Face Recognition
A Deep Learning Approach

Lihi Shiloh

Tal Perl

Deep Learning Seminar
School of Electrical Engineer – Tel Aviv University

Outline

- Classical face recognition
- Modern face recognition
- DeepFace
- FaceNet
- Comparison
- Discussion

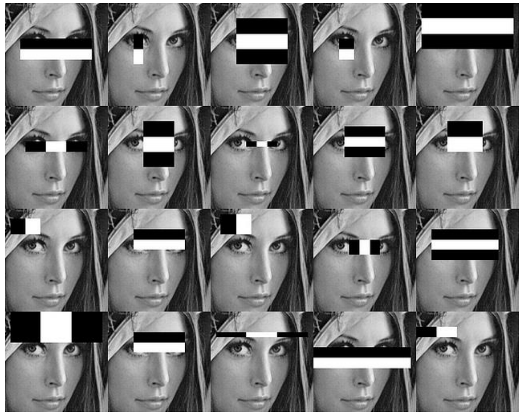
Articles

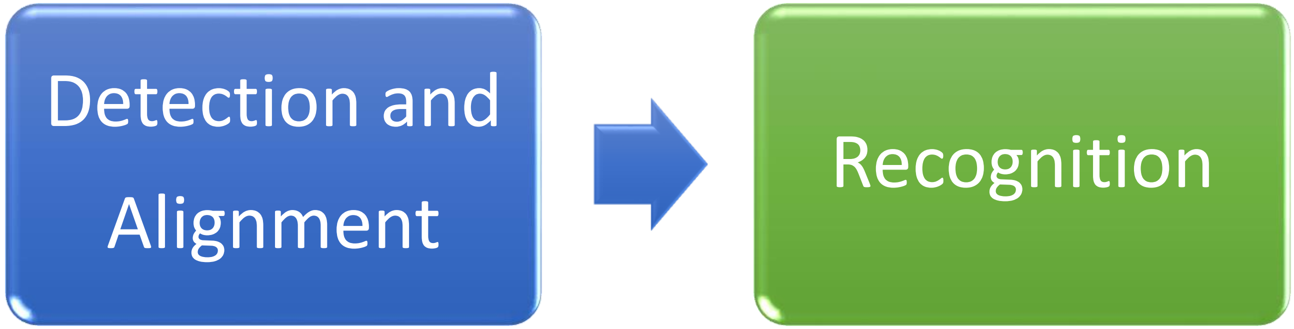
- DeepFace: Closing the Gap to Human-Level Performance in Face Verification
  - Link - https://www.cs.toronto.edu/~ranzato/publications/taigman_cvpr14.pdf
- FaceNet: A Unified Embedding for Face Recognition and Clustering
  - Link - https://arxiv.org/pdf/1503.03832v3.pdf

The Face Recognition Problem

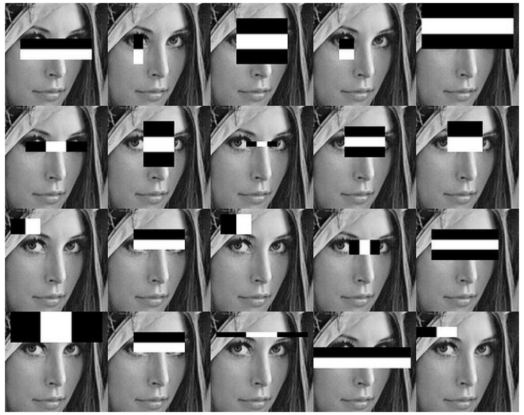
- Matrix of pixels
- 512x512=262144
- Need to find a face
- Compare to a database
- Curse of dimensionality

Face recognition pipeline (Classical)

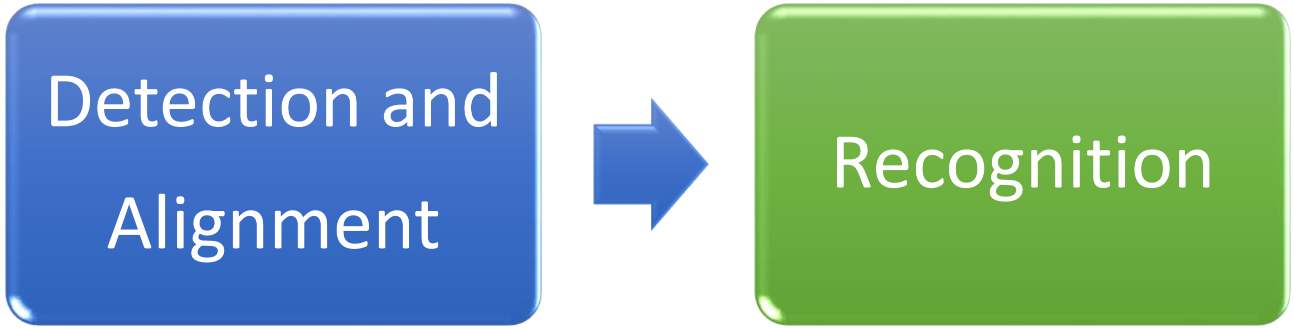




Face recognition pipeline (Classical)



- Features → handpicked
- Works well on small datasets
- Fails on illumination variations and facial expressions
- Fails on large datasets



Face patterns lie on a complex nonlinear and non-convex manifold in the high-dimensional space.

Ok… So how can we solve this nonlinear, nonconvex problem?

Modern Face Recognition

- More Data attained by crawling – more faces!
- Google / Facebook etc.
- Stronger Hardware → More powerful statistical models
- Neural networks **/ Deep Learning**

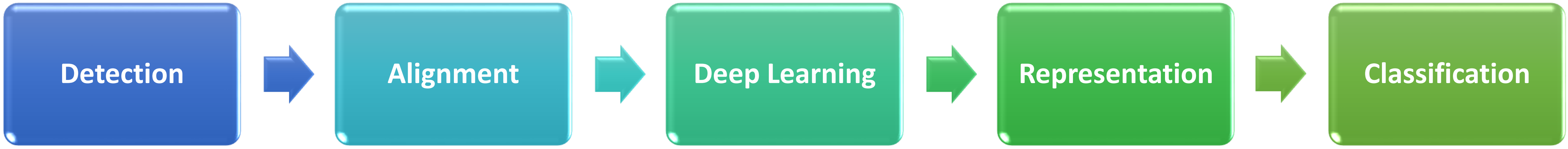
The Network will find the features itself…

Lets dive into 2 examples

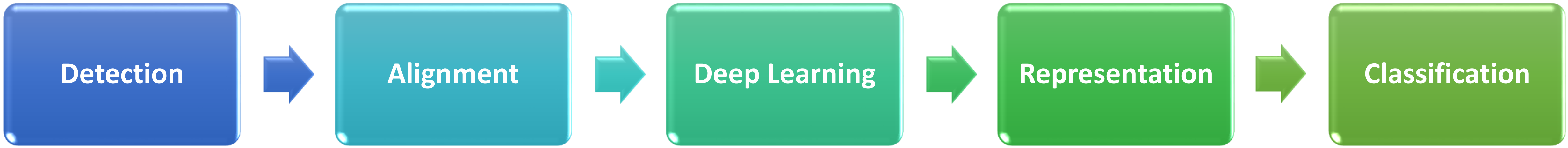
DeepFace (2014)

FaceNet (2015)

DeepFace (Taigman and Wolf 2014)



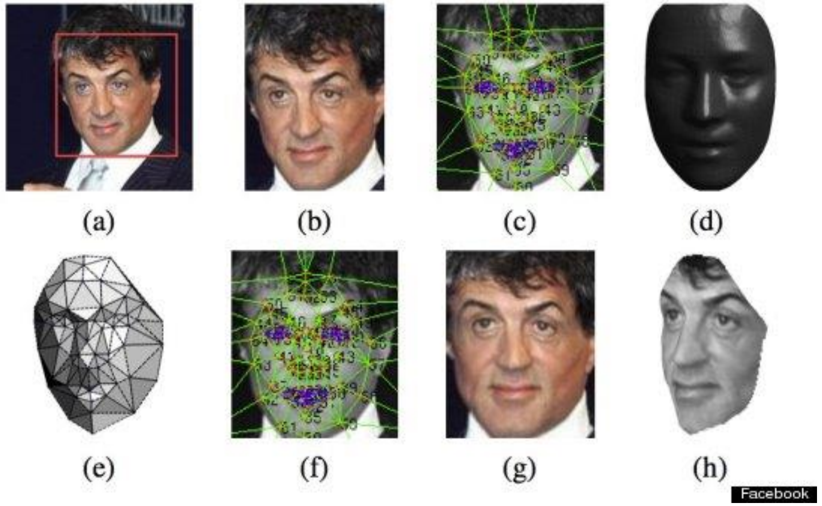
DeepFace (Taigman and Wolf 2014)



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

DeepFace - Alignment (*Frontalization*)

- Fiducial points (face landmarks)
- 2D and 3D affine transformations
- Frontal face view



Detection → Alignment → Deep Learning → Representation → Classification

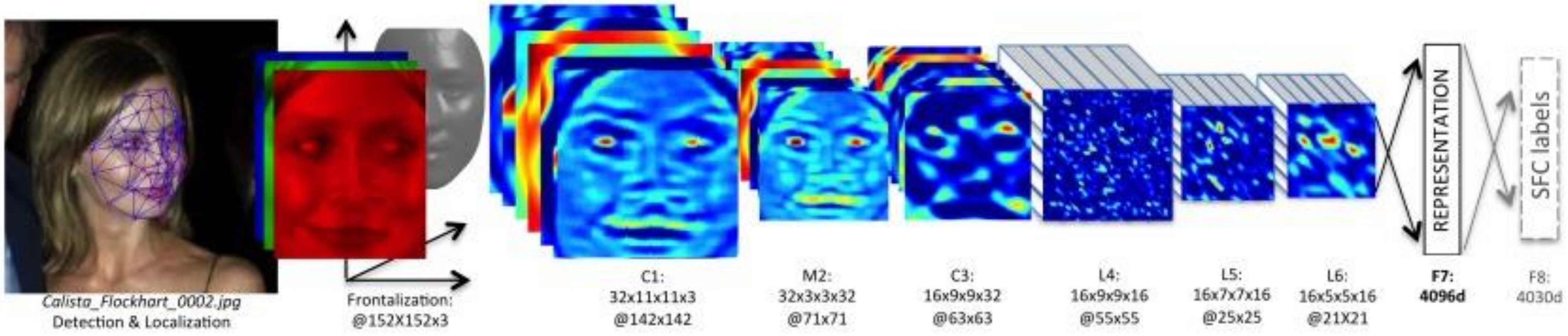
DeepFace - Deep Learning

- Input: 3D aligned 3 channel (RGB) face image 152x152 pixels
- 9 layer deep neural network architecture
- Performs soft max for minimizing cross entropy loss
- Uses SGD, Dropout, ReLU
- Outputs k-Class prediction

Detection → Alignment → Deep Learning → Representation → Classification

DeepFace - Deep Learning

- Architecture

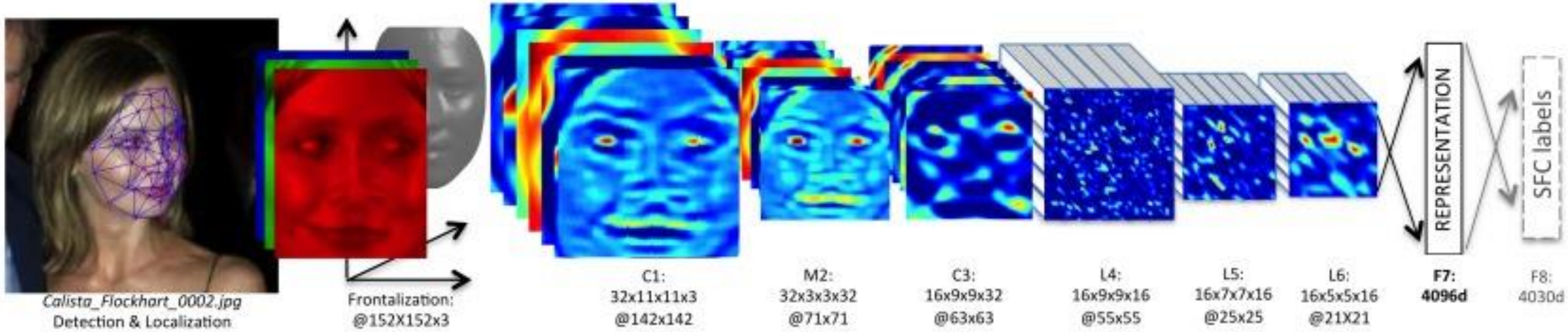


Detection → Alignment → **Deep Learning** → Representation → Classification

DeepFace - Deep Learning

Layer 1-3 : Intuition

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

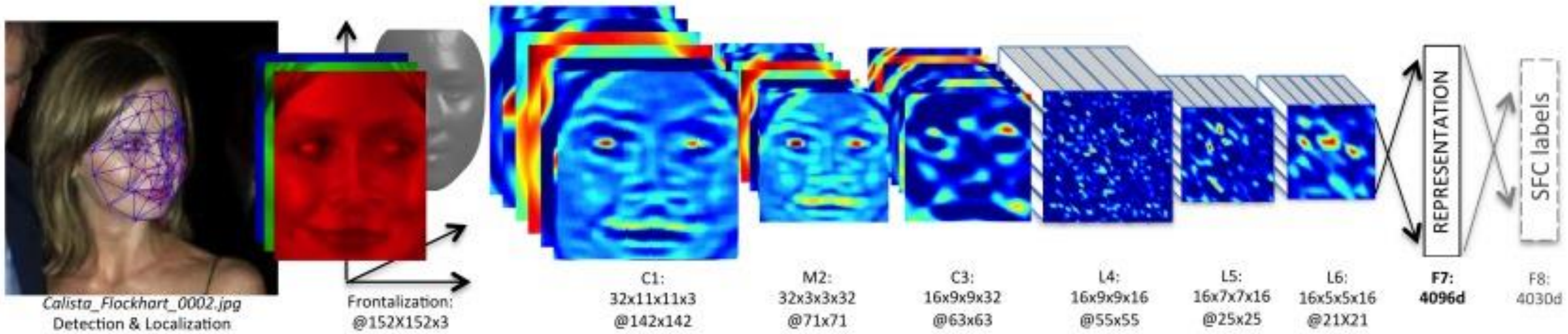


Detection → Alignment → **Deep Learning** → Representation → Classification

DeepFace - Deep Learning

Layer 4-6: Intuition

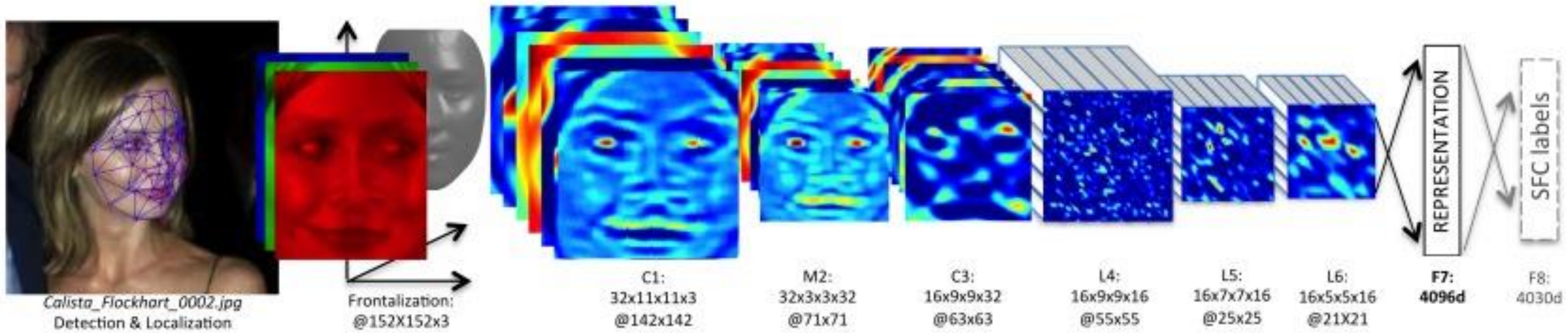
- Apply filters to different locations on the map
- Similar to a conv. layer but spatially dependent



Detection → Alignment → **Deep Learning** → Representation → Classification

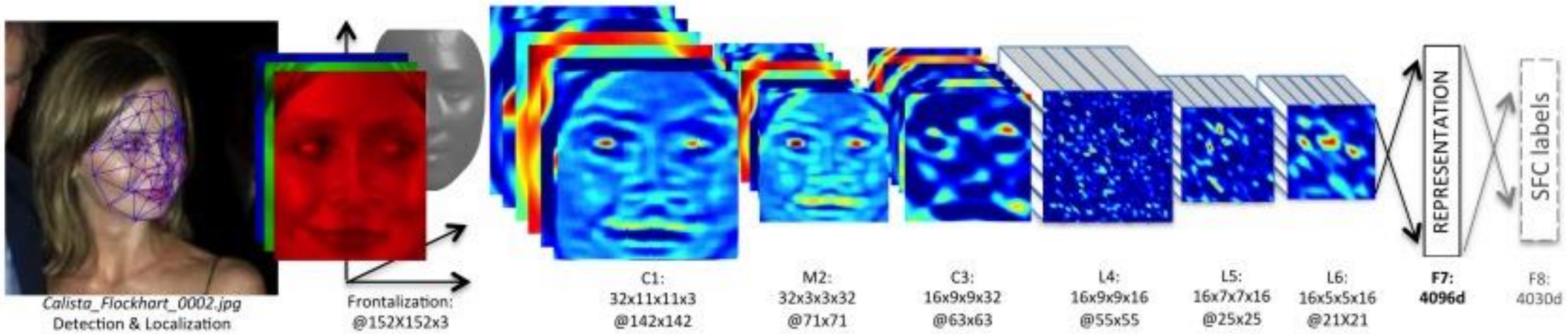
DeepFace - Deep Learning

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



DeepFace - Deep Learning

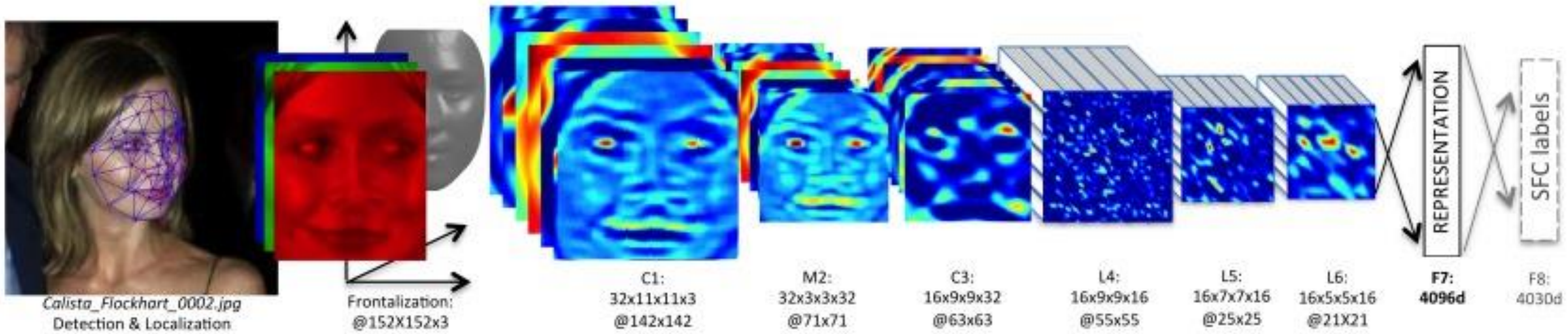
- Layer F8 is fully connected and generates 4030d vector
- Acts as K class SVM



Detection → Alignment → Deep Learning → Representation → Classification

DeepFace – Representation

- 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 SGD and performs backpropagation



Detection → Alignment → Deep Learning → Representation → Classification

DeepFace – Training

- Trained on SFC 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

- DF was evaluated on LFW (Labeled Faces in the Wild) dataset
  - 13233 images collected from the web
  - 1680 identities.

- 0.9735±0.0025 accuracy

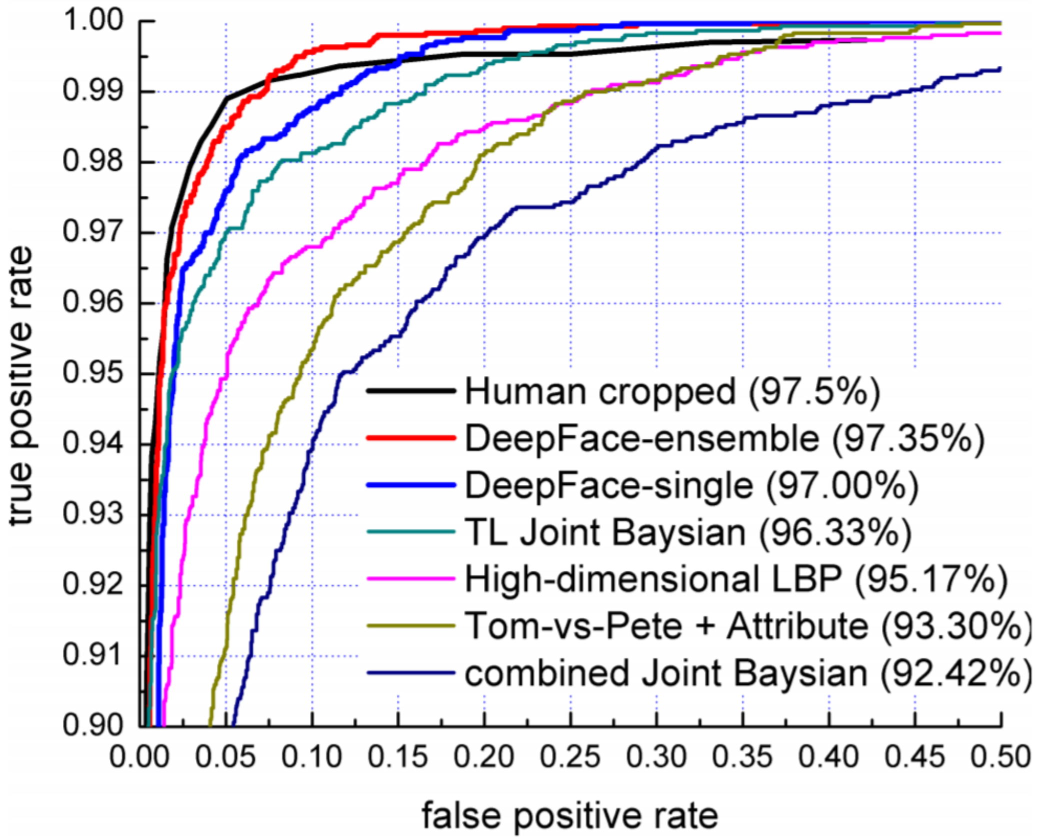


Figure 3. The ROC curves on the *LFW* dataset. Best viewed in color.

DeepFace – results

- Experimented with different derivatives of deep architecture
- Single – the network we discussed
- Ensemble – a combination of 3 networks with different inputs
  - Final K class computer by:

$$K_{\text{combined}} = K_{\text{single}} + K_{\text{gradient}} + K_{\text{align2d}}$$

$$K\left(x, y\right) = -\left\|x - y\right\|_2$$

DeepFace-single	**0.9592** ±0.0029	unsupervised
DeepFace-single	**0.9700** ±0.0028	restricted
DeepFace-ensemble	**0.9715** ±0.0027	restricted
DeepFace-ensemble	**0.9735** ±0.0025	unrestricted
Human, cropped	0.9753	

DeepFace – deeper is better

- Experimented with different depths of networks
- Removed C3, L4, L5
- Compared error rate to number of classes K

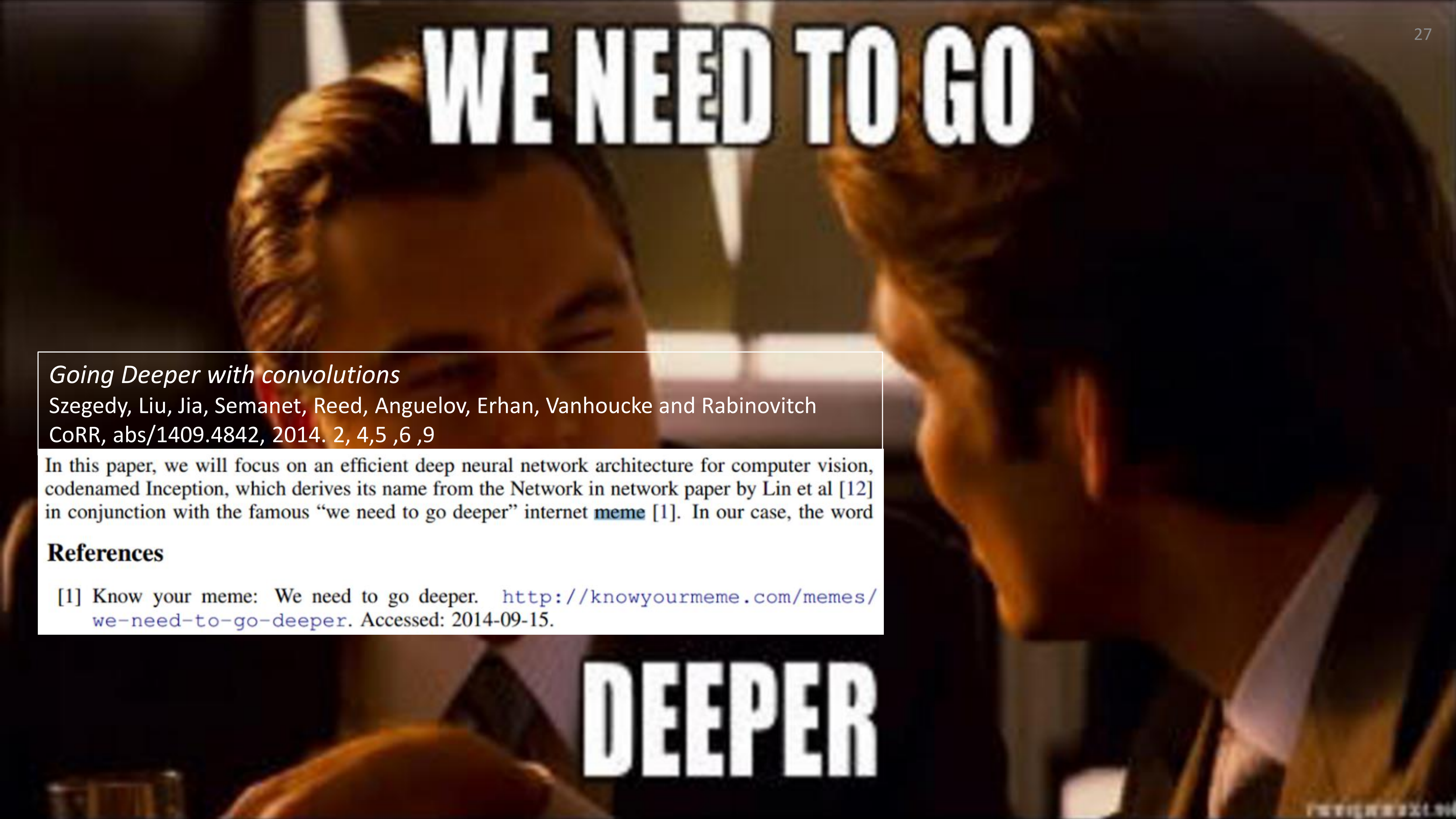
Network	Error
DF-1.5K	7.00%
DF-3.3K	7.22%
DF-4.4K	8.74%

Network	Error
DF-10%	20.7%
DF-20%	15.1%
DF-50%	10.9%

Network	Error
DF-sub1	11.2%
DF-sub2	12.6%
DF-sub3	13.5%

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
- Our work is done…



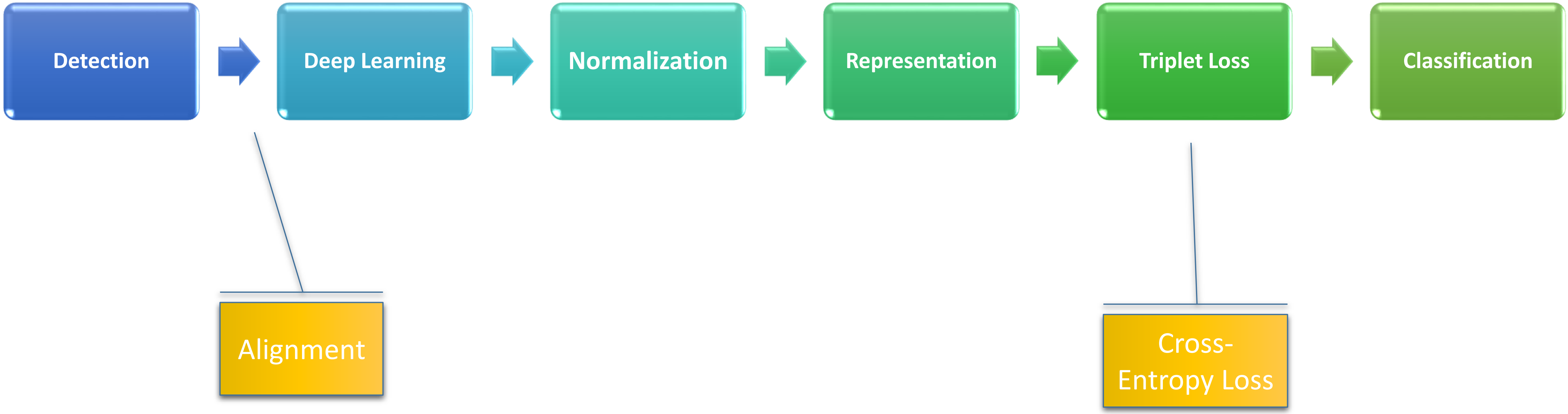
Going Deeper with convolutions
Szegedy, Liu, Jia, Semanet, Reed, Anguelov, Erhan, Vanhoucke and Rabinovitch
CoRR, abs/1409.4842, 2014. 2, 4,5 ,6 ,9

In this paper, we will focus on an efficient deep neural network architecture for computer vision, codenamed Inception, which derives its name from the Network in network paper by Lin et al [12] in conjunction with the famous "we need to go deeper" internet meme [1]. In our case, the word

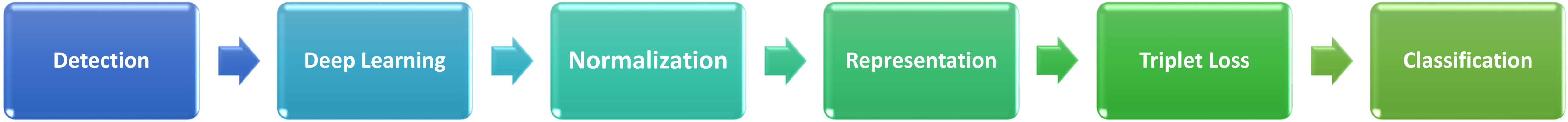
References

[1] Know your meme: We need to go deeper. http://knowyourmeme.com/memes/we-need-to-go-deeper. Accessed: 2014-09-15.

FaceNet (Schroff and Philbin 2015)



FaceNet (Schroff and Philbin 2015)



- 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

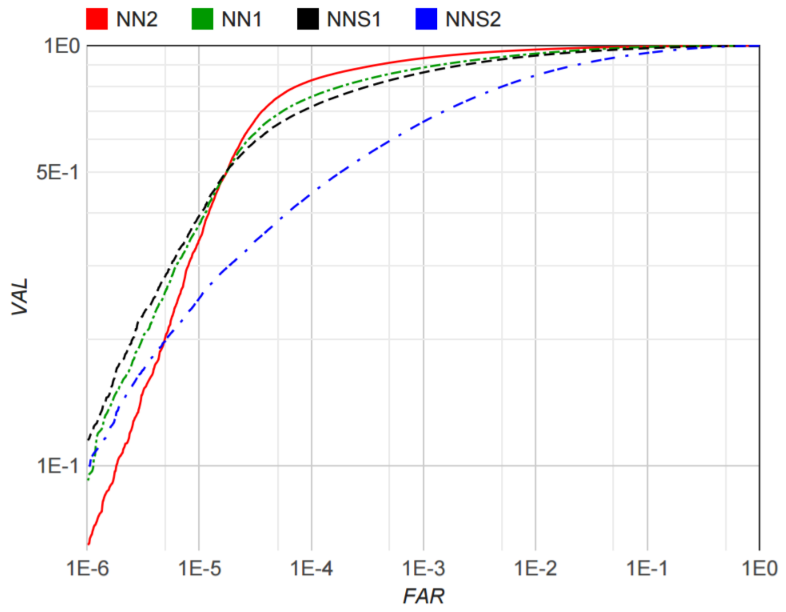


Figure 5. **Network Architectures.** This plot shows the complete ROC for the four different models on our personal photos test set from section 4.2. The sharp drop at 10E-4 FAR can be explained by noise in the groundtruth labels. The models in order of performance are: **NN2:** 224×224 input Inception based model; **NN1:** Zeiler&Fergus based network with 1×1 convolutions; **NNS1:** small Inception style model with only 220M FLOPS; **NNS2:** tiny Inception model with only 20M FLOPS.

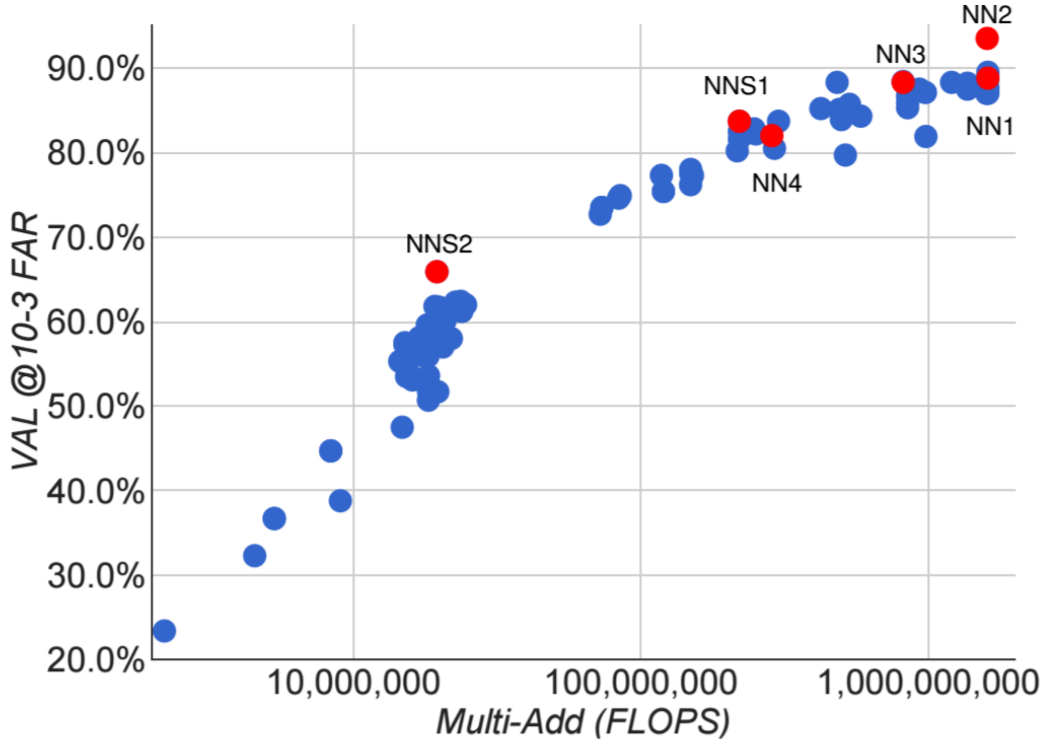


Figure 4. **FLOPS *vs.* Accuracy trade-off.** Shown is the trade-off between FLOPS and accuracy for a wide range of different model sizes and architectures. Highlighted are the four models that we focus on in our experiments.

Detection → Deep Learning → Normalization → Representation → Triplet Loss → Classification

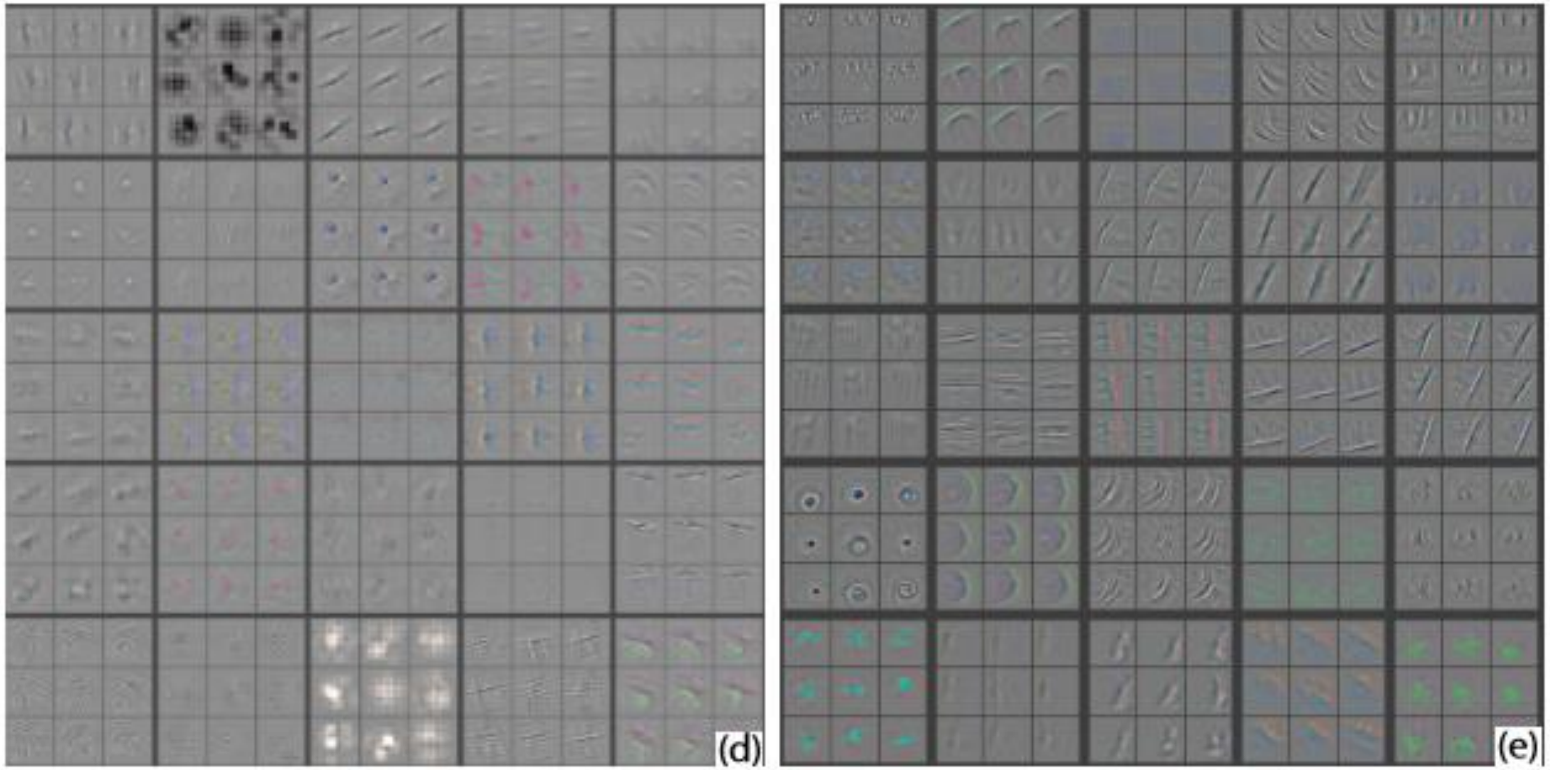
FaceNet – Deep Learning

- 22 layers:
  - ✓11 convolutions
  - ✓3 normalizations
  - ✓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
fc7128	1×32×128	1×1×128		524K	0.5M
L2	1×1×128	1×1×128		0	
total				140M	1.6B

Detection → **Deep Learning** → Normalization → Representation → Triplet Loss → Classification

FaceNet – Normalization

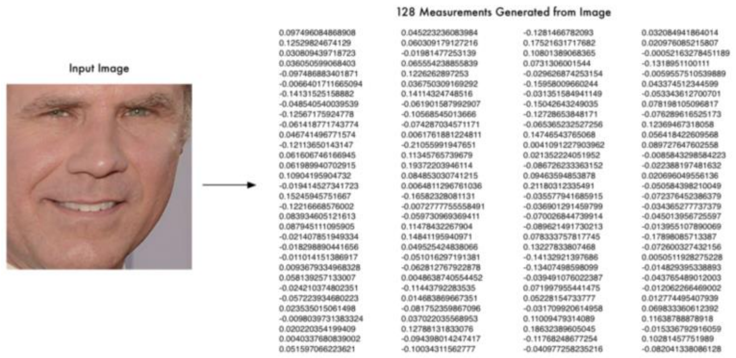


Visualizing and understanding convolutional networks
M. D. Zeiler and R. Fergus
CoRR, abs/1311.2901, 2013. 2, 3, 4, 6

FaceNet – Representation

- Objective is to minimize L2 distance between same face representations
- Embedding concept:

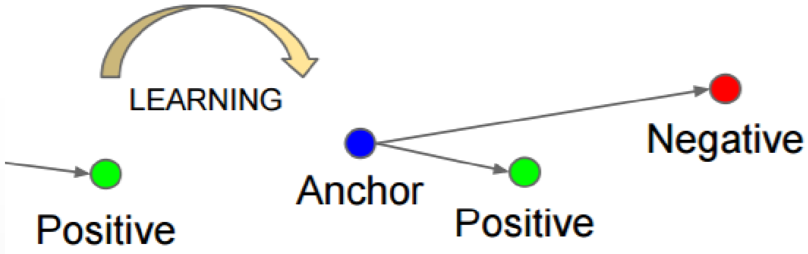
  Transform an image to a low dimensional feature space (128 d)
- Concept known for Natural Language Processing (NLP)



Detection → Deep Learning → Normalization → Representation → Triplet Loss → Classification

FaceNet – Triplet Loss

- Train model with triplets of roughly aligned matching / non-matching face patches
- T – set of all poss
- Loss function:

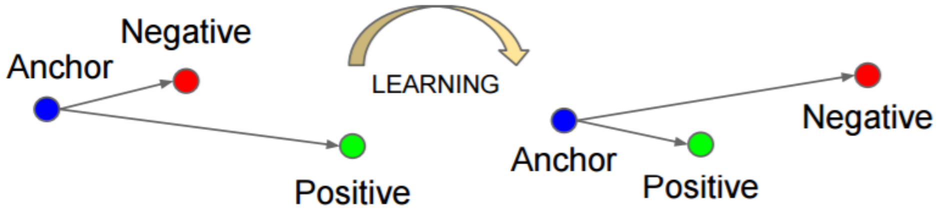


$$L = \sum_{i=1}^{N} \left\| f\left(x_i^a\right) - f\left(x_i^p \right.\right.$$

$$\forall \left(x_i^a, x_i^p, x_i^n\right) \in \mathrm{T}, f\left(x\right) \in \mathbb{R}^d$$

- Constraint : $\left\| f\left(x\right) \right\|_2 = 1$

FaceNet – Triplet Selection



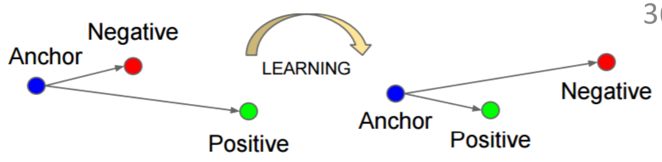
- Crucial to ensure fast convergence
- Select triplets that violet the triplet constraint:

$$\left\| f\left(x_i^a\right) - f\left(x_i^p\right) \right\|_2^2 + \alpha < \left\| f\left(x_i^a\right) - f\left(x_i^n\right) \right\|_2^2 \qquad \forall \left(x_i^a, x_i^p, x_i^n\right) \in \mathrm{T}$$

Offline
Generate triplets every n steps, using the most recent network checkpoint and computing argmin/argmax on a subset of the data

Online
Selecting hard positive/negative examplers from a mini-batch

Detection → Deep Learning → Normalization → Representation → **Triplet Loss** → Classification

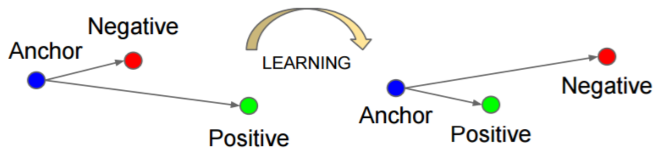


FaceNet – Online Triplet Selection

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

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

Detection → Deep Learning → Normalization → Representation → **Triplet Loss** → Classification

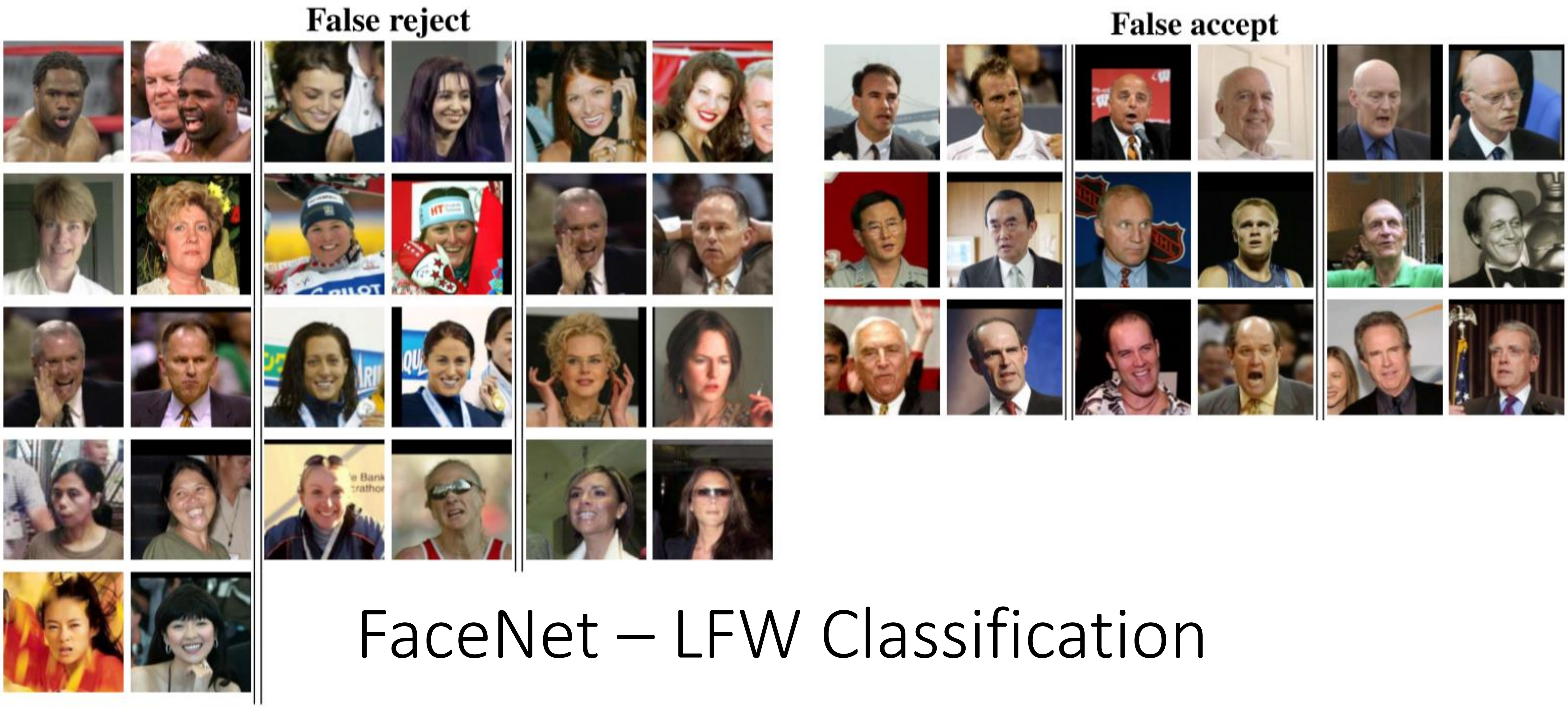


FaceNet – Classification

- Clustering: K-means or other clustering algorithms
- LFW (Labeled Faces in the Wild) dataset: 13233 images collected from the web, 1680 identities.
- Results: 0.9887±0.15 accuracy

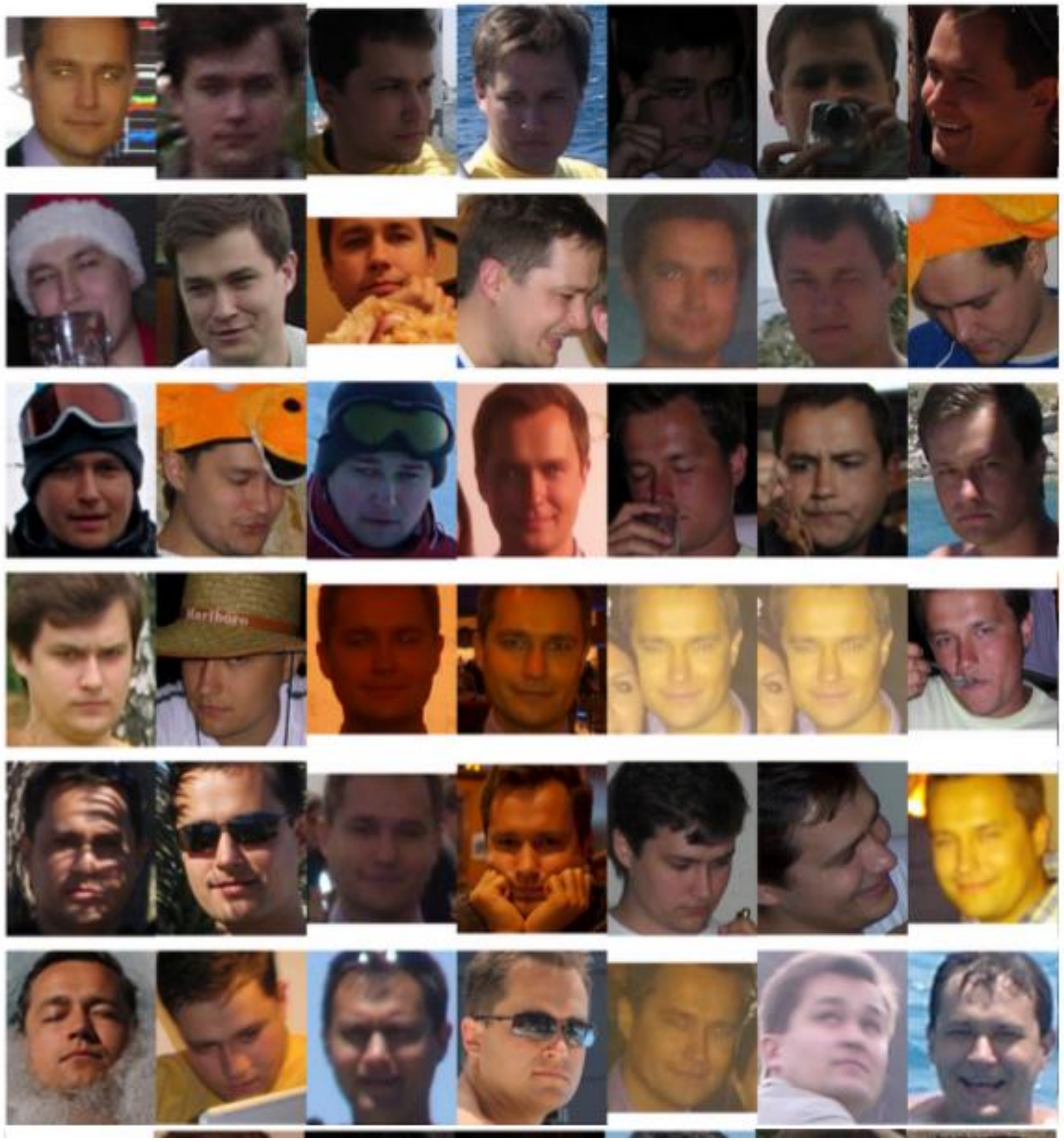
with face alignment **0.9963±0.09**

(DeepFace: **0.9735±0.0025** accuracy)



FaceNet – LFW Classification

Detection → Deep Learning → Normalization → Representation → Triplet Loss → Classification



Detection → Deep Learning → Normalization → Representation → Triplet Loss → Classification

FaceNet – Some more information

jpeg q	val-rate
10	67.3%
20	81.4%
30	83.9%
50	85.5%
70	86.1%
90	86.5%

#pixels	val-rate
1,600	37.8%
6,400	79.5%
14,400	84.5%
25,600	85.7%
65,536	86.4%

Table 4. **Image Quality.** The table on the left shows the effect on the validation rate at 10E-3 precision with varying JPEG quality. The one on the right shows how the image size in pixels effects the validation rate at 10E-3 precision. This experiment was done with NN1 on the first split of our test hold-out dataset.

#dims	VAL
64	$86.8\% \pm 1.7$
128	$87.9\% \pm 1.9$
256	$87.7\% \pm 1.9$
512	$85.6\% \pm 2.0$

Table 5. **Embedding Dimensionality.** This Table compares the effect of the embedding dimensionality of our model NN1 on our hold-out set from section 4.1. In addition to the VAL at 10E-3 we also show the standard error of the mean computed across five splits.

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
- Is our work done?

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*

Labeled Faces in the Wild

DeepFace-ensemble[41]	0.9735 ± 0.0025
ConvNet-RBM[42]	0.9252 ± 0.0038
POOF-gradhist[44]	0.9313 ± 0.0040
POOF-HOG[44]	0.9280 ± 0.0047
FR+FCN[45]	0.9645 ± 0.0025
DeepID[46]	0.9745 ± 0.0026
GaussianFace[47]	0.9852 ± 0.0066
DeepID2[48]	0.9915 ± 0.0013
TCIT[53]	0.9333 ± 0.0124
DeepID2+[55]	0.9947 ± 0.0012
betaface.com[56]	0.9808 ± 0.0016
DeepID3[57]	0.9953 ± 0.0010
insky.so[59]	0.9551 ± 0.0013
Uni-Ubi[60]	0.9900 ± 0.0032
FaceNet[62]	0.9963 ± 0.0009
Tencent-BestImage[63]	0.9965 ± 0.0025
Baidu[64]	0.9977 ± 0.0006
AuthenMetric[65]	0.9977 ± 0.0009
MMDFR[67]	0.9902 ± 0.0019
CW-DNA-1[70]	0.9950 ± 0.0022
Faceall[71]	0.9940 ± 0.0010
JustMeTalk[72]	0.9887 ± 0.0016
Facevisa[74]	0.9955 ± 0.0014
pose+shape+expression augmentation[75]	0.9807 ± 0.0060
ColorReco[76]	0.9940 ± 0.0022
Asaphus[77]	0.9815 ± 0.0039
Daream[78]	0.9968 ± 0.0009
Dahua-FaceImage[80]	0.9978 ± 0.0007
Easen Electron[81]	0.9968 ± 0.0009
Skytop Gaia[82]	0.9630 ± 0.0023

Discussion

- Different representations (1-hot vs. feature vector)
- NN depth importance
- Computational complexity vs. classification performance (trade-off)

- Results shown here are updated with the articles
  Better results have already been shown


Questions


NOT SURE IF THEY'RE CLAPPING FOR MY PRESENTATION
OR BECAUSE ITS FINISHED

Thank you!