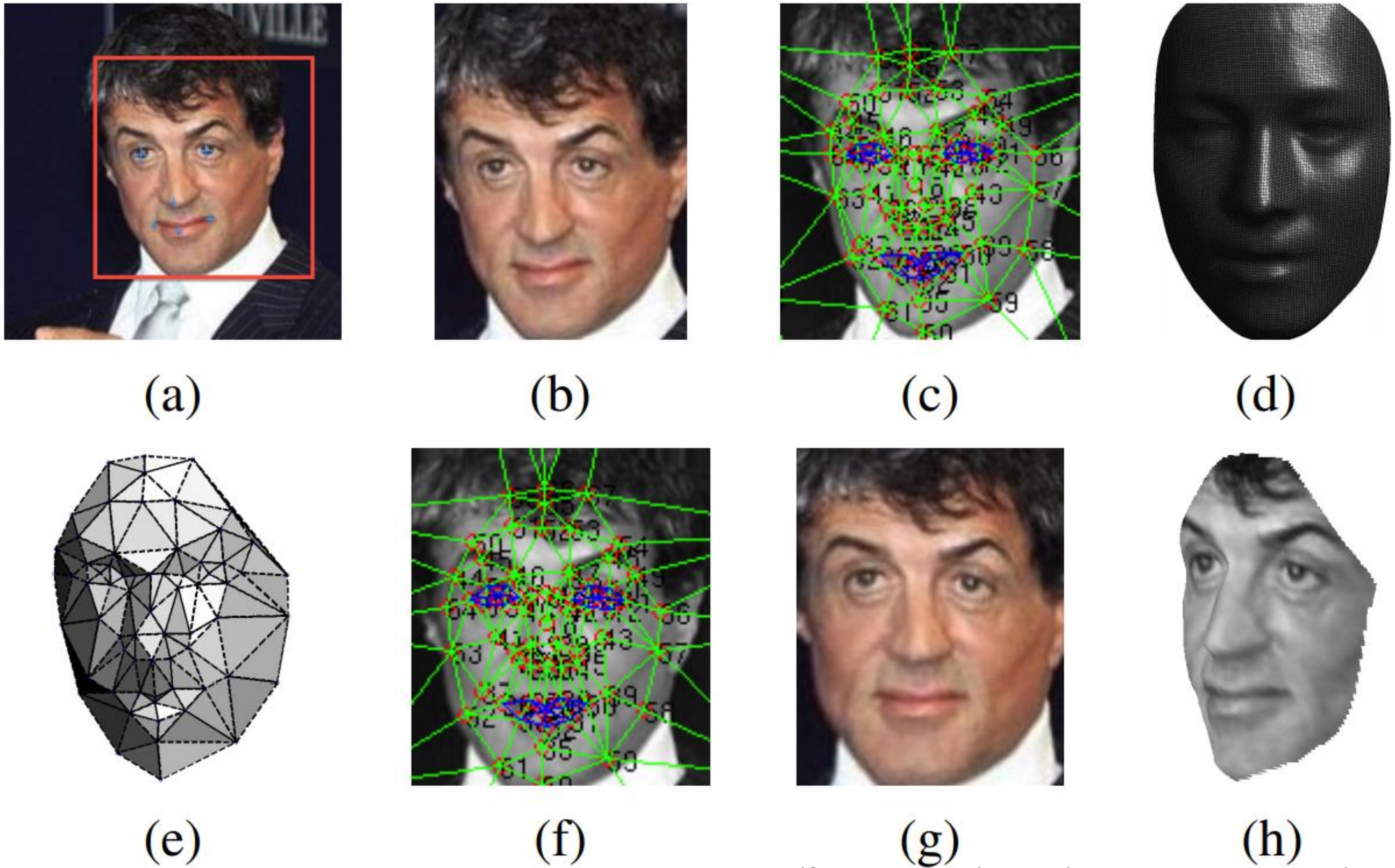
DeepFace (Facebook, 2014)

- The conventional pipeline:

  Detect ⇒ align ⇒ represent ⇒ classify
- **Face alignment**: Transform a face to be in a canonical pose
- **Face representation**: Find a representation of a face which is suitable for follow-up tasks (small size, computationally cheap to compare, invariant to irrelevant changes)
- 3D face modeling
- A nine-layer deep neural network
- More than 120 million parameters

Taigman; Yang; Ranzato, Wolf, DeepFace: Closing the Gap to Human-Level Performance in Face Verification, 2014

Alignment Process



Taigman; Yang; Ranzato, Wolf, DeepFace: Closing the Gap to Human-Level Performance in Face Verification, 2014

Alignment Process

(a) 2D Alignment - detecting 6 fiducial points inside the detection crop, centered at the center of the eyes, tip of the nose and mouth locations.

(b) 2D Alignment - aligned crop: composing the final 2D transformation.

(c) 2D Alignment - localizing additional 67 fiducial points

(d) 3D Alignment - The reference 3D shape transformed to the 2D-aligned crop image-plane.

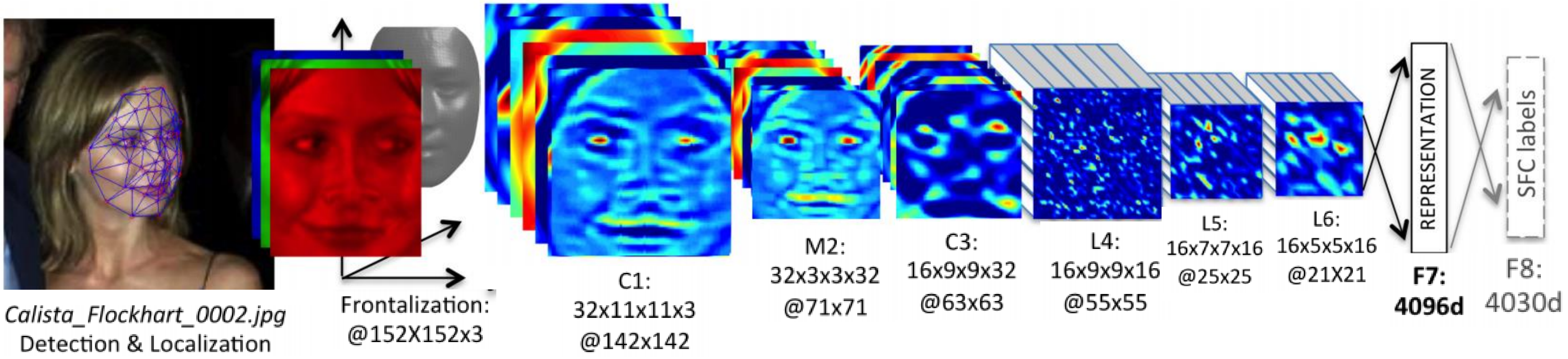
Taigman; Yang; Ranzato, Wolf, DeepFace: Closing the Gap to Human-Level Performance in Face Verification, 2014

Alignment Process

(e) 3D Alignment - The 67 fiducial points induced by the 3D model that are used to direct the piece-wise affine wrapping.

(f) 2D Alignment - The final frontalized crop

The DeepFace architecture



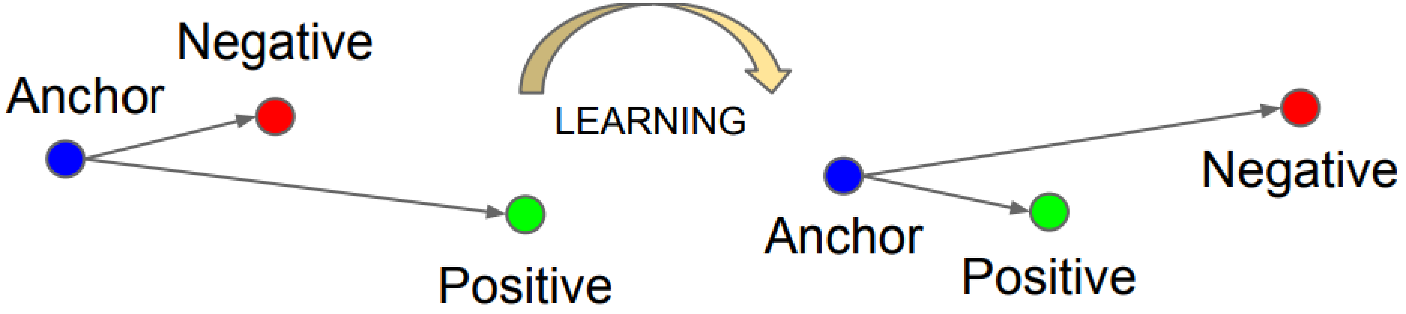
C1, M2, C3 : Extract low-level features (simple edges and texture)

L4, L5, L6: Locally connected Layers

L7, L8: Fully connected Layers

Taigman; Yang; Ranzato, Wolf, DeepFace: Closing the Gap to Human-Level Performance in Face Verification, 2014

Triplets Network



- 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, Kalenichenko, Philbin, FaceNet: A Unified Embedding for Face Recognition and Clustering, 2015

Triplet Network

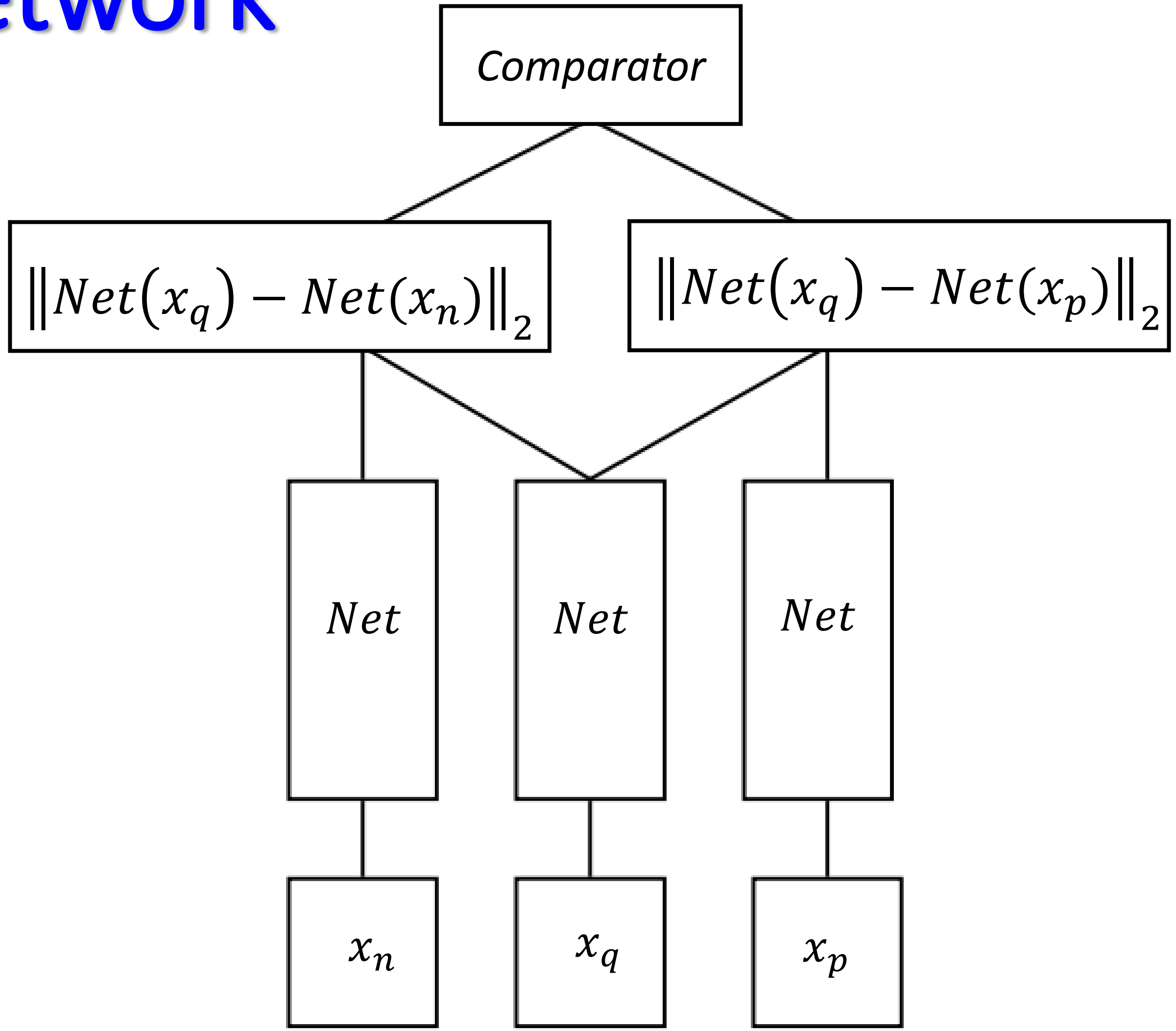


Figure 1: Triplet network structure

Hoffer, Ailon, DEEP METRIC LEARNING USING TRIPLET NETWORK , 2015

Training

- We learn an embedding f(x), from an image x into a feature space R d , such that the squared distance between all faces of the same identity is small, whereas the squared distance between a pair of face images from different identities is large.

- Loss Function:

$$L = \sum_{i}^{N} \left[\left\| f(x_{q,i}) - f(x_{p,i}) \right\|_2^2 - \left\| f(x_{q,i}) - f(x_{n,i}) \right\|_2^2 + \alpha \right]_+$$

$f(\quad)$- the embedding function

Schroff, Kalenichenko, Philbin, FaceNet: A Unified Embedding for Face Recognition and Clustering, 2015

Triplets Selection

- We could select $(x_{p,i})$ and $\left(x_{n,i}\right)$:

  (1) hard positive- argmax $(x_{p,i})$ $\left\|f\left(x_{q,i}\right) - f(x_{p,i})\right\|_2^2$

  (2) hard negative- argmin $(x_{q,i})$ $\left\|f\left(x_{q,i}\right) - f(x_{n,i})\right\|_2^2$

- For fast convergence it is crucial to select triplets that violate the triplet constraint.
- It is infeasible to compute the argmin and argmax across the whole training set.
- Further, it may lead to poor training, as mislabelled and poorly imaged faces would dominate the hard positives and negatives.

Schroff, Kalenichenko, Philbin, FaceNet: A Unified Embedding for Face Recognition and Clustering, 2015

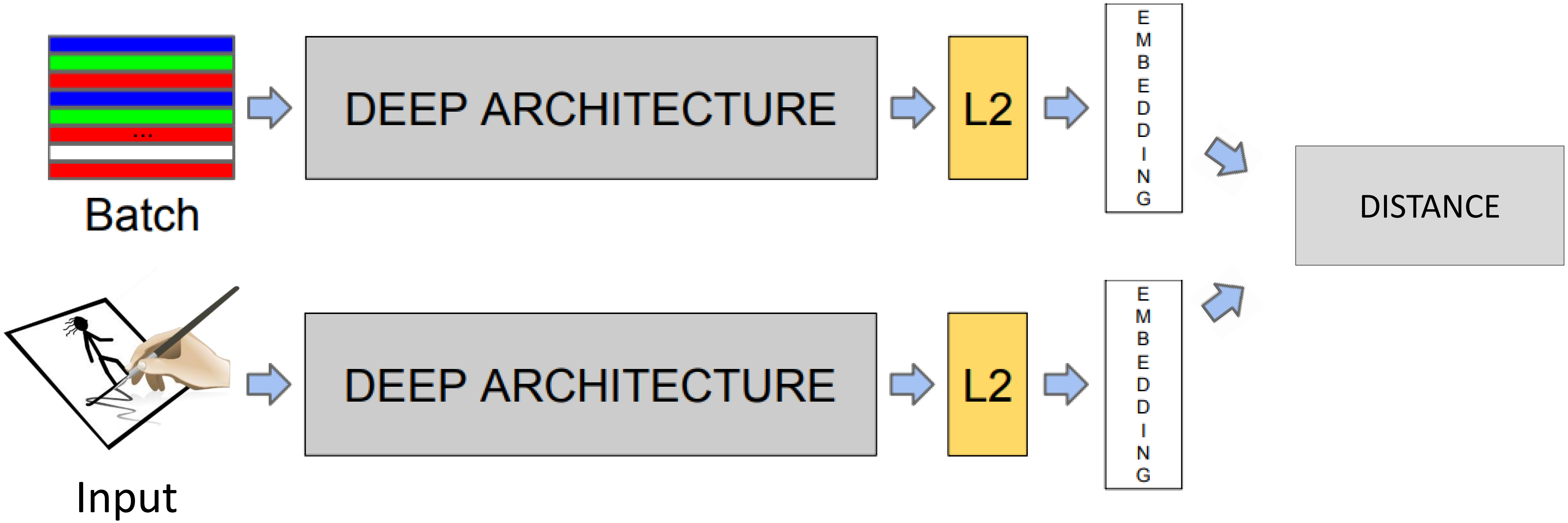
Triplet Selection

- There are two obvious choices that avoid this issue:
  - Generate triplets offline every n steps, using the most recent network checkpoint and computing the argmin and argmax on a subset of the data.
  - Generate triplets online. This can be done by selecting the hard positive/negative examples from within a mini-batch.
- To avoid local minima, we use ”semi-hard” examples such that

$$\left\| f(x_{q,i}) - f(x_{p,i}) \right\|_2^2 < \left\| f(x_{q,i}) - f(x_{n,i}) \right\|_2^2$$

Schroff, Kalenichenko, Philbin, FaceNet: A Unified Embedding for Face Recognition and Clustering, 2015

The Model structure
(after the training)



Schroff, Kalenichenko, Philbin, FaceNet: A Unified Embedding for Face Recognition and Clustering, 2015

The FaceNet architecture (NN1)

layer	size-in	size-out	kernel	param	FLPS
conv1	220×220×3	110×110×64	7×7×3, 2	9K	115M
pool1	110×110×64	55×55×64	3×3×64, 2	0	
rnorm1	55×55×64	55×55×64		0	
conv2a	55×55×64	55×55×64	1×1×64, 1	4K	13M
conv2	55×55×64	55×55×192	3×3×64, 1	111K	335M
rnorm2	55×55×192	55×55×192		0	
pool2	55×55×192	28×28×192	3×3×192, 2	0	
conv3a	28×28×192	28×28×192	1×1×192, 1	37K	29M
conv3	28×28×192	28×28×384	3×3×192, 1	664K	521M
pool3	28×28×384	14×14×384	3×3×384, 2	0	
conv4a	14×14×384	14×14×384	1×1×384, 1	148K	29M
conv4	14×14×384	14×14×256	3×3×384, 1	885K	173M
conv5a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv5	14×14×256	14×14×256	3×3×256, 1	590K	116M
conv6a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv6	14×14×256	14×14×256	3×3×256, 1	590K	116M
pool4	14×14×256	7×7×256	3×3×256, 2	0	
concat	7×7×256	7×7×256		0	
fc1	7×7×256	1×32×128	maxout p=2	103M	103M
fc2	1×32×128	1×32×128	maxout p=2	34M	34M
fc7128	1×32×128	1×1×128		524K	0.5M
L2	1×1×128	1×1×128		0	
total				140M	1.6B

Schroff, Kalenichenko, Philbin, FaceNet: A Unified Embedding for Face Recognition and Clustering, 2015

The FaceNet architecture (NN2)

type	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj (p)	params	FLOPS
conv1 (7×7×3, 2)	112×112×64	1							9K	119M
max pool + norm	56×56×64	0						m 3×3, 2		
inception (2)	56×56×192	2		64	192				115K	360M
norm + max pool	28×28×192	0						m 3×3, 2		
inception (3a)	28×28×256	2	64	96	128	16	32	m, 32p	164K	128M
inception (3b)	28×28×320	2	64	96	128	32	64	L_2, 64p	228K	179M
inception (3c)	14×14×640	2	0	128	256,2	32	64,2	m 3×3,2	398K	108M
inception (4a)	14×14×640	2	256	96	192	32	64	L_2, 128p	545K	107M
inception (4b)	14×14×640	2	224	112	224	32	64	L_2, 128p	595K	117M
inception (4c)	14×14×640	2	192	128	256	32	64	L_2, 128p	654K	128M
inception (4d)	14×14×640	2	160	144	288	32	64	L_2, 128p	722K	142M
inception (4e)	7×7×1024	2	0	160	256,2	64	128,2	m 3×3,2	717K	56M
inception (5a)	7×7×1024	2	384	192	384	48	128	L_2, 128p	1.6M	78M
inception (5b)	7×7×1024	2	384	192	384	48	128	m, 128p	1.6M	78M
avg pool	1×1×1024	0								
fully conn	1×1×128	1							131K	0.1M
L2 normalization	1×1×128	0								
total									7.5M	1.6B

Schroff, Kalenichenko, Philbin, FaceNet: A Unified Embedding for Face Recognition and Clustering, 2015

The results

- Performance on Youtube Faces DB: 95.12% accuracy
- Performance on Labeled Faces in the Wild DB: 99.63% accuracy

- Sensitivity to Image Quality:

(The CNN was trained on 220x220 input images)

# Pixels	Val-Rate
1,600	37.8%
6,400	79.5%
14,400	84.5%
25,600	85.7%
65,539	86.4%

Schroff, Kalenichenko, Philbin, FaceNet: A Unified Embedding for Face Recognition and Clustering, 2015

FaceNet – Image clustering

Schroff, Kalenichenko, Philbin, FaceNet: A Unified Embedding for Face Recognition and Clustering, 2015

References

- Chopra, Hadsell, LeCun, Learning a Similarity Metric Discriminatively, with Application to Face Verification , 2005
- *Taigman; Yang; Ranzato, Wolf, DeepFace: Closing the Gap to Human-Level Performance in Face Verification, 2014*
- Schroff, Kalenichenko, Philbin, FaceNet: A Unified Embedding for Face Recognition and Clustering, 2015
- Koch, Zemel, Salakhutdinov, Siamese Neural Networks for One-shot Image Recognition, 2015
- Hermans, Beyer, Leibe, In Defense of the Triplet Loss for Person Re-Identification, 2017
- LeCun, Chopra, Hadsell, Ranzato, Huang, A Tutorial on Energy-Based Learning, 2006
  - Bell, Bala, Learning visual similarity for product design with convolutional neural networks, 2015
  - Ahmed, Jones, Marks, An improved de ep learning architecture for person re-identification, 2015
  - Nando de Freias, Max-margin learning, transfer and memory networks, 2015
  - Jagannatha Rao, Wang, Cottrell, A Deep Siamese Neural Network Learns the Human-Perceived Similarity Structure of Facial Expressions Without Explicit Categories, 2011
  - Hoffer, Ailon, DEEP METRIC LEARNING USING TRIPLET NETWORK , 2015
  - Bellet, Habrard, Sebban, A survey on metric learning for feature vectors and structured data, 2013
  - Andre Ng, Lecture slides on Face Recognition, Coursera : Convolution Neural Network, 2017.