

EE 4211 Computer Vision

Lecture 11A: Face recognition with deep learning

Semester A, 2020-2021

Schedules

Week	Date	Topics
1	Sep. 4	Introduction/Imaging
2	Sep. 11	Image enhancement in spatial domain
3	Sep. 18	Image enhancement in frequency domain (HW1 out)
4	Sep. 25	Morphological processing
5	Oct. 2	Image restoration(HW1 due)
6	Oct. 9	Image restoration
7	Oct. 16	Midterm (no tutorials this week)
8	Oct. 23	Edge detection (HW2 out, illustrate the project)
9	Oct. 30	Image segmentation (HW2 due)
10	Nov. 6	Face recognition with PCA, LDA (tutorial on deep learning framework)
11	Nov. 13	Face recognition based on deep learning Image segmentation based on deep learning (tutorial on coding)
12	Nov. 20	Object detection with traditional methods (Quiz) Object detection based on deep learning
13	Nov. 27	Project presentation and summary
14	Dec. 4	Review and Summary

Quiz 2

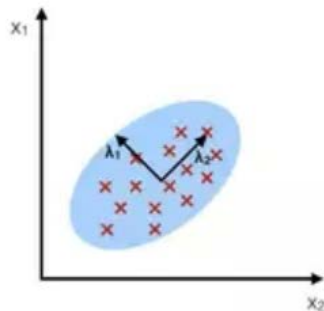
- The University adopts a **2-device approach** to examination monitoring.
- Students will be required to arrange a location, readied with a computer with camera and Zoom installed, plus a secondary device also with camera (on a stand) when taking the final examinations.
- Two cameras in Quiz 2 (open book, open notes)
- I will publish the Quiz 2 in the Canvas
- Time: 12:00 PM-13:00 PM on Nov. 20 (close at 13:10 PM, 10 mins for submission)
- No late submission is allowed

PCA vs. LDA

- Eigenfaces exploit the **max scatter** of the training images in face space
- Fisherfaces attempt to **maximize the between class scatter**, while **minimizing the within class scatter**.

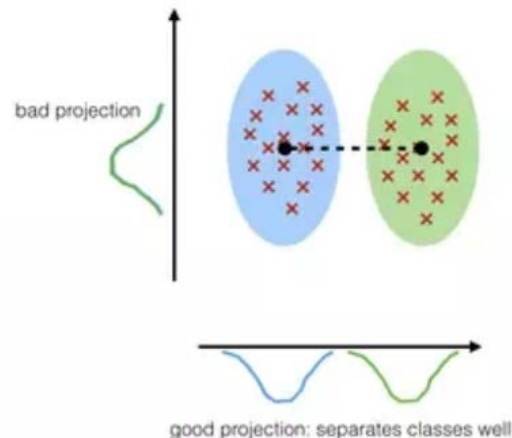
PCA:

component axes that maximize the variance



LDA:

maximizing the component axes for class-separation



Outline

- Introduction of deep learning
- DeepFace
- FaceNet

Overview

- Face recognition problem
 - Given a still image, identify one or more persons in this image using a stored database of facial images



Who is she?

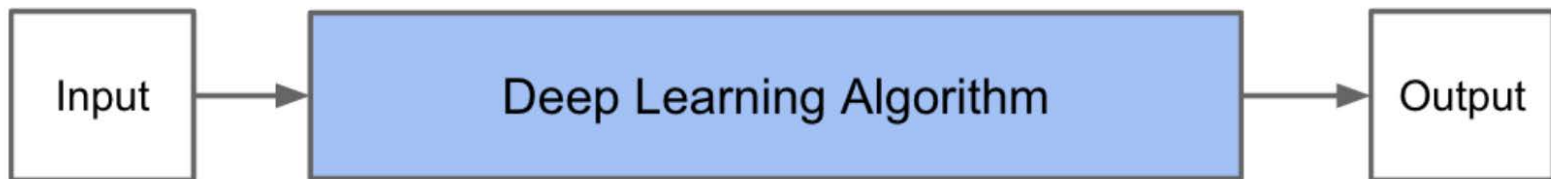


Machine learning vs deep learning

- Face recognition



Traditional Machine Learning Flow



Introduction of deep learning

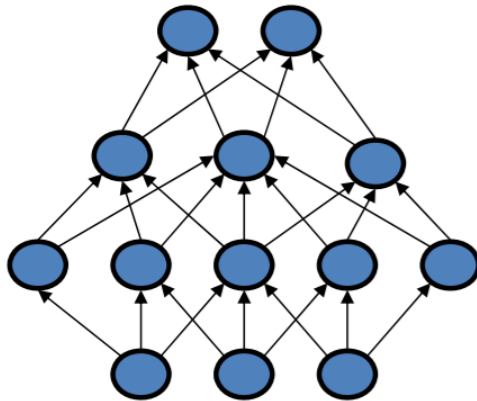
Neural network
Back propagation



Nature



1986



- Solve general learning problems
- Tied with biological system

Introduction of deep learning

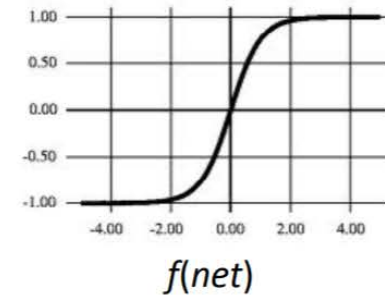
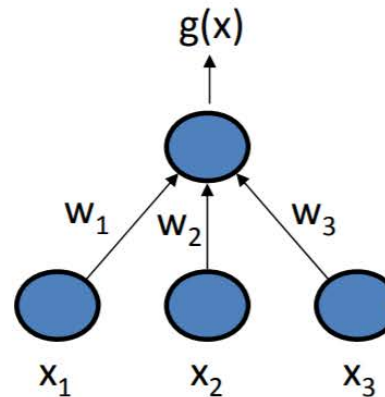
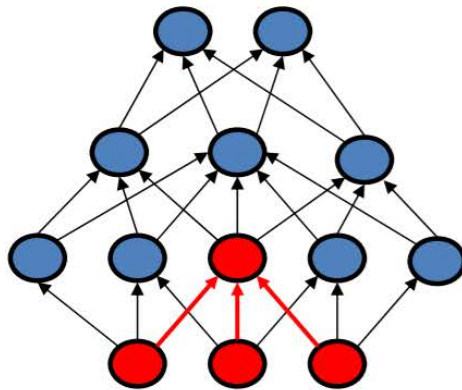
Neural network
Back propagation



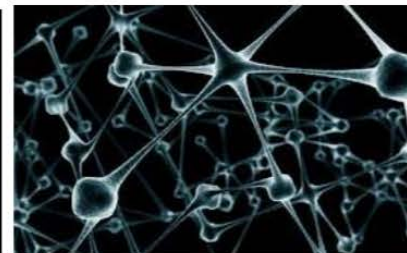
Nature



1986



$$g(\mathbf{x}) = f\left(\sum_{i=1}^d x_i w_i + w_0\right) = f(\mathbf{w}^t \mathbf{x})$$



Introduction of deep learning

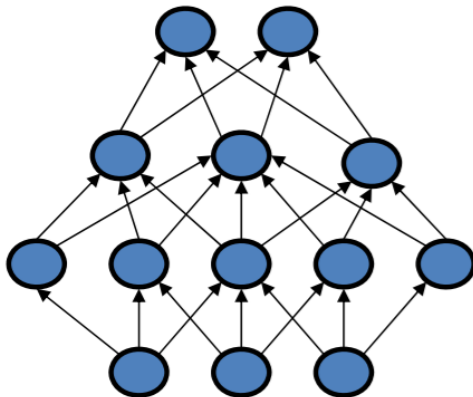
Neural network
Back propagation



Nature



1986



- Solve general learning problems
- Tied with biological system

But it is given up...

- Hard to train
- Insufficient computational resources
- Small training sets
- Does not work well

Introduction of deep learning

Neural network
Back propagation



Nature

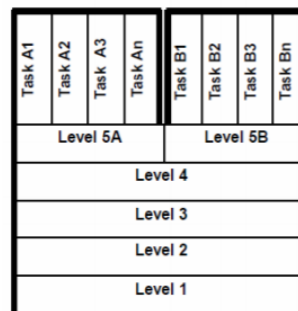


1986

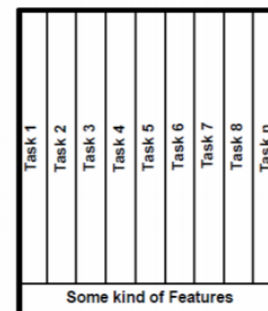
2006

- SVM
- Boosting
- Decision tree
- KNN
- ...
- Flat structures
- Loose tie with biological systems
- Specific methods for specific tasks
 - Hand crafted features (GMM-HMM, SIFT, LBP, HOG)

Deep Hierarchy

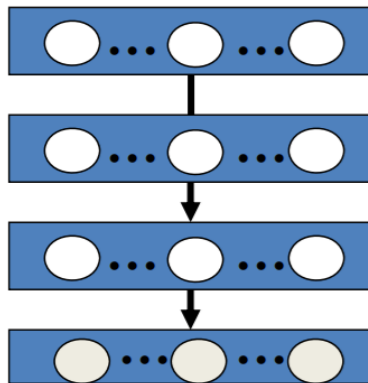
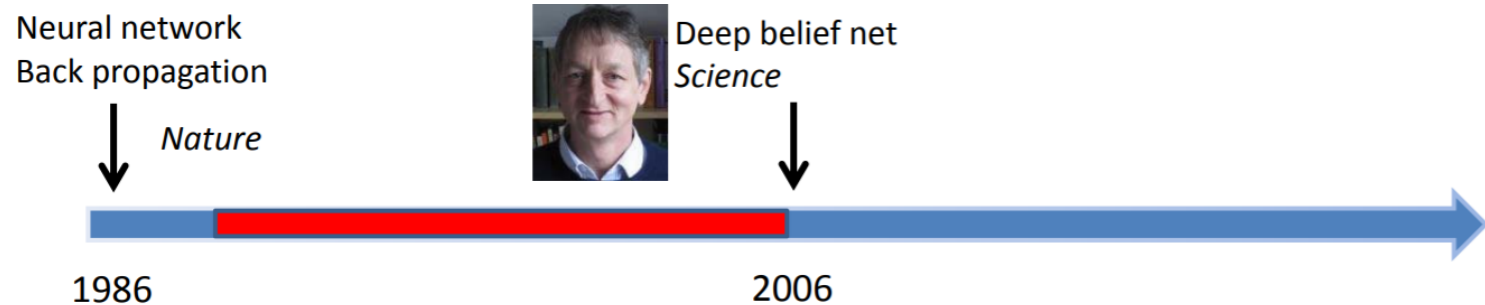


Flat Processing Scheme



Kruger et al. TPAMI'13

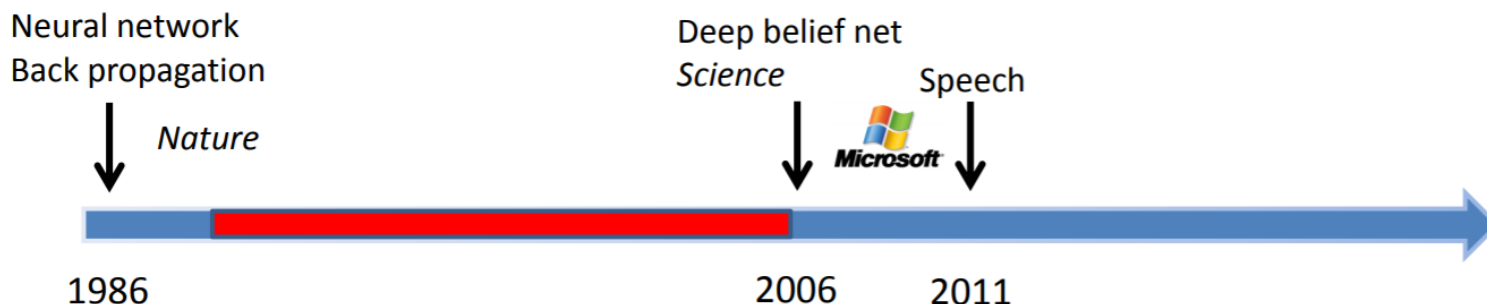
Introduction of deep learning



- Unsupervised & Layer-wised pre-training
- Better designs for modeling and training (normalization, nonlinearity, dropout)
- New development of computer architectures
 - GPU
 - Multi-core computer systems
- Large scale databases

Big Data !

Introduction of deep learning



deep learning results

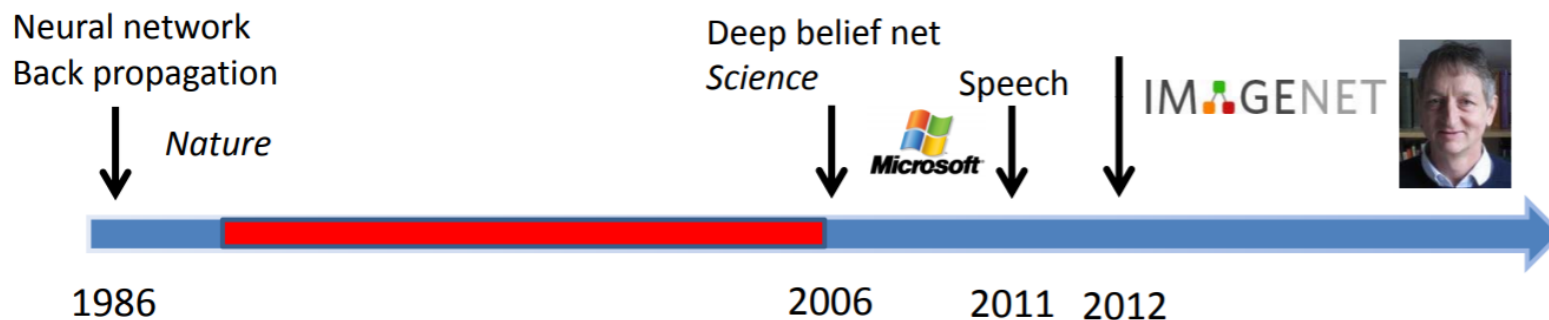
task	hours of training data	DNN-HMM	GMM-HMM with same data
Switchboard (test set 1)	309	18.5	27.4
Switchboard (test set 2)	309	16.1	23.6
English Broadcast News	50	17.5	18.8
Bing Voice Search (Sentence error rates)	24	30.4	36.2
Google Voice Input	5,870	12.3	
Youtube	1,400	47.6	52.3

Deep Networks Advance State of Art in Speech

Deep Learning leads to breakthrough in speech recognition at MSR.



Introduction of deep learning



Rank	Name	Error rate	Description
1	U. Toronto	0.15315	Deep learning
2	U. Tokyo	0.26172	Hand-crafted features and learning models. Bottleneck.
3	U. Oxford	0.26979	
4	Xerox/INRIA	0.27058	

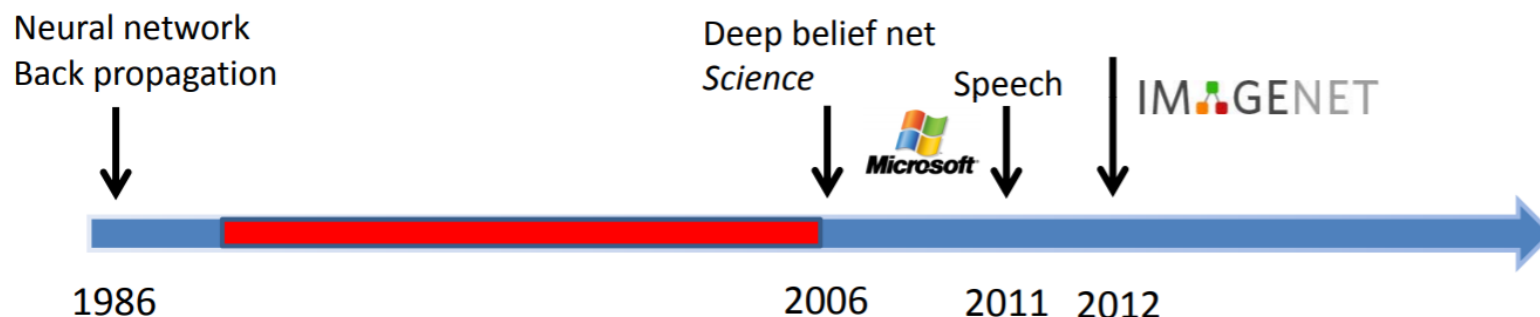
Object recognition over 1,000,000 images and 1,000 categories (2 GPU)

Introduction of deep learning

- Examples from ImageNet (<http://www.image-net.org/about-stats>)



Introduction of deep learning



- ImageNet 2013 – image classification challenge

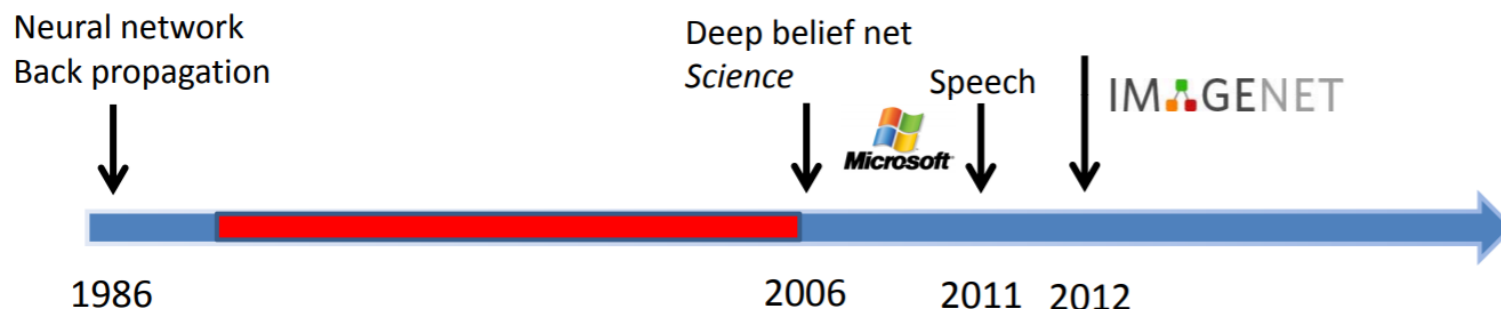
Rank	Name	Error rate	Description
1	NYU	0.11197	Deep learning
2	NUS	0.12535	Deep learning
3	Oxford	0.13555	Deep learning

MSRA, IBM, Adobe, NEC, Clarifai, Berkley, U. Tokyo, UCLA, UIUC, Toronto Top 20 groups all used deep learning

- ImageNet 2013 – object detection challenge

Rank	Name	Mean Average Precision	Description
1	UvA-Euvision	0.22581	Hand-crafted features
2	NEC-MU	0.20895	Hand-crafted features
3	NYU	0.19400	Deep learning

Introduction of deep learning



- ImageNet 2014 – Image classification challenge

Rank	Name	Error rate	Description
1	Google	0.06656	Deep learning
2	Oxford	0.07325	Deep learning
3	MSRA	0.08062	Deep learning

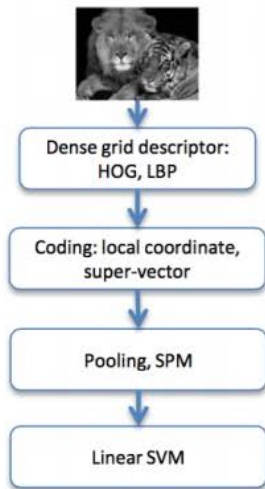
- ImageNet 2014 – object detection challenge

Rank	Name	Mean Average Precision	Description
1	Google	0.43933	Deep learning
2	CUHK	0.40656	Deep learning
3	DeepInsight	0.40452	Deep learning
4	UvA-Euvision	0.35421	Deep learning
5	Berkley Vision	0.34521	Deep learning

Deep networks for ImageNet

Year 2010

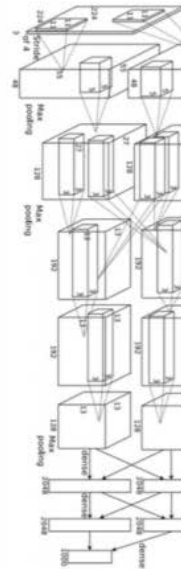
NEC-UIUC



[Lin CVPR 2011]

Year 2012

AlexNet



[Krizhevsky NIPS 2012]

Year 2014

GoogLeNet



[Szegedy arxiv 2014]

VGG



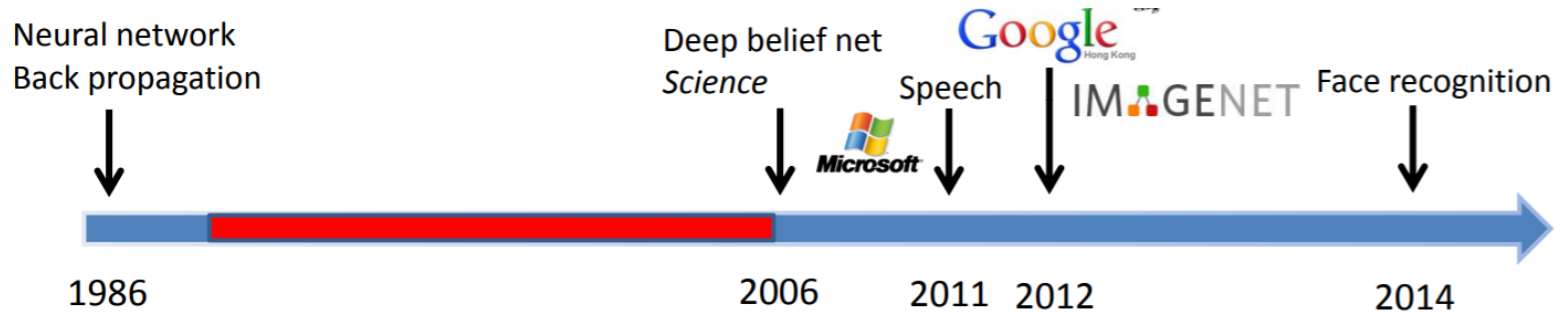
[Simonyan arxiv 2014]

Year 2015

MSRA ResNet



Introduction of deep learning



- Deep learning achieves 99.15% face verification accuracy on Labeled Faces in the Wild (LFW), close to human performance

Y. Sun, X. Wang, and X. Tang. Deep Learning Face Representation by Joint Identification-Verification. NIPS, 2014.

Labeled Faces in the Wild (2007)



Random guess (50%)
Eigenface (60%)

Best results without deep learning

TL Joint Bayesian (96.33%), 2013
Human cropped (97.53%)

Our deep learning result (99.15%)
Human funneled (99.20%)



Introduction of deep learning

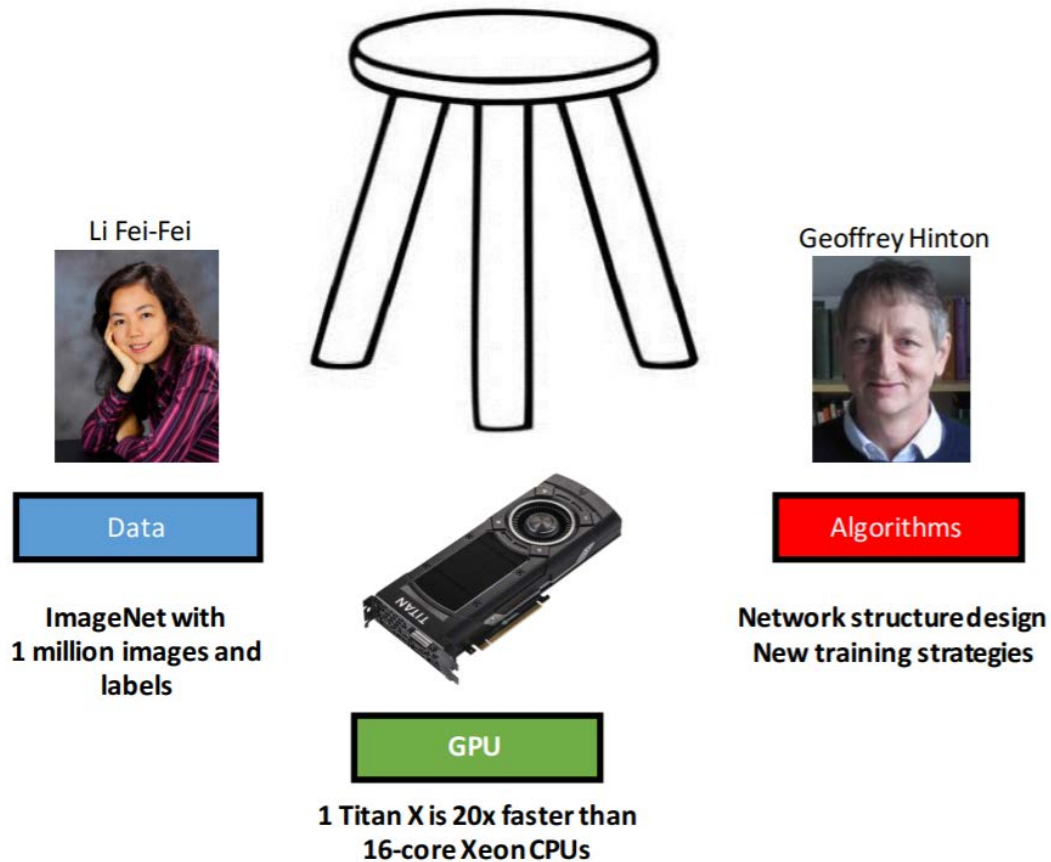
■ 10 breakthrough technologies 2013

Deep Learning With massive amounts of computational power, machines can now recognize objects and translate speech in real time. Artificial intelligence is finally getting smart. →	Temporary Social Media Messages that quickly self-destruct could enhance the privacy of online communications and make people freer to be spontaneous. →	Prenatal DNA Sequencing Reading the DNA of fetuses will be the next frontier of the genomic revolution. But do you really want to know about the genetic problems or musical aptitude of your unborn child? →	Additive Manufacturing Skeptical about 3-D printing? GE, the world's largest manufacturer, is on the verge of using the technology to make jet parts. →	Baxter: The Blue-Collar Robot Rodney Brooks's newest creation is easy to interact with, but the complex innovations behind the robot show just how hard it is to get along with people. →
Memory Implants A maverick neuroscientist believes he has deciphered the code by which the brain forms long-term memories. Next: testing a prosthetic implant for people suffering from long-term memory loss. →	Smart Watches The designers of the Pebble watch realized that a mobile phone is more useful if you don't have to take it out of your pocket. →	Ultra-Efficient Solar Power Doubling the efficiency of a solar cell would completely change the economics of renewable energy. Nanotechnology just might make it possible. →	Big Data from Cheap Phones Collecting and analyzing information from simple cell phones can provide surprising insights into how people move about and behave – and even help us understand the spread of diseases. →	Supergrids A new high-power circuit breaker could finally make highly efficient DC power grids practical. →

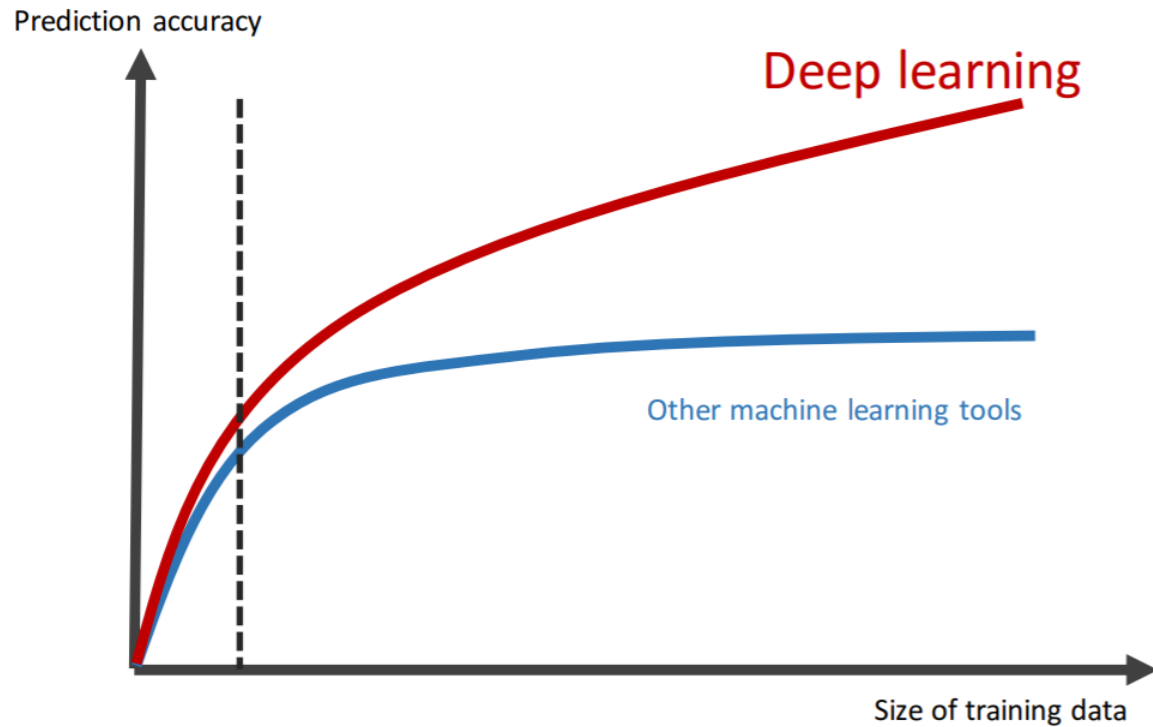
What leads to the breakthrough?

- So, why indeed, did purely supervised learning with backpropagation not work well in the past?
- Geoffrey Hinton summarized the findings up to today in these four points:
 - 1. Our labeled datasets were thousands of times too small.
 - 2. Our computers were millions of times too slow.
 - 3. We initialized the weights in a stupid way.
 - 4. We used the wrong type of non-linearity

What leads to the breakthrough?



Some observations



Outline

- Introduction of deep learning
- DeepFace
- FaceNet

DeepFace (Taigman and Wolf)

- 2D/3D face modeling and alignment using affine transformations
- 9 layer deep neural network
- 120 million parameters



DeepFace

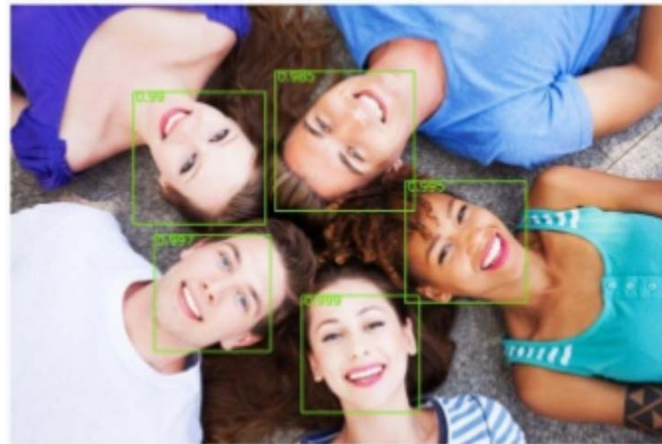
- Five stages
 - **Detect** the face in the image
 - **Align** the detected face
 - Design **deep learning** models
 - **Represent** facial images through a large deep neural network
 - **Classify** the image comparing a feature vector



DeepFace

■ Detection

- The detected face is cropped and used as input for alignment
- DeepFace algorithm uses already developed Face detection algorithm



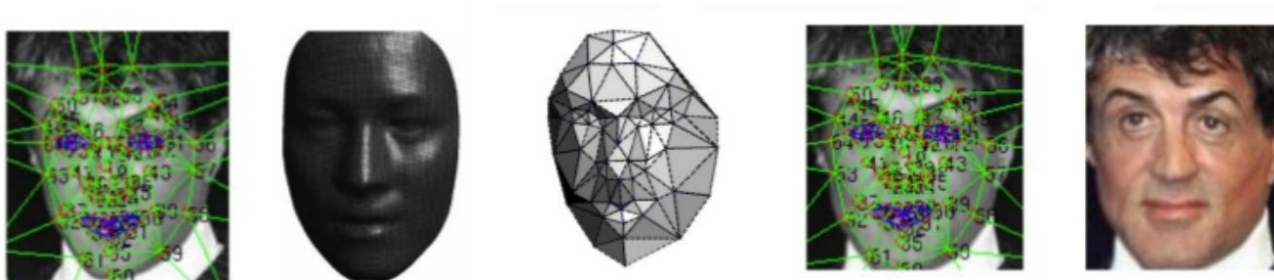
DeepFace

- 2D alignment
 - Detect 6 fiducial points to scale, rotate and translate the image
 - The fiducial points are detected using support vector regressor



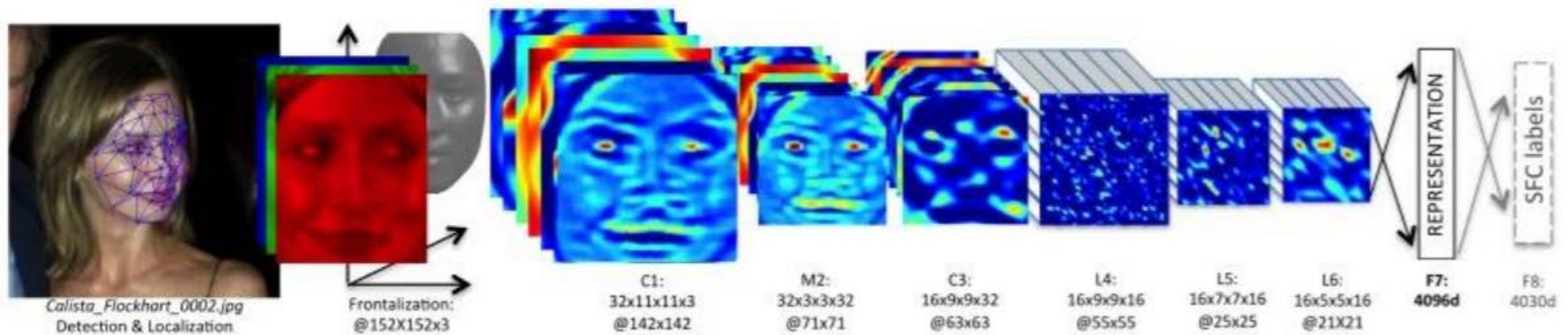
DeepFace

- 3D alignment
 - Detect 67 fiducial points
 - Compare with them with a fixed 3D model of a generic face
 - Reconstruct the 3d model of the detected face
 - Generate the 3d aligned version of the crop
 - The final image is given to the DNN



DeepFace - Deep Learning

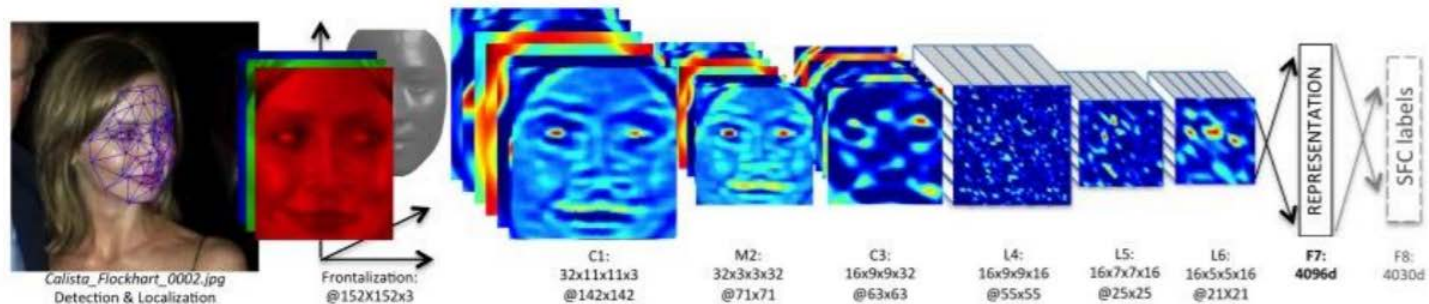
- Deep learning
 - Input: 3D aligned 3 channel (RGB) face image 152x152 pixels



DeepFace - Deep Learning

■ Layer 1-3 : Intuition

- Convolution layers - extract low-level features (e.g. simple edges and texture)
- ReLU after each conv. Layer: reduce vanishing gradient issues
- Max-pooling: make convolution network more robust to local translations.

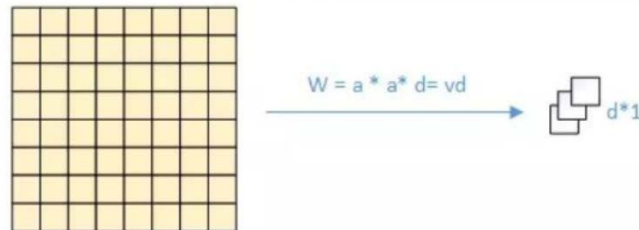


DeepFace - Deep Learning

■ Layer 4-6: Intuition

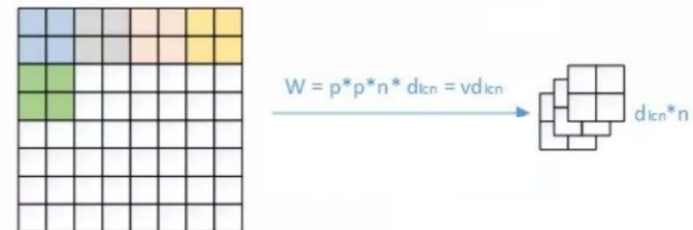
- Locally Connected Layers
- Apply filters to different locations on the map
- Similar to a conv. layer but spatially dependent

Fully-Connected layer (DNN block)



Locally-Connected layer (LCN block)

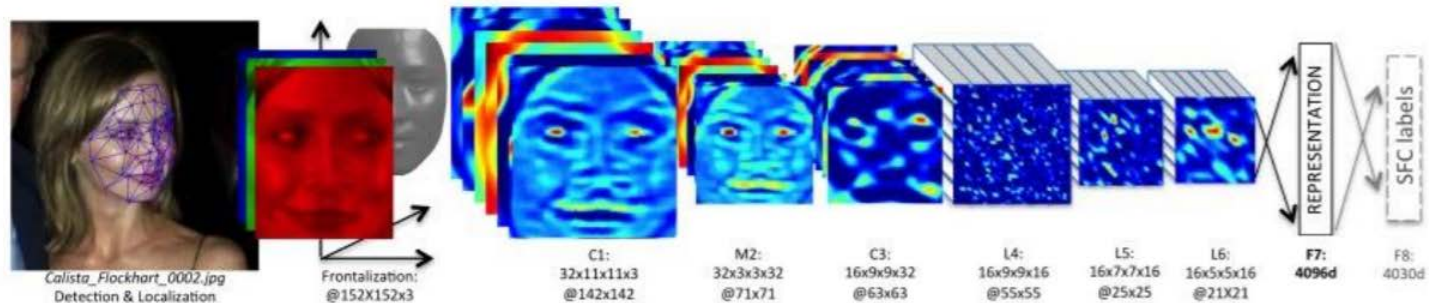
假设 $p = 2$



DeepFace - Deep Learning

■ Layer 7: Intuition

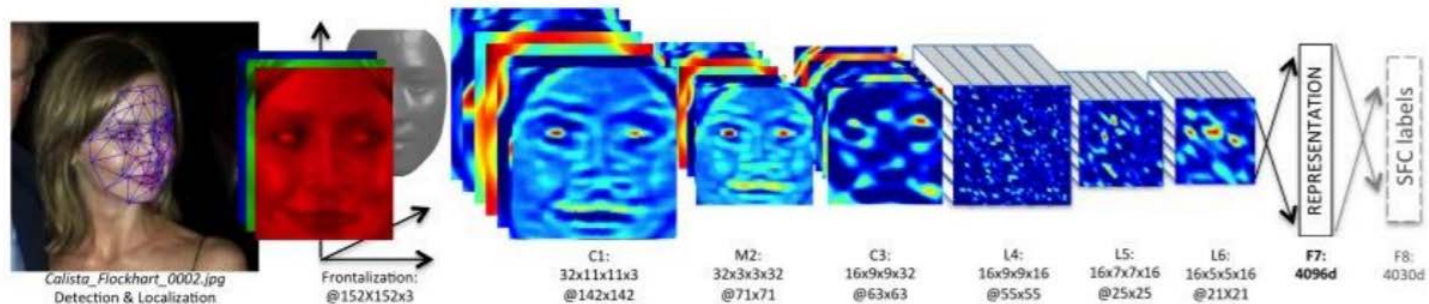
- Layer F7 is fully connected and generates 4096d vector
- Sparse representation of face descriptor
- 75% of outputs are zero



DeepFace - Deep Learning

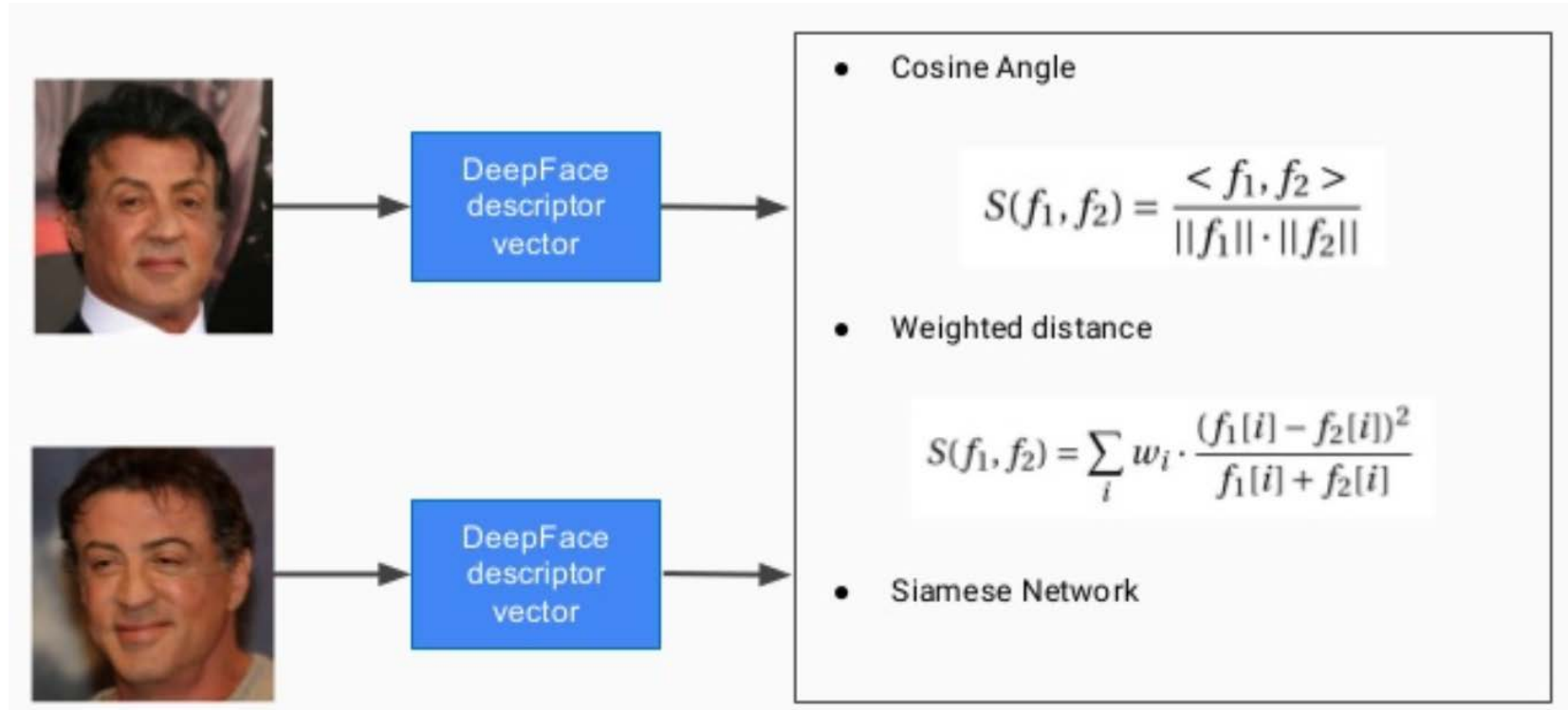
■ Layer 8: Intuition

- Layer F8 is fully connected and generates 4030d vector, acts as K class SVM
- F8 calculates probability with softmax $p_k = \exp(o_k) / \sum_h \exp(o_h)$
- Cross-entropy loss function: $L = -\sum_k \log(p_k)$
- Computed using Stochastic Gradient Descent (SGD) and performed backpropagation



DeepFace

■ Classification

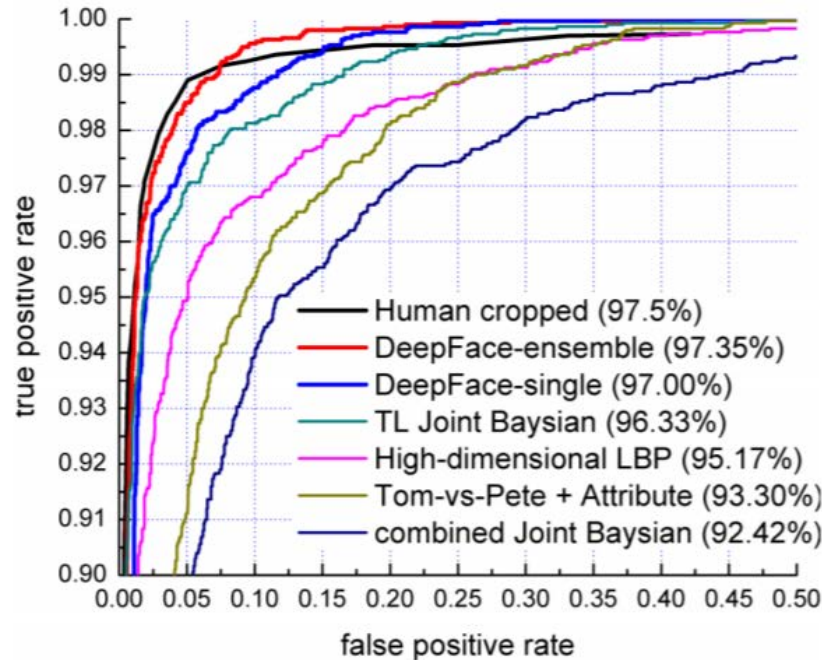


DeepFace – Training

- Trained on 4M faces (4030 identities, 800-1200 images per person)
- We will focus on Labeled Faces in the Wild (LFW) evaluation
- Used SGD with momentum of 0.9
- Learning rate 0.01 with manual decreasing, final rate was 0.0001
- Random weight initializing
- 15 epochs of training
- 3 days total on a GPU-based engine

DeepFace – results

- DeepFace was evaluated on LFW dataset
- 13233 images collected from the web
- 1680 identities.
- 0.9735 ± 0.0025 accuracy



DeepFace – Summary

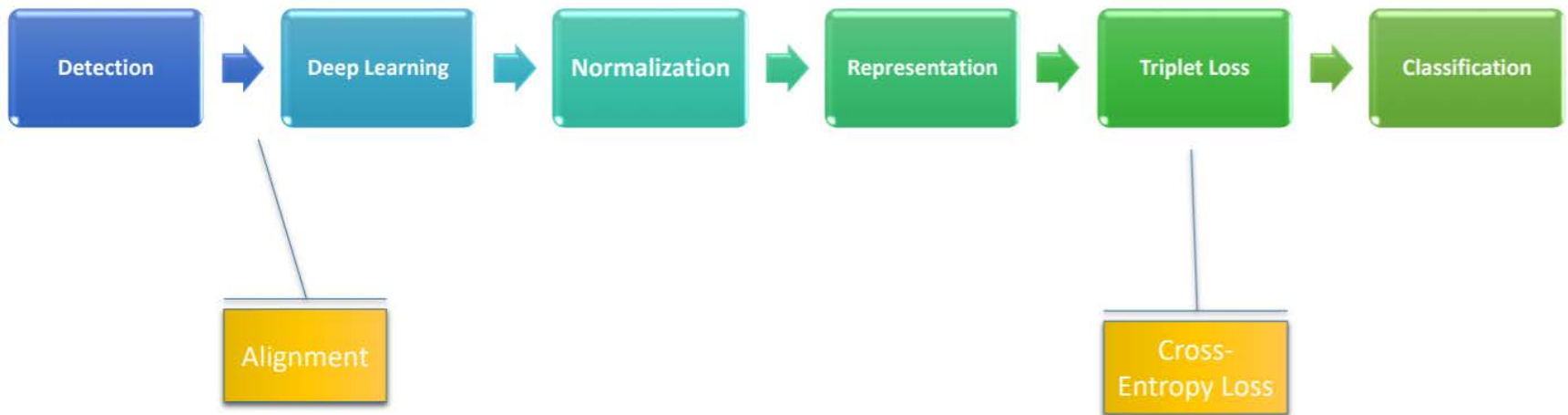
- Close to human accuracy
- 120M parameters
- Proves that going deeper brings better results
- Computation efficiency – 0.33 second per face image @2.2GHz CPU
- Invariant to pose, illumination, expression and image quality

Outline

- Introduction of deep learning
- DeepFace
- FaceNet

FaceNet (Schroff and Philbin)

■ Workflow



FaceNet

- Deep CNN (22 layers)
- Works on pure data
- Embedding (State-Of-The-Art face recognition using only 128 features per face → efficient!)
- Triplet images for training and loss function
- Uses SGD, Dropout, ReLU



FaceNet - Deep Learning

- 22 layers:
 - 11 convolutions
 - 2 normalizations (reduce internal covariance shift, stabilize the learning process and dramatically reduce the number of training epochs required to train deep networks)
 - 4 max-pooling
 - 1 concatenation
 - 3 fully-connected
- 140 million parameters

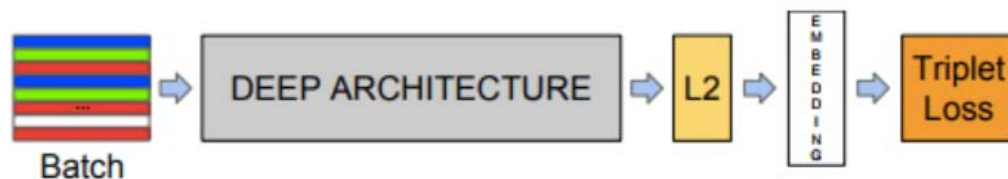
layer	size-in	size-out	kernel	param	FLPS
conv1	220×220×3	110×110×64	7×7×3, 2	9K	115M
pool1	110×110×64	55×55×64	3×3×64, 2	0	
rnorm1	55×55×64	55×55×64		0	
conv2a	55×55×64	55×55×64	1×1×64, 1	4K	13M
conv2	55×55×64	55×55×192	3×3×64, 1	111K	335M
rnorm2	55×55×192	55×55×192		0	
pool2	55×55×192	28×28×192	3×3×192, 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
fc7/128	1×32×128	1×1×128		524K	0.5M
L2	1×1×128	1×1×128		0	
total				140M	1.6B



FaceNet - Representation

■ Representation:

- Feature will go through the L2 normalization
- Embedding concept: Transform an image to a low dimensional feature space (128 d)

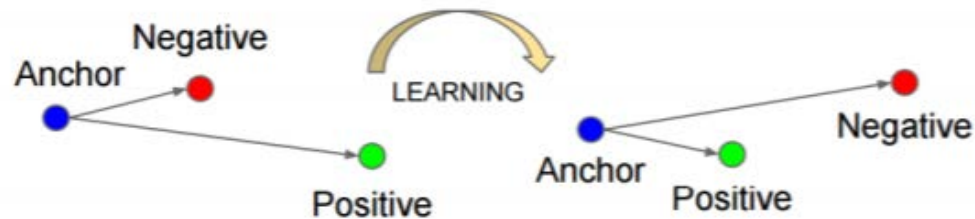


FaceNet – Triplet Loss

- Crucial to ensure fast convergence
- Select triplets that satisfy the triplet constraint
 - Anchor and Positive $\rightarrow$ same identity
 - Negative $\rightarrow$ different identity

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

α : Margin between positive and negative pairs



FaceNet – Triplet Loss

- Loss function

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

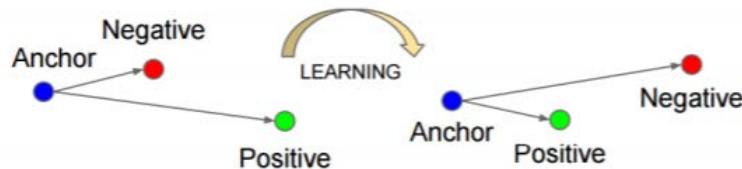
- However, generating all possible triplets would result in many computation
- **Offline generation**: every n steps, using the most recent network checkpoint and computing the argmin and armax on a subset of the data
- **Online generation**: done by selecting positive or negative examples from within a mini-batch



FaceNet – Triplet Loss

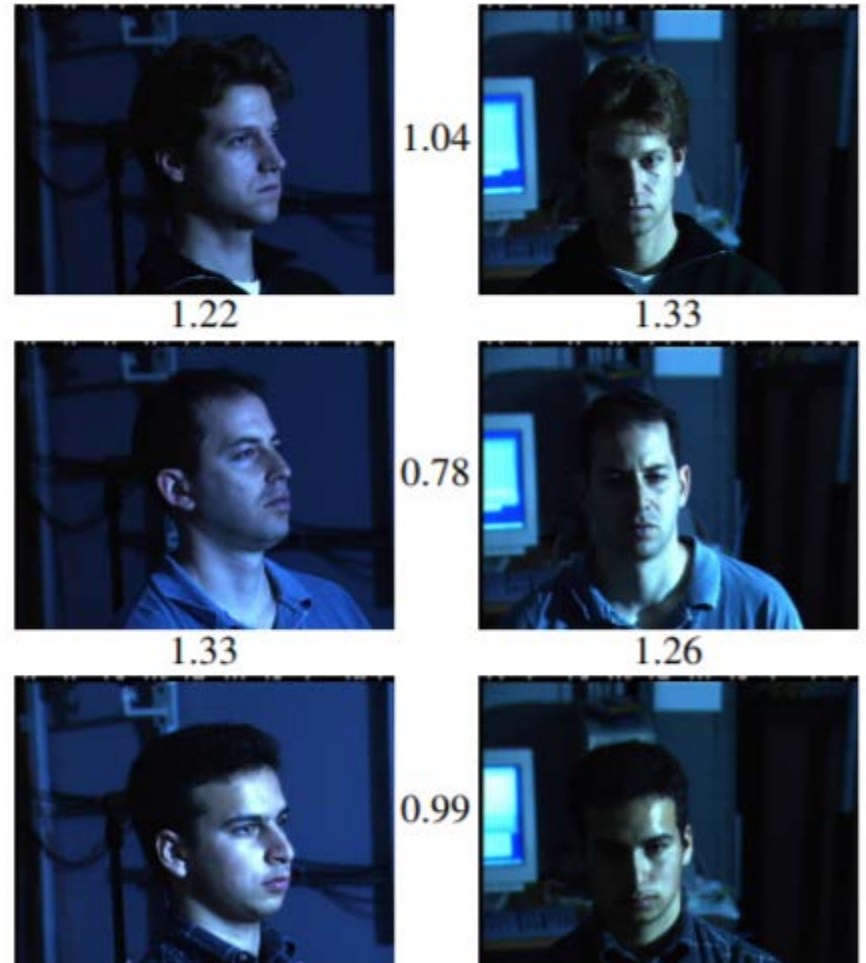
- Compute $\arg \max_{x_i^p} \|f(x_i^a) - f(x_i^p)\|_2^2, \arg \min_{x_i^n} \|f(x_i^a) - f(x_i^n)\|_2^2$
- Better: Choose all anchor-positive pairs in a mini-batch while selecting only hard-negatives
- To avoid local minima, they chose negative semi-hard exemplars that satisfy

$$\|f(x_i^a) - f(x_i^p)\|_2^2 < \|f(x_i^a) - f(x_i^n)\|_2^2$$



FaceNet – Triplet Loss

- Illumination and Pose invariance.
 - A distance of 0.0 means the faces are identical, 4.0 corresponds to the opposite spectrum, two different identities.
 - You can see that a threshold of 1.1 would classify every pair correctly



FaceNet – Classification

- LFW (Labeled Faces in the Wild) dataset: 13233 images collected from the web, 1680 identities.
- Results: 0.9887 ± 0.15 accuracy
with face alignment 0.9963 ± 0.09
(DeepFace: 0.9735 ± 0.0025 accuracy)



FaceNet – Summary

- Important new concepts: Triplet loss and Embeddings
- 140M parameters
- Proves that going deeper brings better results for the face recognition problem
- Computation efficiency ~0.73 second per face image (1.6B FLOPS) @2.2GHZ CPU
- Invariant to pose, illumination, expression and image quality



Comparison

DeepFace	FaceNet
Multi-class probability	Embedding
9 layers	22 layers
120M parameters	140M parameters
0.33 sec per image @2.2GHZ CPU	~0.73 sec per image @2.2GHZ CPU
2D/3D alignment	Crop and scaling
Cross-Entropy Loss	Triplet loss
0.9735±0.0025	0.9963±0.09

