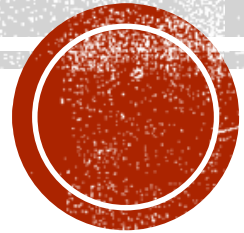
FACENET:
A UNIFIED EMBEDDING FOR FACE RECOGNITION AND CLUSTERING



CVPR 2015, Citations: 2633

Florian Schroff, Dmitry Kalenichenko, James Philbin

Google Inc.

INTRODUCTION

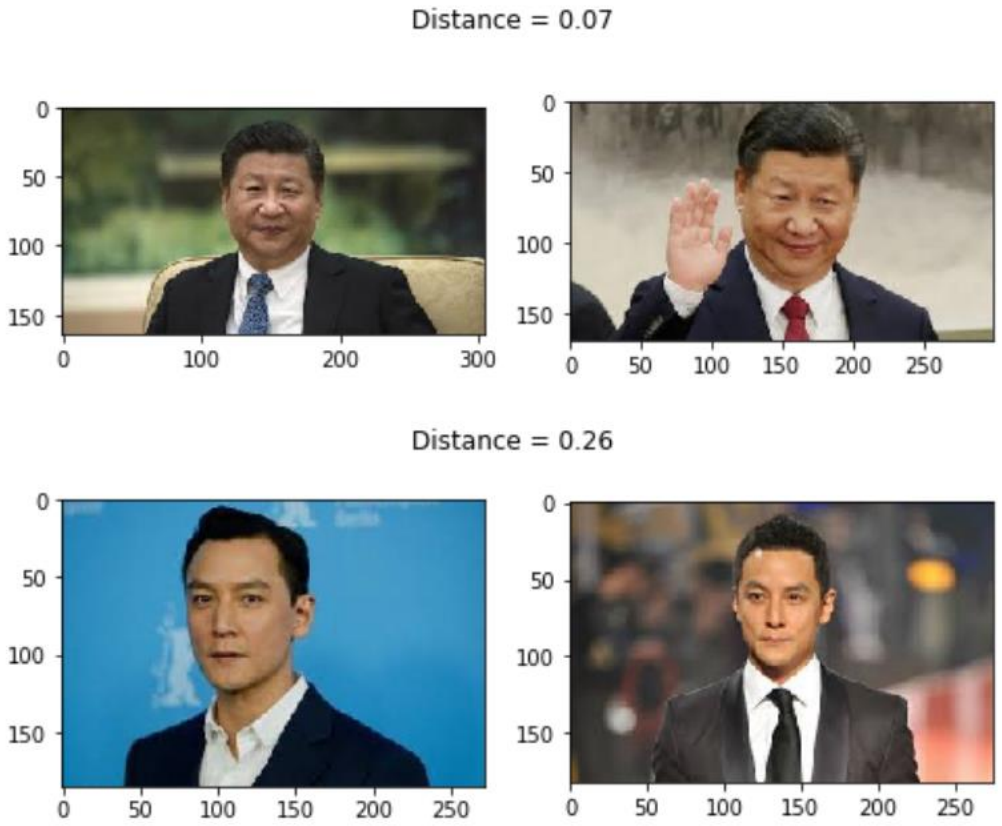
- Unified system
  - For face verification, recognition and clustering.
  - Method: a deep convolutional network.
    - Learn a Euclidean embedding per image.
    - The squared L2 distances in the embedding space directly correspond to face similarity.

UNIFIED EMBEDDING

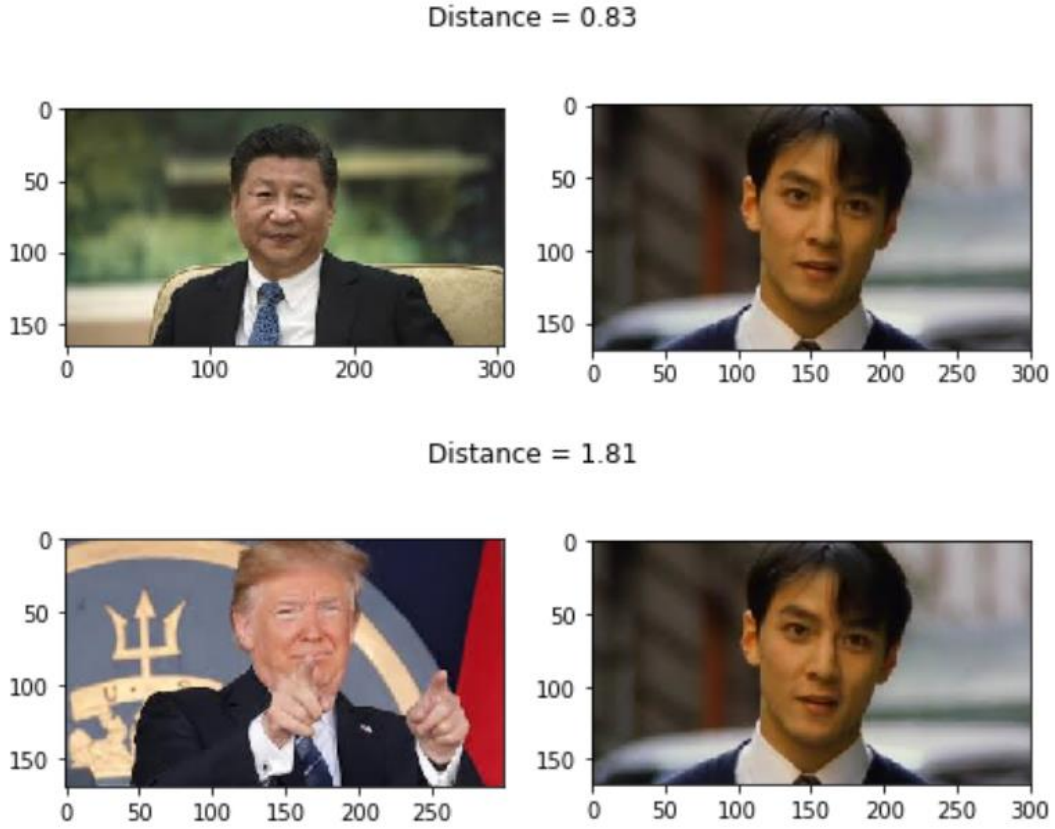
```
[ 3.54288220e-02  1.74266189e-01 -7.24426378e-03  1.08134009e-01
 -7.20467493e-02  1.52707309e-01 -4.96066455e-03 -2.68189646e-02
 -1.03350431e-02  1.15886353e-01  1.80659786e-01 -1.21214485e-03
 -1.49686364e-02 -1.37237757e-01 -6.20993786e-03 -1.06722243e-01
 -8.67237672e-02  1.80031687e-01 -3.29912244e-03 -3.18019204e-02
  1.81659255e-02 -4.85233814e-02  2.70264223e-02  8.00061598e-02
  1.09783344e-01 -1.26300439e-01  2.17632223e-02 -1.27550792e-02
  1.20648239e-02  6.38654679e-02  2.14887806e-03 -9.67042744e-02
 -3.60493958e-02  5.18226102e-02  1.17877955e-02 -7.27367848e-02
 -7.23751485e-02 -6.66370317e-02  4.46268953e-02  3.92403789e-02
 -1.29824923e-02 -1.28949985e-01 -4.10131142e-02  1.03805430e-01
 -1.45990163e-01 -2.83476412e-01  2.76148468e-02  1.69149693e-02
 -2.18285434e-03  1.20380759e-01  1.94540218e-01  1.44852459e-01
 -1.25978395e-01  2.47789267e-02 -4.03356738e-02  2.36927364e-02
  9.03213862e-03 -4.88751084e-02  3.66384685e-02 -8.81527141e-02
 -1.04331419e-01 -1.10255167e-01  1.23068556e-01 -8.18326324e-02
 -3.29800583e-02 -1.18121682e-02  6.51722774e-02 -1.45074978e-01
 -1.18661776e-01  1.59136832e-01 -2.16157828e-02  1.08660318e-01
  3.27619389e-02  1.04886152e-01 -3.41106616e-02 -1.83374971e-01
  1.25901289e-02 -2.66399663e-02 -4.14114036e-02  8.44075233e-02
 -5.93594313e-02 -3.17510851e-02  1.08349159e-01 -7.56721536e-04
  2.59755123e-02  2.06608307e-02 -2.21327823e-02 -6.66070879e-02
  1.60892934e-01  1.17503293e-02 -5.65223545e-02 -1.11236297e-01
 -2.54597398e-04 -5.01743369e-02 -3.78362499e-02 -2.15896398e-01
 -9.12959501e-02  3.91836390e-02 -1.73303615e-02 -2.86671743e-02
  2.65374817e-02  1.32940948e-01  8.75084326e-02  3.27028744e-02
  1.03489101e-01  5.73462956e-02  1.04937464e-01  4.32944894e-02
  5.78612015e-02  1.71624906e-02 -9.60274599e-04  7.11224005e-02
  2.71533523e-02  6.63196295e-02  1.47252542e-03 -8.52344036e-02
  1.37573108e-01  4.19461876e-02  1.56400278e-01  2.55480558e-02
  6.69547021e-02 -1.08182922e-01 -2.13477835e-02 -9.31352284e-03
  7.39142373e-02 -6.70083798e-03  1.81702837e-01  1.35250896e-01]
```

128 bytes

UNIFIED EMBEDDING



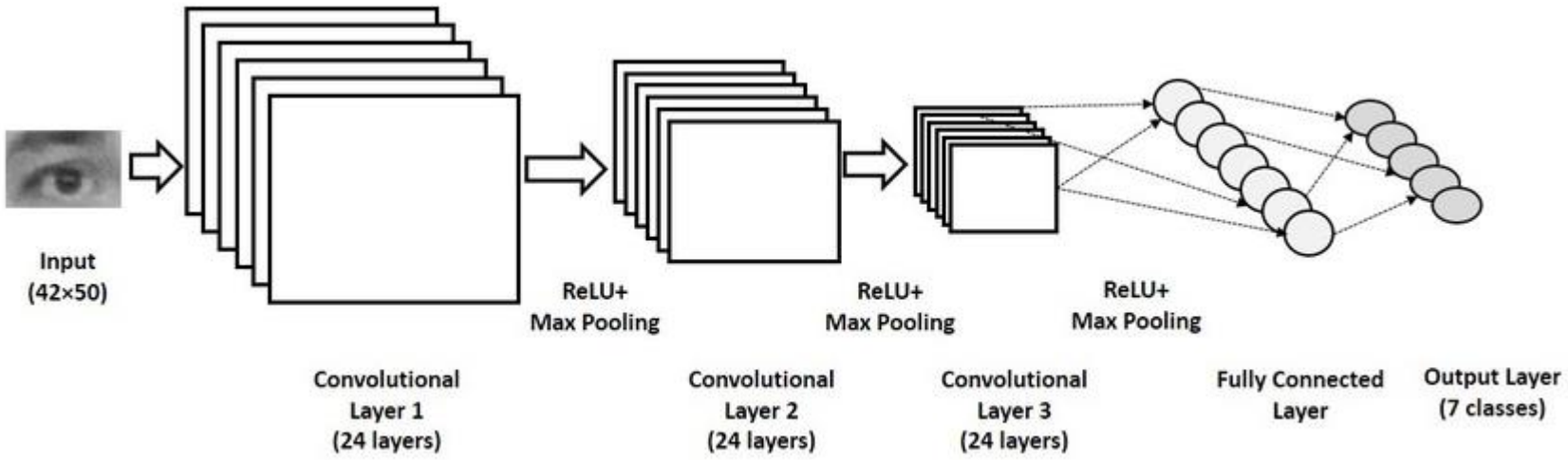
Same person



Different person

DEEP CONVOLUTIONAL NETWORK

- How previous approaches get embedding?
  (1)Train a K people classifier (2)Pick middle layer as embedding vector.

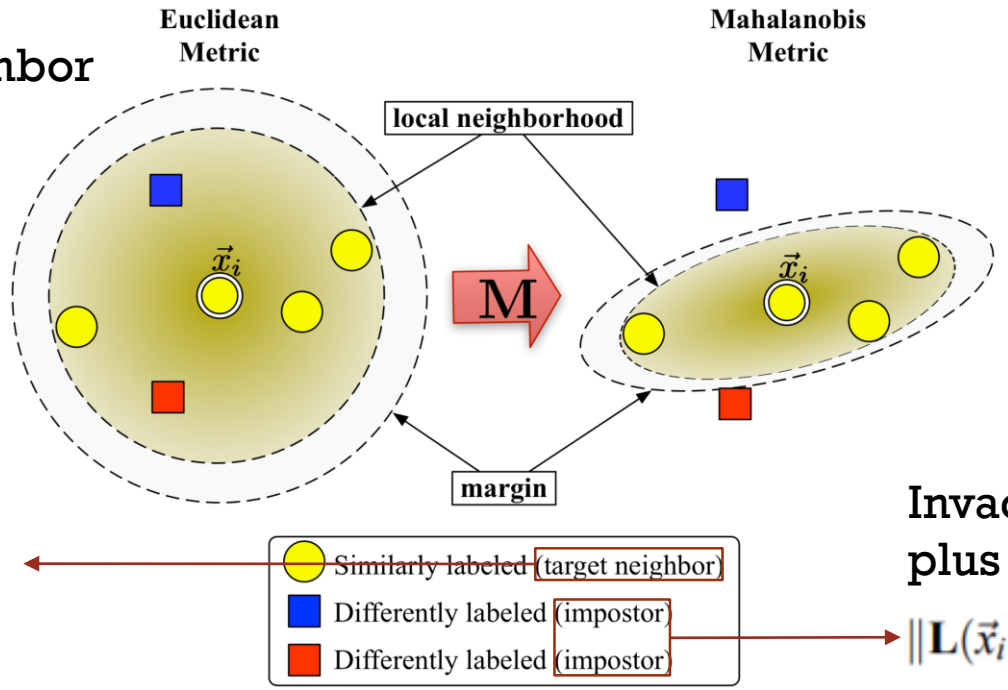


- Drawbacks: indirectness, inefficient…

FACENET APPROACH

- Trains its output to be a compact 128-D embedding using a triplet-based loss function based on LMNN.

Large Margin Nearest Neighbor

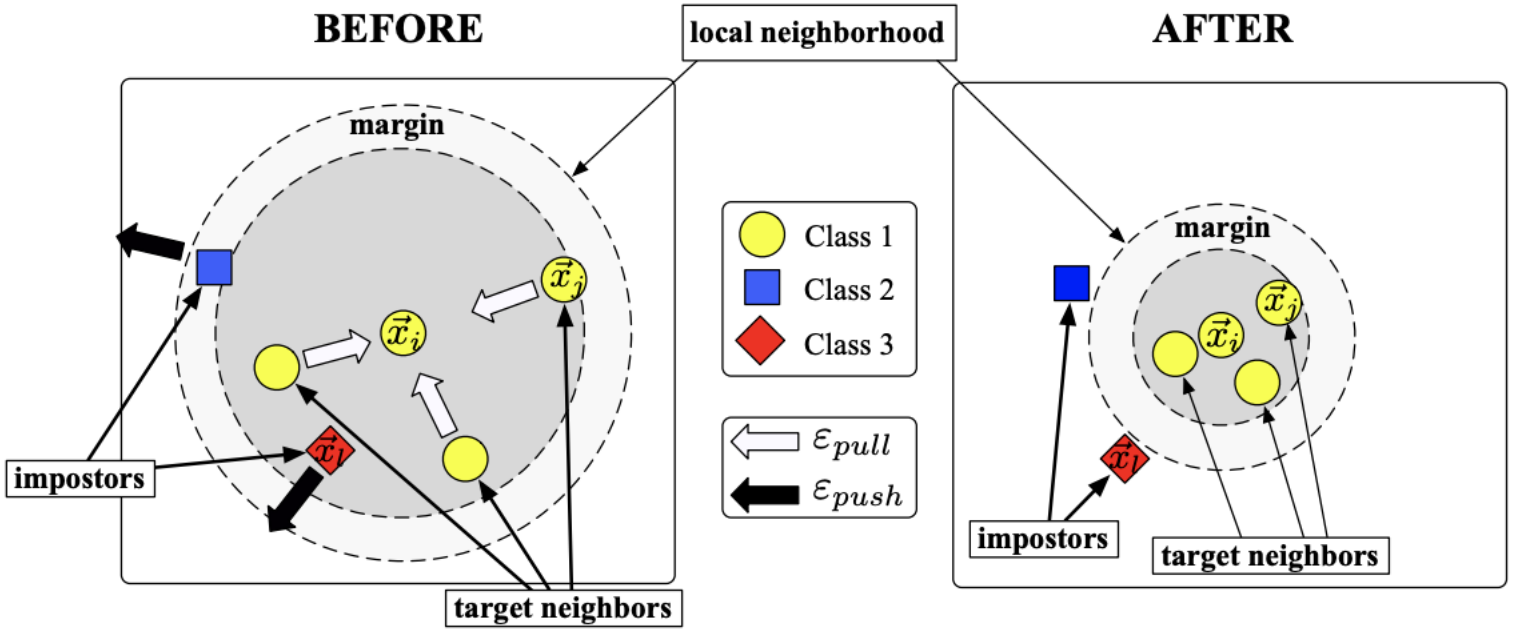


1. Prior knowledge
2. Compute k nearest neighbor with the same class label

Invades the perimeter plus unit margin

$$\|\mathbf{L}(\vec{x}_i - \vec{x}_l)\|^2 \leq \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 + 1$$

LOSS FUNCTION OF LMNN



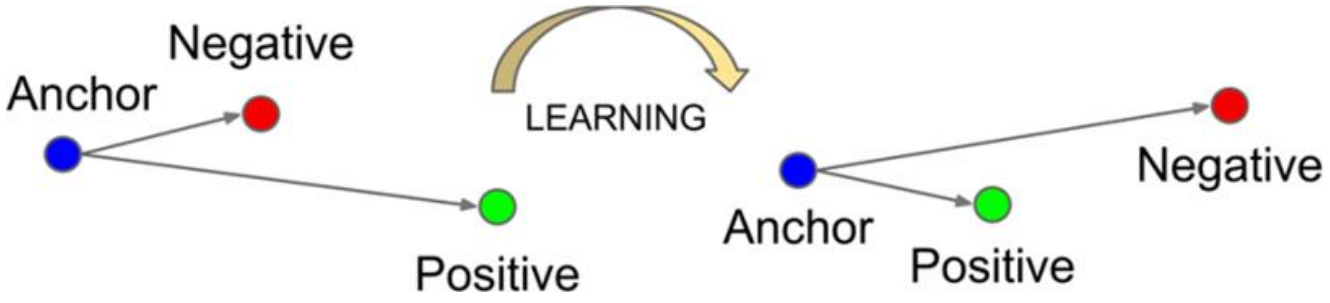
$$\varepsilon(\mathbf{L}) = (1-\mu)\,\varepsilon_{pull}(\mathbf{L}) + \mu\varepsilon_{push}(\mathbf{L}).$$

- Pull target neighbors closer together $\quad \varepsilon_{\text{pull}}(\mathbf{L}) = \sum_{j \rightsquigarrow i} \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2$
- Push differently labeled examples further apart

$$\varepsilon_{\text{push}}(\mathbf{L}) = \sum_{i,j \rightsquigarrow i} \sum_{l} (1-y_{il}) \left[1 + \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 - \|\mathbf{L}(\vec{x}_i - \vec{x}_l)\|^2\right]_+$$

TRIPLET-BASED LOSS FUNCTION

- Triplet
  - Two matching face thumbnails and a non-matching face thumbnail.
  - Thumbnails: tight crop of the face area.
- Loss $$L = \sum_{i}^{N} \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+$$
  - Aims to separate the positive from the negative by a distance margin.



TRIPLET SELECTION

- Select triplets that violate the triplet constraint

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

  - ~~Across the whole training set~~
    - Infeasible to compute
    - May lead to poor training
  - Offline: on a subset of the data
  - Online: within a mini-batch

GENERATE TRIPLETS ONLINE

- Sample the training data
  - 40 faces are selected per identity per mini-batch
  - Random sampled negative faces are added to each mini-batch
- Use all anchor-positive pairs in a mini-batch, while still selecting the hard negatives
  - Semi-hard negatives

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

- Large mini-batch, size: 1800
  - Improve convergence during SGD
  - The way of selecting hard relevant triplets

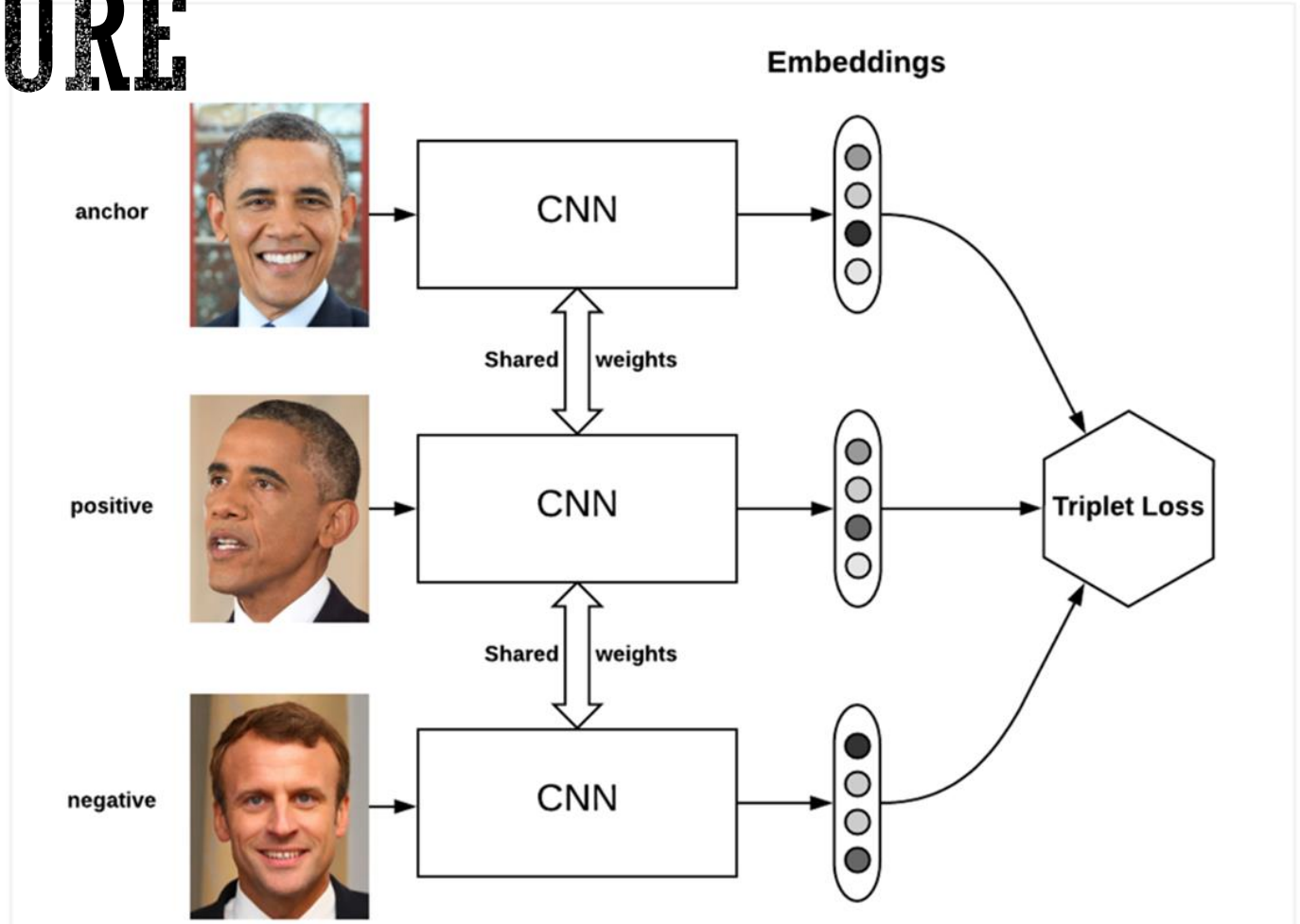
MODEL STRUCTURE

- End-to-end learning
  - Batch input layer, deep CNN, L_2 normalization
  - Followed by the triplet loss during training

  - Explore two different deep network architectures
    - Zeiler&Fergus model
    - Inception model

MODEL STRUCTURE

- End-to-end learning



Triplet loss on two positive faces (Obama) and one negative face (Macron)

DATASETS AND EVALUATION

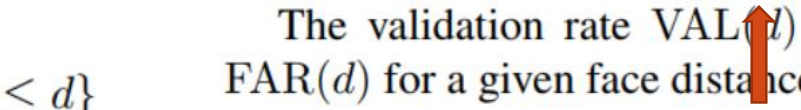
- True Accepts

$$\mathrm{TA}(d) = \{(i,j) \in \mathcal{P}_{\text{same}}, \text{with } D(x_i, x_j) \leq d\}$$

- False Accepts

$$\mathrm{FA}(d) = \{(i,j) \in \mathcal{P}_{\text{diff}}, \text{with } D(x_i, x_j) \leq d\}$$

ROC



The validation rate VAL(d) and the false accept rate FAR(d) for a given face distance d are then defined as

$$\mathrm{VAL}(d) = \frac{|\mathrm{TA}(d)|}{|\mathcal{P}_{\text{same}}|}, \quad \mathrm{FAR}(d) = \frac{|\mathrm{FA}(d)|}{|\mathcal{P}_{\text{diff}}|}. \quad (7)$$

- Standard protocol for unrestricted, labeled outside data
  - Mean accuracy of 10-folders verification set and the standard error of the mean

- Training
  - 100M – 200M face thumbnails
  - 8M different identities

DATASETS AND EVALUATION

- Hold-out Test Set
  - One million images: same distribution as training set, but disjoint identities
  - Five splits of 200k images each
    - VAL and FAR are computes on 100k*100k pairs each split
    - Standard error across the five splits

- Personal Photos
  - 12k images: similar distribution to training set, but manually verified to have clean labels.
  - Compute the VAL and FAR across all 12k squared pairs.

- Academic Datasets
  - LWF (Labeled Faces in the Wild): 13233 images, 5749 people.
  - Youtube Faces DB: 3,425 videos of 1,595 different people.

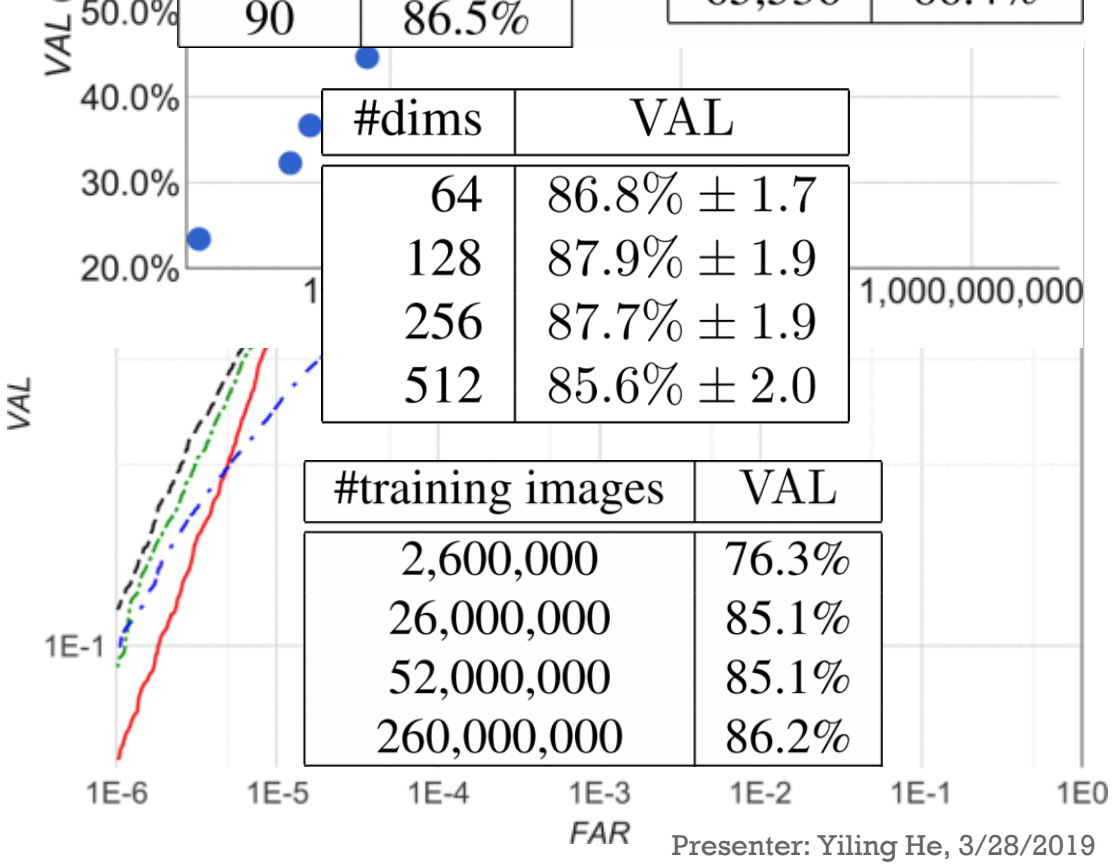
EXPERIMENTS

- Computation Accuracy Trade-off
- Effect of CNN Model
- Sensitivity to Image Quality
- Embedding Dimensionality
- Amount of Training Data

architecture	VAL
NN	
NN	
NN	
NN	
NN	
NN	

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%



#dims	VAL
64	86.8% ± 1.7
128	87.9% ± 1.9
256	87.7% ± 1.9
512	85.6% ± 2.0

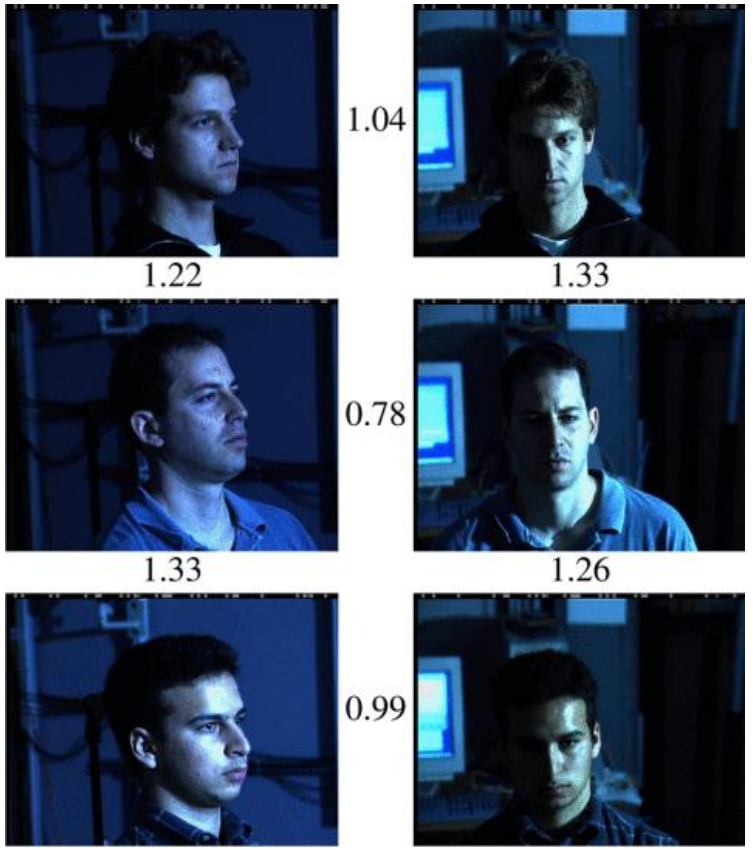
#training images	VAL
2,600,000	76.3%
26,000,000	85.1%
52,000,000	85.1%
260,000,000	86.2%

PERFORMAN DATASETS

- LFW
  - 1. Fixed center cro . 98.87%±0.15
  - 2. A proprietary fa un on the provided LFW thumbnails. If pens for two images), the LFW alignment 99.63%±0.09
    - DeepFace: (error)r eepId2+: by 30%
- Youtube Faces DB
  - Use the average si hundred frames that our face detector c 95.12%±0.39
    - DeepFace: 91.4%, I

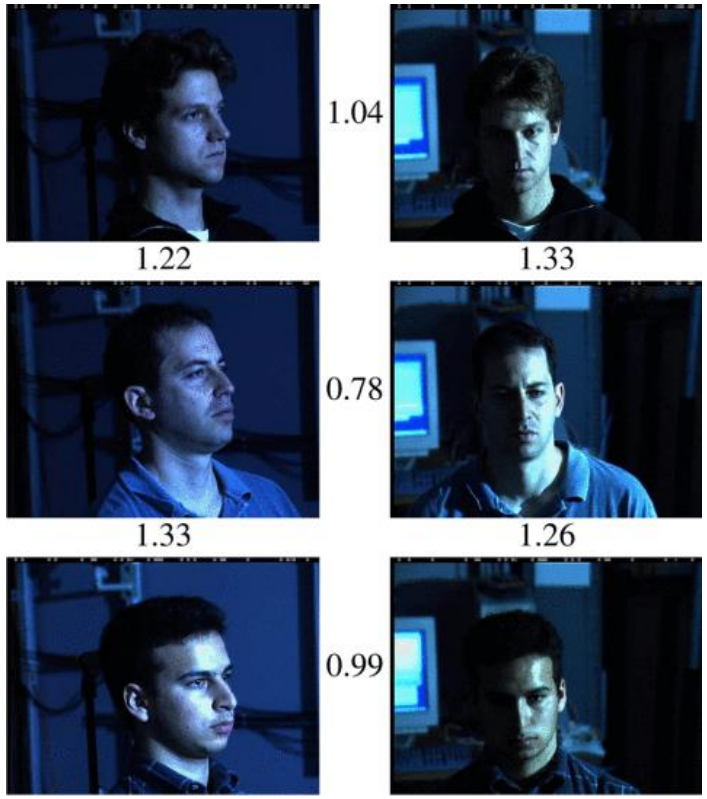
an overview of *all* failure cases

ILLUMINATION AND POSE INVARIANCE



- A threshold of 1.1 would classify every pair correctly.

FACE CLUSTERING



All these images in the users personal photo collection were clustered together.

- an exemplar cluster for one user.

SUMMARY

- Provide a method to directly learn an embedding into an Euclidean space for face verification
  - sets it apart from other methods (e.g. DeepFace / DeepId2+)
    - use the CNN bottleneck layer
    - require additional post-processing
- End-to-end training both simplifies the setup and shows that directly optimizing a loss relevant to the task at hand improves performance.
- Only requires minimal alignment (tight crop around the face area).
  - DeepFace: performs a complex 3D alignment

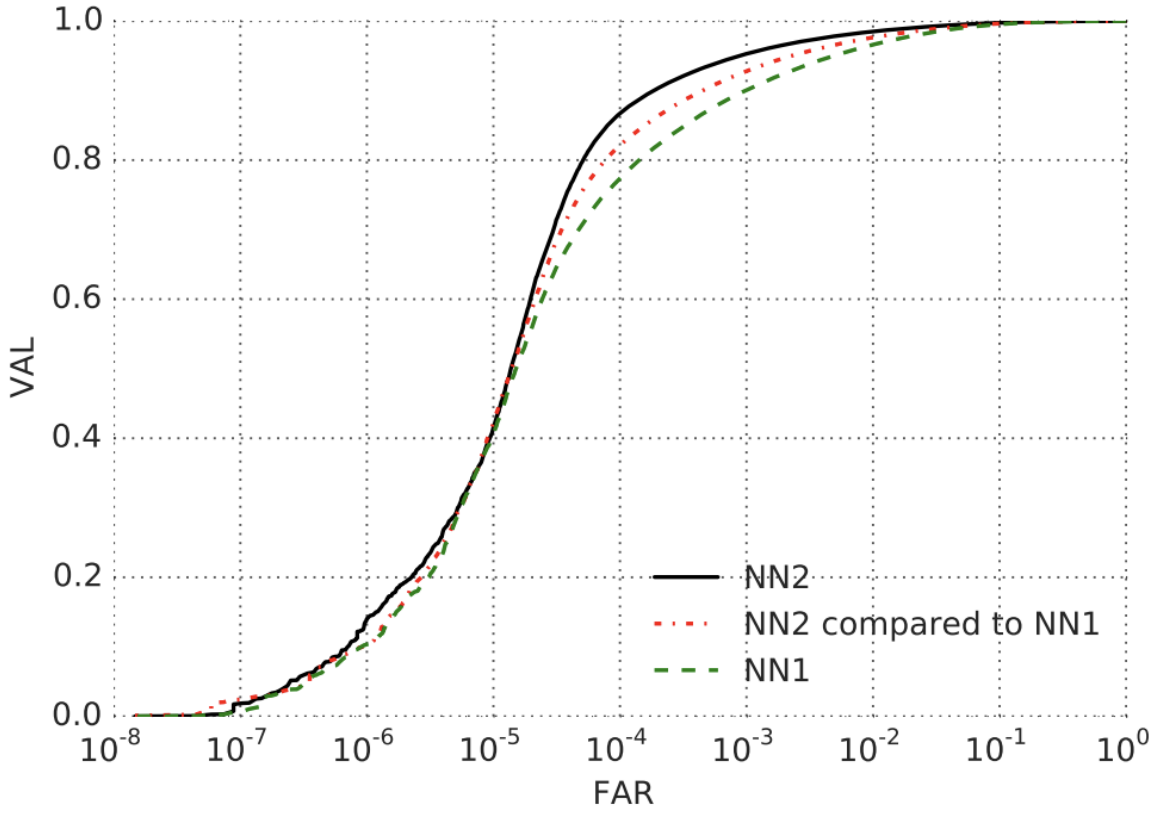
LIMITATIONS

- Reduce model size and CPU requirements.
- Improve currently extremely long training times (1000-2000h)
  - Smaller batch size
  - Offline as well as online positive and negative mining
- Not have a side-by-side comparison of hard anchor-positive pairs versus all anchor-positive pairs.
- Not directly compare to other losses.

HARMONIC EMBEDDING

Compatibility:

- a set of embeddings generated by different models v1 and v2 can be compared to each other.
- greatly simplifies upgrade paths



THANK YOU

NETWORK ARCHITECTURES

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

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

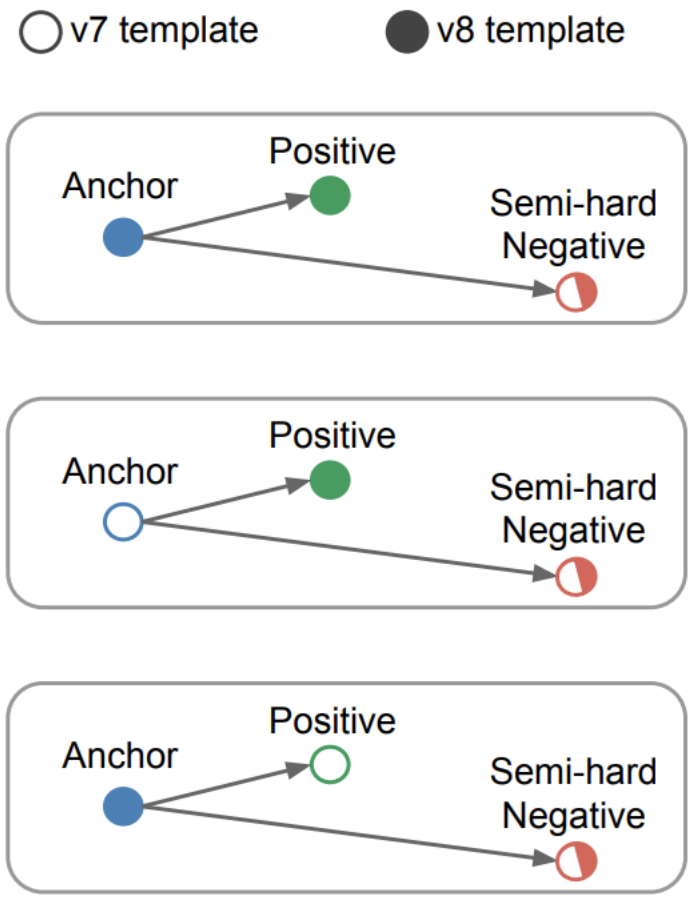


Figure 9. **Learning the Harmonic Embedding.** In order to learn a *harmonic* embedding, we generate triplets that mix the v1 embeddings with the v2 embeddings that are being trained. The semi-hard negatives are selected from the whole set of both v1 and v2 embeddings.