



Computer Vision; Face Recognition and Detection



[YouTube Playlist](#)

Maziar Raissi

Assistant Professor

Department of Applied Mathematics

University of Colorado Boulder

maziar.raissi@colorado.edu



Boulder

DeepFace: Closing the Gap to Human-Level Performance in Face Verification



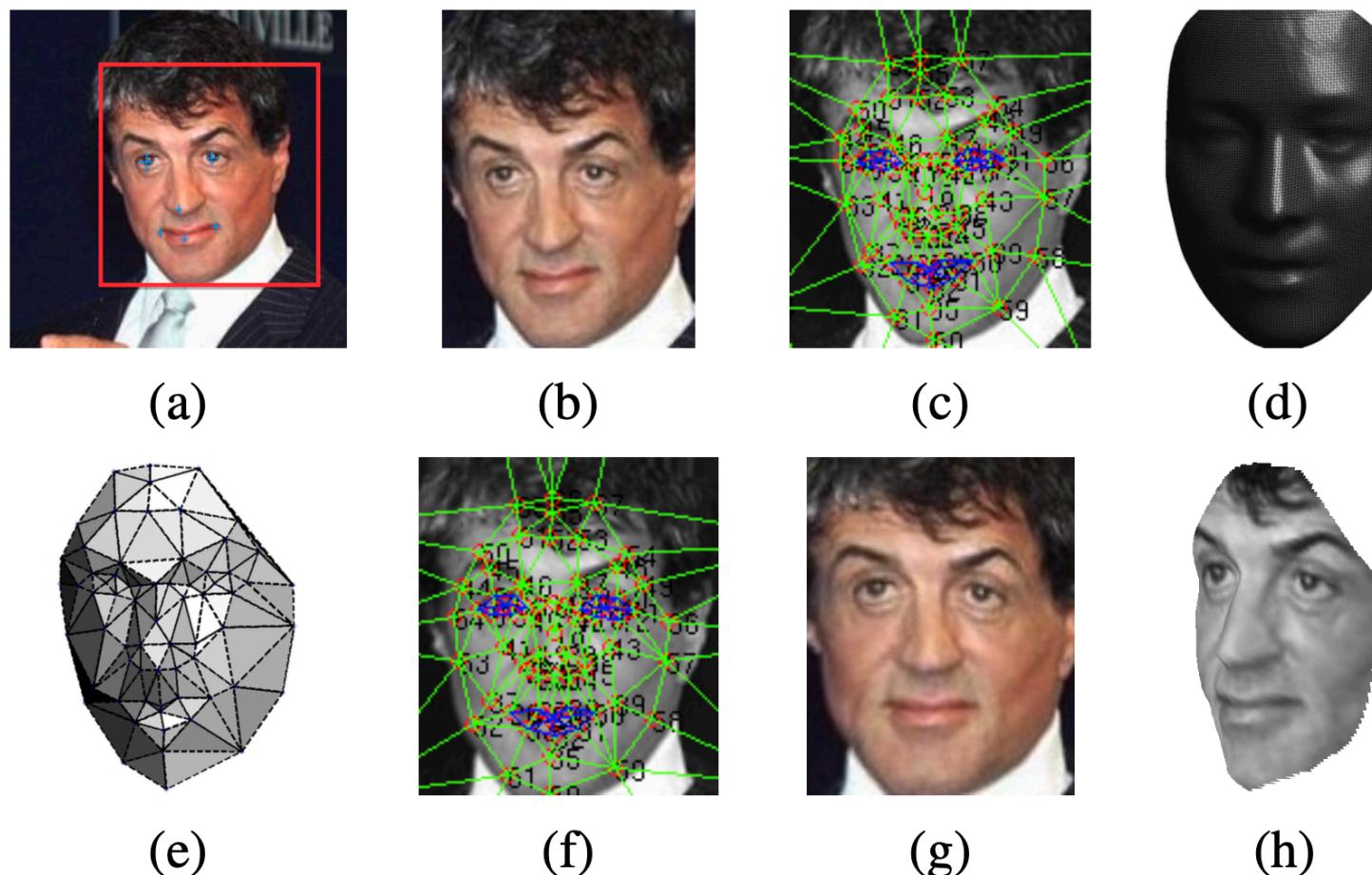
[YouTube Video](#)

detect \triangleright align \triangleright represent \triangleright classify

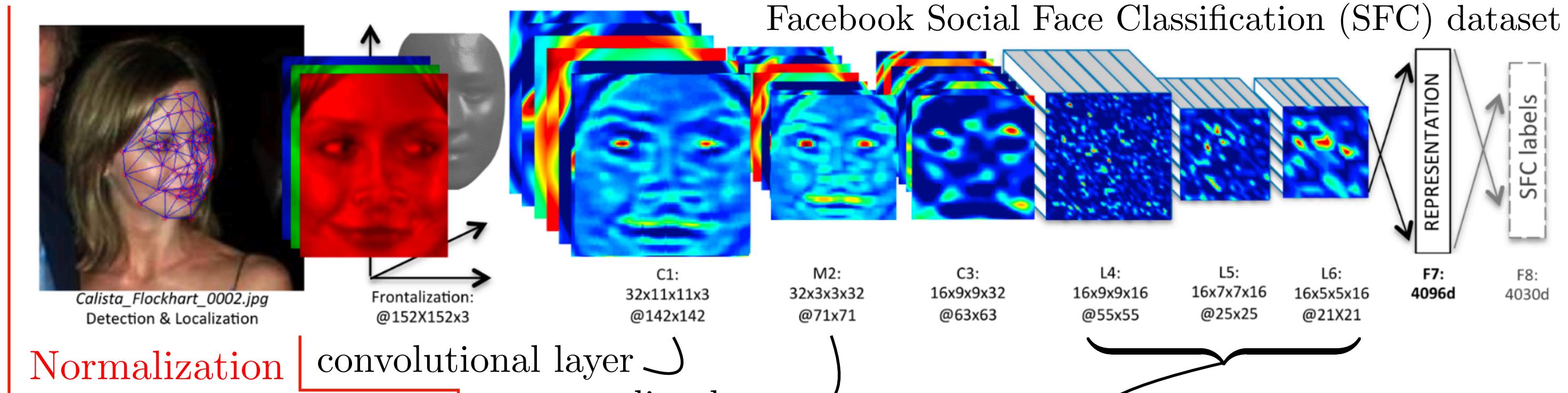
Model-based alignment: Explicit 3D face modeling in order to apply a piecewise affine transformation

Face representation from a nine-layer deep neural network

- Labeled Faces in the Wild (LFW) dataset
- YouTube Faces (YTF) dataset



Alignment pipeline. (a) The detected face, with 6 initial fiducial points. (b) The induced 2D-aligned crop. (c) 67 fiducial points on the 2D-aligned crop with their corresponding Delaunay triangulation, we added triangles on the contour to avoid discontinuities. (d) The reference 3D shape transformed to the 2D-aligned crop image-plane. (e) Triangle visibility w.r.t. to the fitted 3D-2D camera; darker triangles are less visible. (f) The 67 fiducial points induced by the 3D model that are used to direct the piece-wise affine warpping. (g) The final frontalized crop. (h) A new view generated by the 3D model (not used in this paper).



Normalization

$G(I) \rightarrow \text{representation}$

$$f(I) := \overline{G}(I)/\|\overline{G}(I)\|_2$$

$$\overline{G}(I)_i = \frac{G(I)_i}{\max(G(I)_i, \epsilon)}$$

convolutional layer

max-pooling layer

every location in the feature map learns a different set of filters

Since different regions of an aligned image have different local statistics, the spatial stationarity assumption of convolution cannot hold

Face verification metric: inner product between the two normalized feature vectors

$$\chi^2(f_1, f_2) = \sum_i w_i (f_1[i] - f_2[i])^2 / (f_1[i] + f_2[i])$$

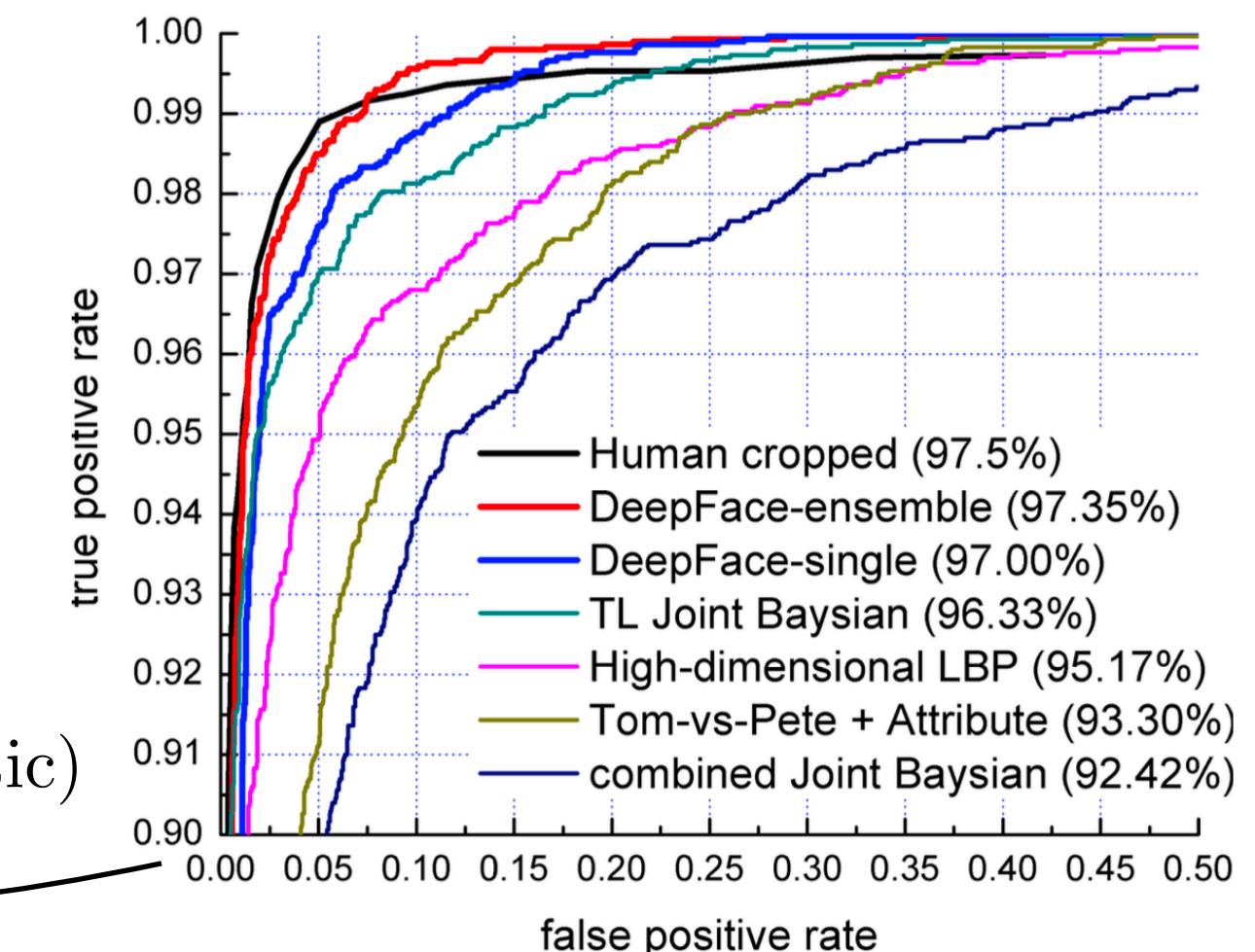
learned using a linear SVM

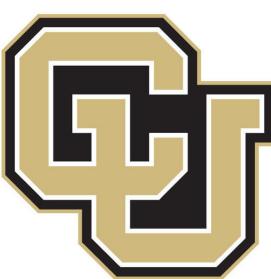
Weighted χ^2 distance

$$d(f_1, f_2) = \sum_i \alpha_i |f_1[i] - f_2[i]|$$

Siamese Network induced distance
(trained by cross entropy loss)

The ROC (Receiver Operating Characteristic)
curves on the LFW dataset





Boulder

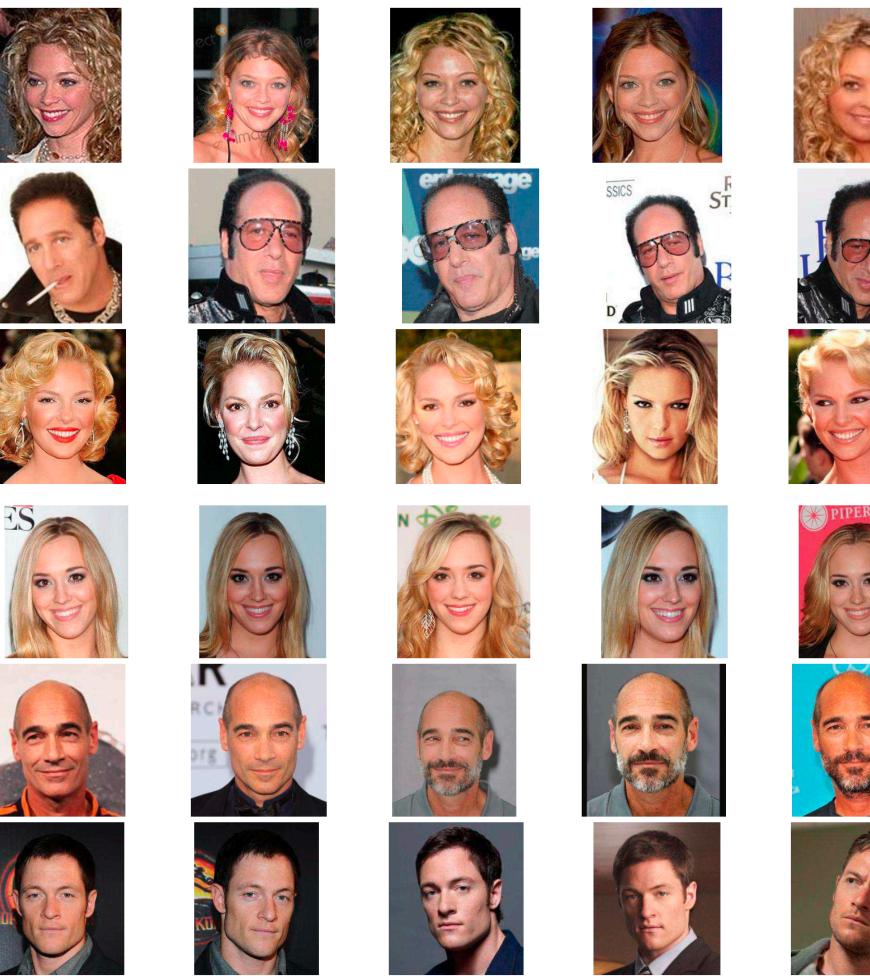


[YouTube Playlist](#)

Deep Face Recognition

Dataset	Identities	Images
LFW	5,749	13,233
WDRef [4]	2,995	99,773
CelebFaces [25]	10,177	202,599

Dataset	Identities	Images
Ours	2,622	2.6M
FaceBook [29]	4,030	4.4M
Google [17]	8M	200M



Stage	Aim	Type	# of persons	# of images per person	total # images	Annotation effort	100%-EER
1	Candidate list generation	A	5,000	200	1,000,000	—	—
2	Image set expansion	M	2,622	2,000	5,244,000	4 days	—
3	Rank image sets	A	2,622	1,000	2,622,000	—	96.90
4	Near dup. removal	A	2,622	623	1,635,159	—	—
5	Final manual filtering	M	2,622	375	982,803	10 days	92.83

Dataset Collection

- IMDB (Internet Movie Data Base)
- Freebase Knowledge Graph
- Google Image Search

No.	Config	Data	Train Align.	Test Align.	Embedding	100% - EER
1	A	C	No	No	No	92.83
2	A	F	No	No	No	95.80
3	A	F	No	Yes	No	96.70
4	B	F	No	Yes	No	97.27
5	B	F	Yes	Yes	No	96.17
6	D	F	No	Yes	No	96.73
7	B	F	No	Yes	Yes	99.13

The top 50 images (based on Google search rank in the downloaded set) for each identity are used as positive training samples, and the top 50 images of all other identities are used as negative training samples. A one-vs-rest linear SVM is trained for each identity using the Fisher Vector Faces descriptor.

VLAD descriptor ▷ Clustering ▷ Near Duplicate Removal

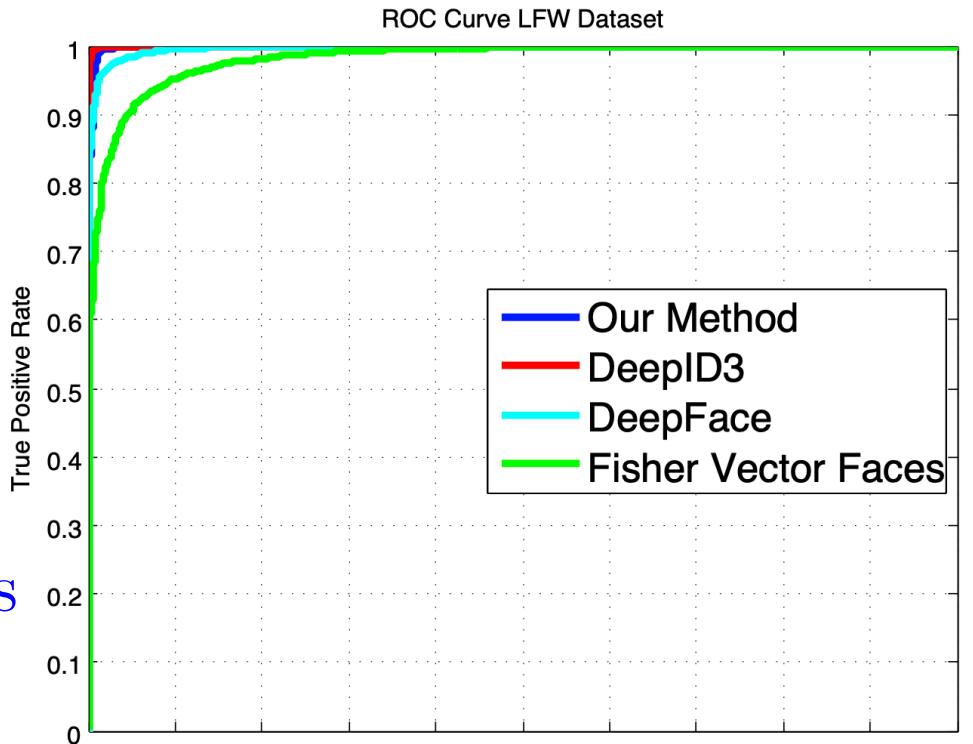
Final manual filtering is aided with AlexNet trained to discriminate between the 2,622 face identities.

layer type name	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
input	conv	relu	conv	relu	mpool	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv	
–	conv1_1	relu1_1	conv1_2	relu1_2	pool1	conv2_1	relu2_1	conv2_2	relu2_2	pool2	conv3_1	relu3_1	conv3_2	relu3_2	conv3_3	relu3_3	pool3	conv4_1	
support	–	3	1	3	1	2	3	1	3	1	2	3	1	3	1	2	3	1	
filt dim	–	3	–	64	–	–	64	–	128	–	–	128	–	256	–	256	–	256	
num filters	–	64	–	64	–	–	128	–	128	–	–	256	–	256	–	256	–	512	
stride	–	1	1	1	1	2	1	1	1	1	2	1	1	1	1	1	2	1	
pad	–	1	0	1	0	0	1	0	1	0	0	1	0	1	0	1	0	1	
layer type name	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	softmax	
relu4_1	conv4_2	relu4_2	conv4_3	relu4_3	pool4	conv5_1	relu5_1	conv5_2	relu5_2	conv5_3	relu5_3	pool5	fc6	relu6	fc7	relu7	fc8	prob	
support	1	3	1	3	1	2	3	1	3	1	2	7	1	1	1	1	1	1	
filt dim	–	512	–	512	–	–	512	–	512	–	–	512	–	4096	–	4096	–	–	
num filters	–	512	–	512	–	–	512	–	512	–	–	4096	–	4096	–	2622	–	–	
stride	1	1	1	1	1	2	1	1	1	1	2	1	1	1	1	1	1	1	
pad	0	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	

Youtube Faces Dataset

EER: Equal Error Rate
error rate @ the ROC operating point where false positive & false negative rates are equal

No.	Method	Images	Networks	Acc.
1	Fisher Vector Faces [21]	-	-	93.10
2	DeepFace [29]	4M	3	97.35
3	Fusion [30]	500M	5	98.37
4	DeepID-2,3	200		99.47
5	FaceNet [17]	200M	1	98.87
6	FaceNet [17] + Alignment	200M	1	99.63
7	Ours	2.6M	1	98.95



K indicates the number of faces used to represent each video

No.	Method	Images	Networks	100%- EER	Acc.
1	Video Fisher Vector Faces [15]	-	-	87.7	83.8
2	DeepFace [29]	4M	1	91.4	91.4
3	DeepID-2,2+,3	200		-	93.2
4	FaceNet [17] + Alignment	200M	1	-	95.1
5	Ours (<i>K</i> = 100)	2.6M	1	92.8	91.6
6	Ours (<i>K</i> = 100) + Embedding learning	2.6M	1	97.4	97.3

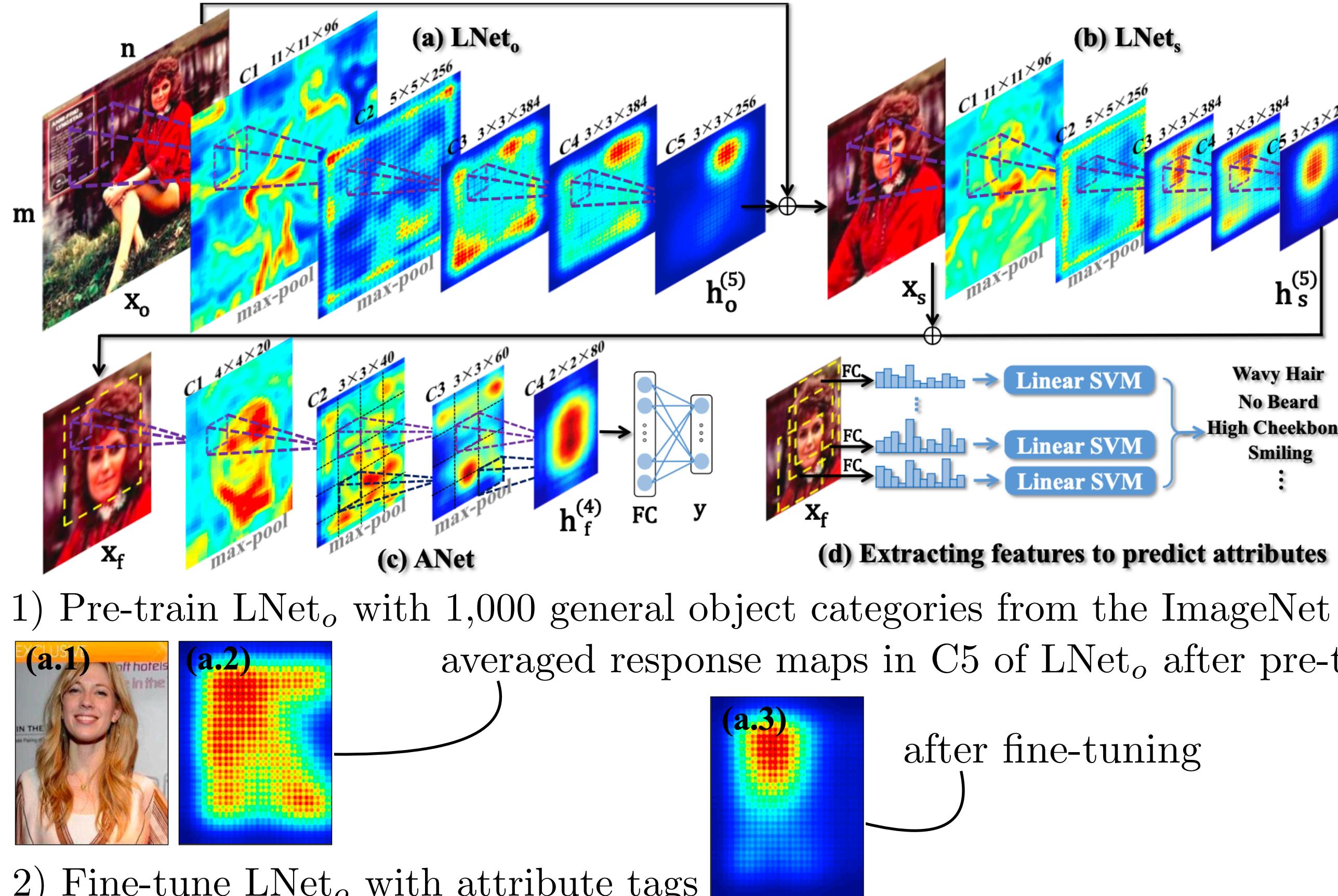


Boulder

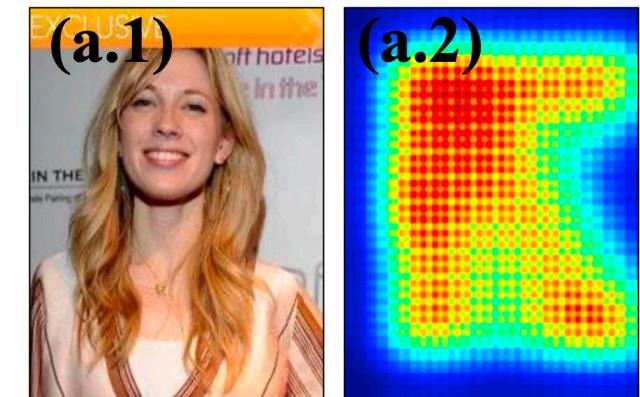
Deep Learning Face Attributes in the Wild



[YouTube Video](#)



- 1) Pre-train LNet_o with 1,000 general object categories from the ImageNet dataset
averaged response maps in C_5 of LNet_o after pre-training

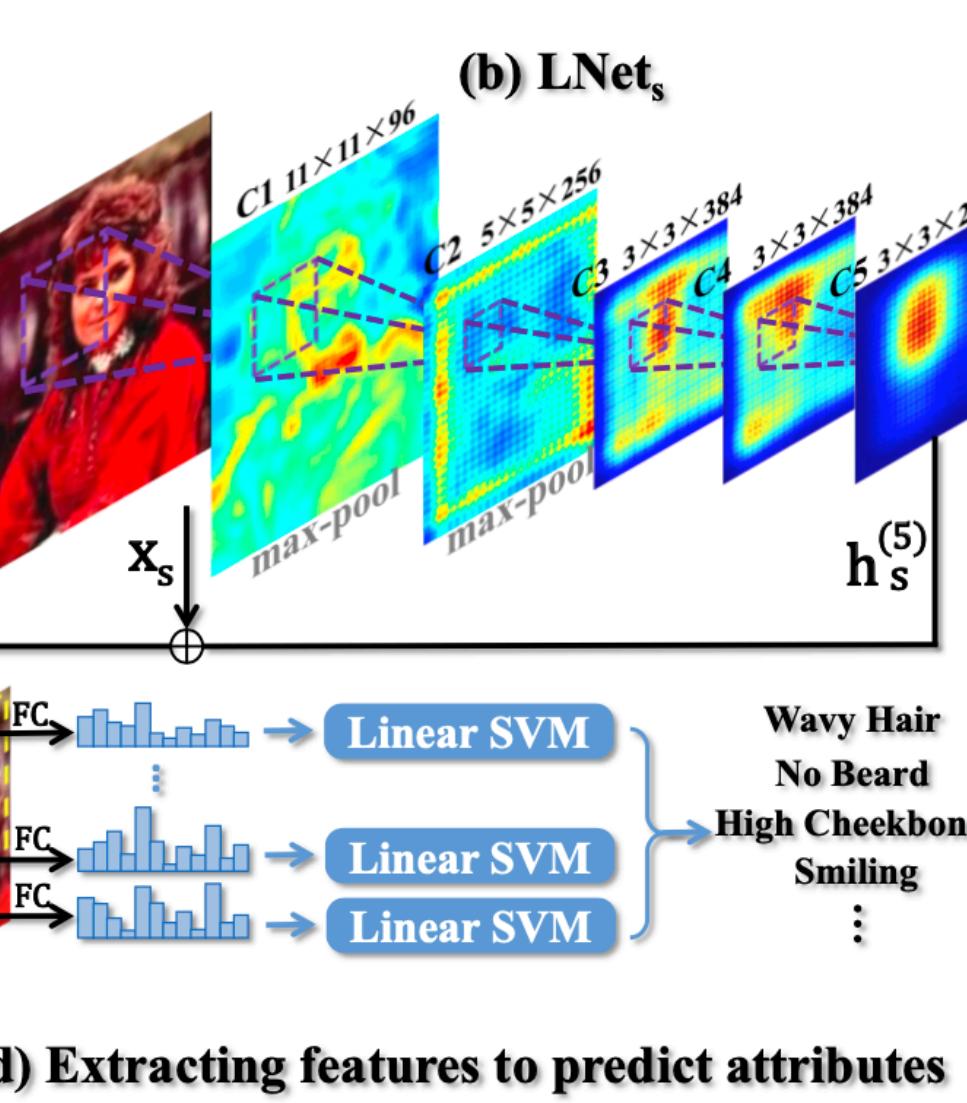


- 2) Fine-tune LNet_o with attribute tags

$$L = \sum_i y_i \log p(y_i|x) + (1 - y_i) \log(1 - p(y_i|x))$$

probability of the i -th attribute given image x

- 3) For each image, EdgeBox is adopted to propose 500 candidate windows, each of which is measured by a score that sums over its response values normalized by its window size. (Localize a face region using the window with the largest score)



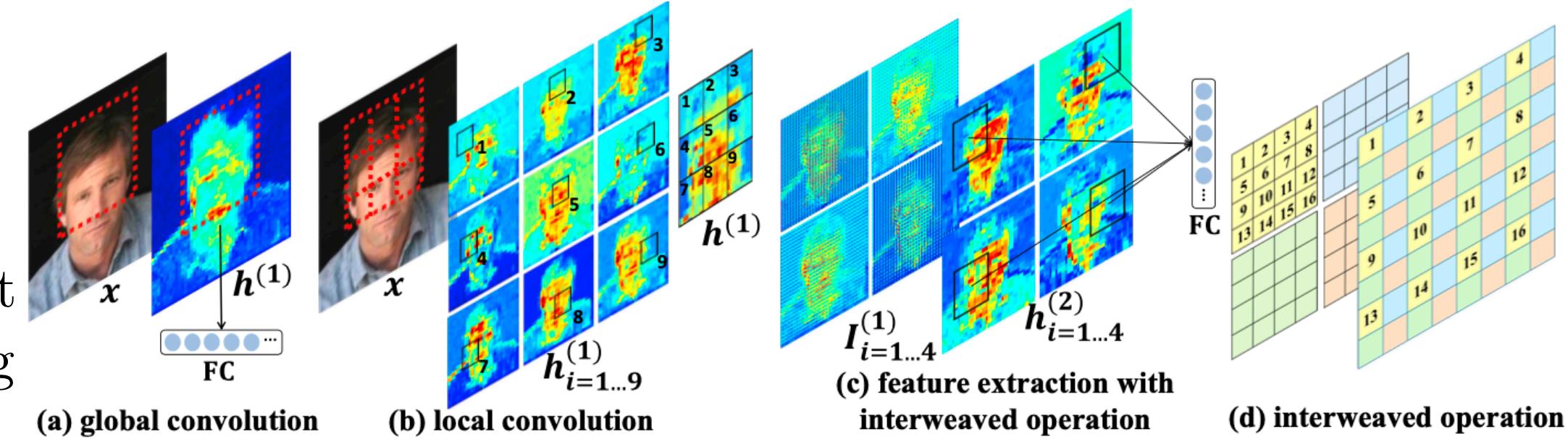
4) Repeat steps 1-3 for LNet_s

5) Pre-train ANet with a large number of face identities from the CelebFaces dataset

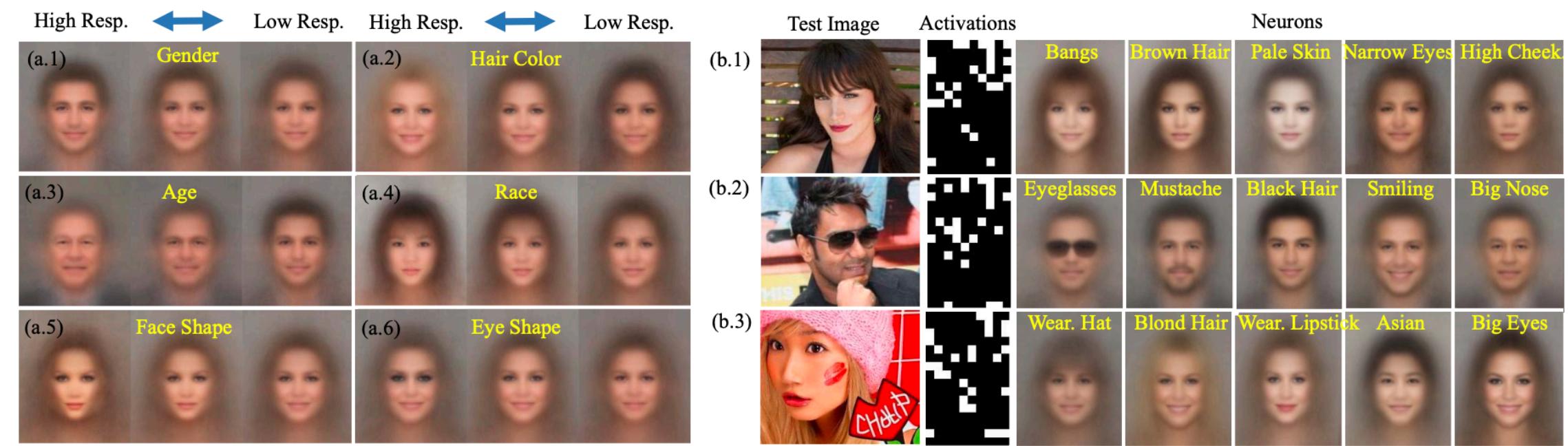
ANet is pre-trained by combining the softmax loss and the similarity loss

$$\mathcal{L} = \sum_{i,y_i=y_j} \| \text{FC}_i - \text{FC}_j \|_2^2 \rightarrow \text{improve intra-class invariance}$$

6) ANet is fine-tuned by attributes to learn high-level feature FC



Detailed pipeline of efficient feature extractions in ANet.



Visualization of neurons in ANet (a) after pre-training (b) after fine-tuning

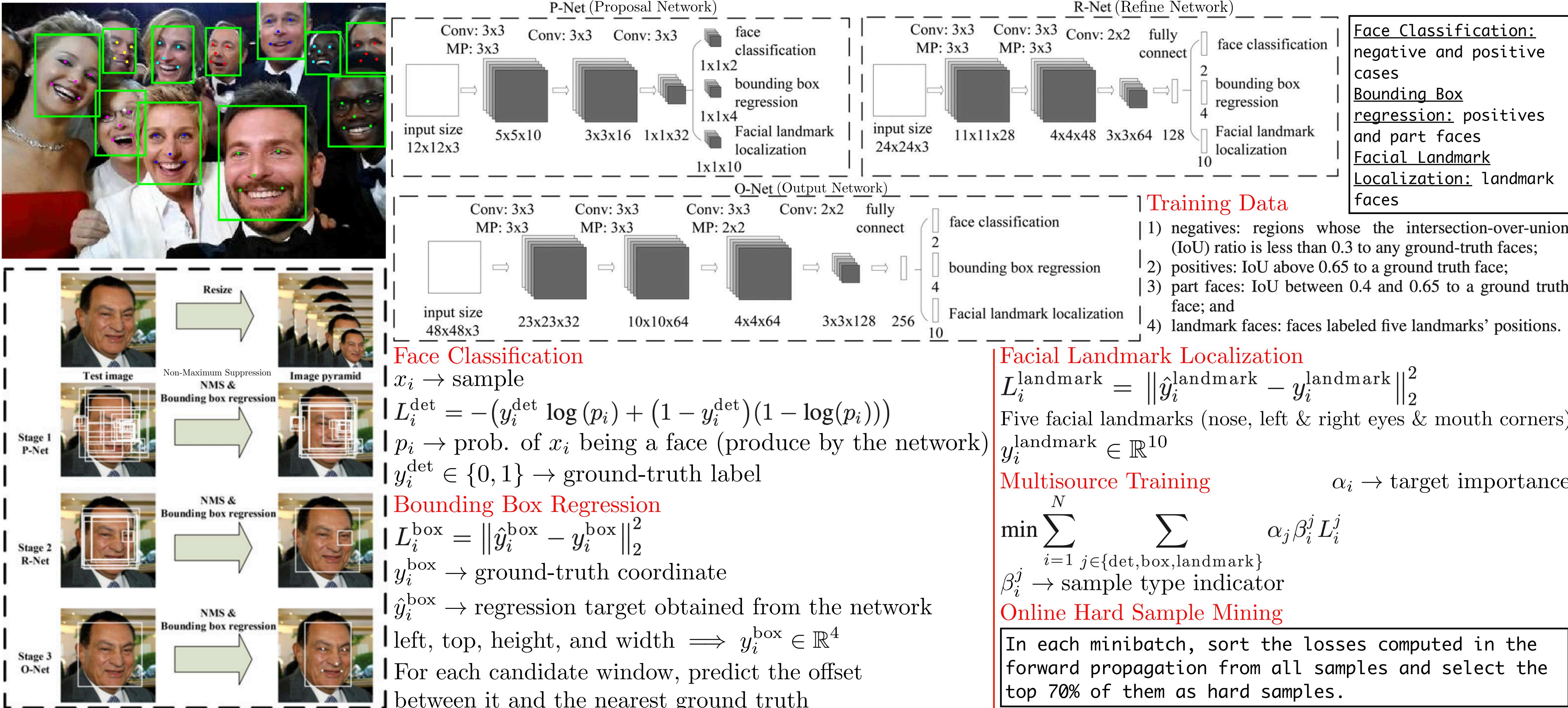


Boulder

Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks



[YouTube Video](#)



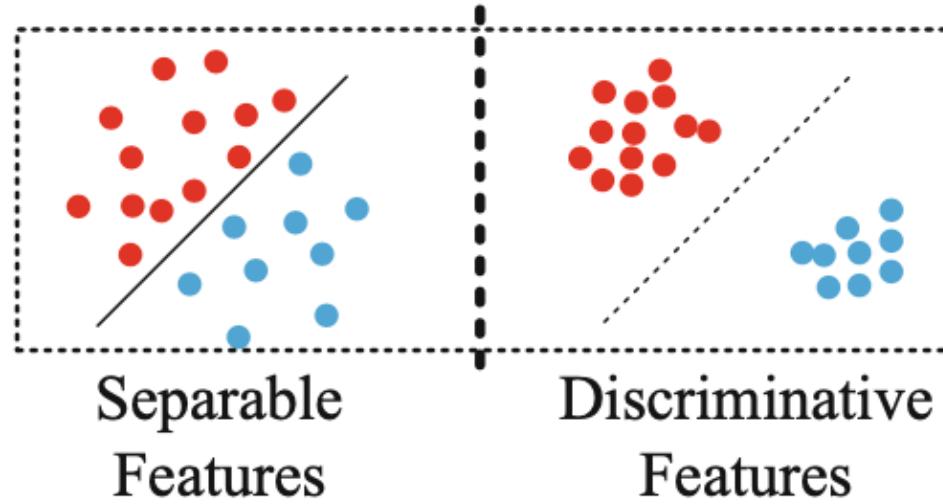


Boulder

A Discriminative Feature Learning Approach for Deep Face Recognition



[YouTube Video](#)



- inter-class dispersion
- intra-class compactness

$$\mathcal{L}_S = - \sum_{i=1}^m \log \frac{e^{W_{y_i}^T \mathbf{x}_i + b_{y_i}}}{\sum_{j=1}^n e^{W_j^T \mathbf{x}_i + b_j}} \rightarrow \text{softmax loss}$$

$\mathbf{x}_i \in \mathbb{R}^d \rightarrow$ deep features of i -th example belonging to class y_i

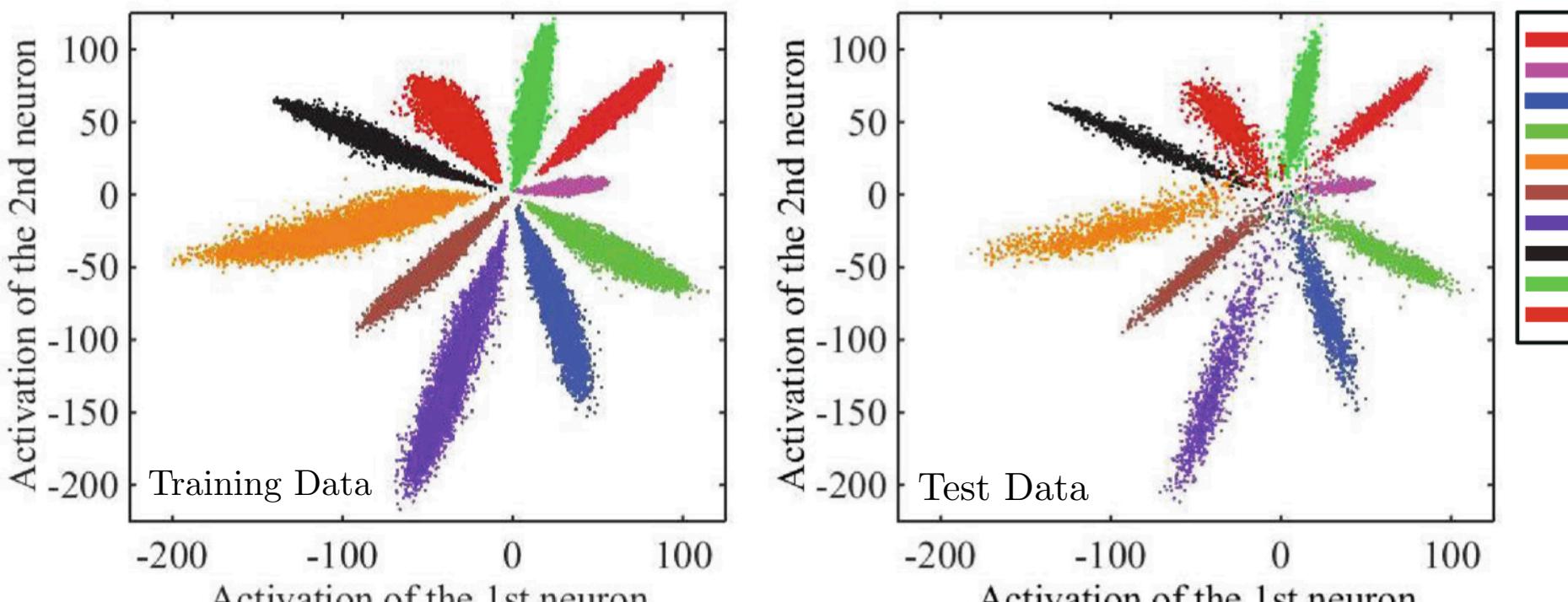
$W_j \in \mathbb{R}^d \rightarrow j$ -th column of the weights $W \in \mathbb{R}^{d \times n}$

$b \in \mathbb{R}^n \rightarrow$ bias

$m \rightarrow$ size of mini-batch

$n \rightarrow$ number of classes

MNIST Example (CNN : $I_i \mapsto \mathbf{x}_i \in \mathbb{R}^2$)



(a)

(b)

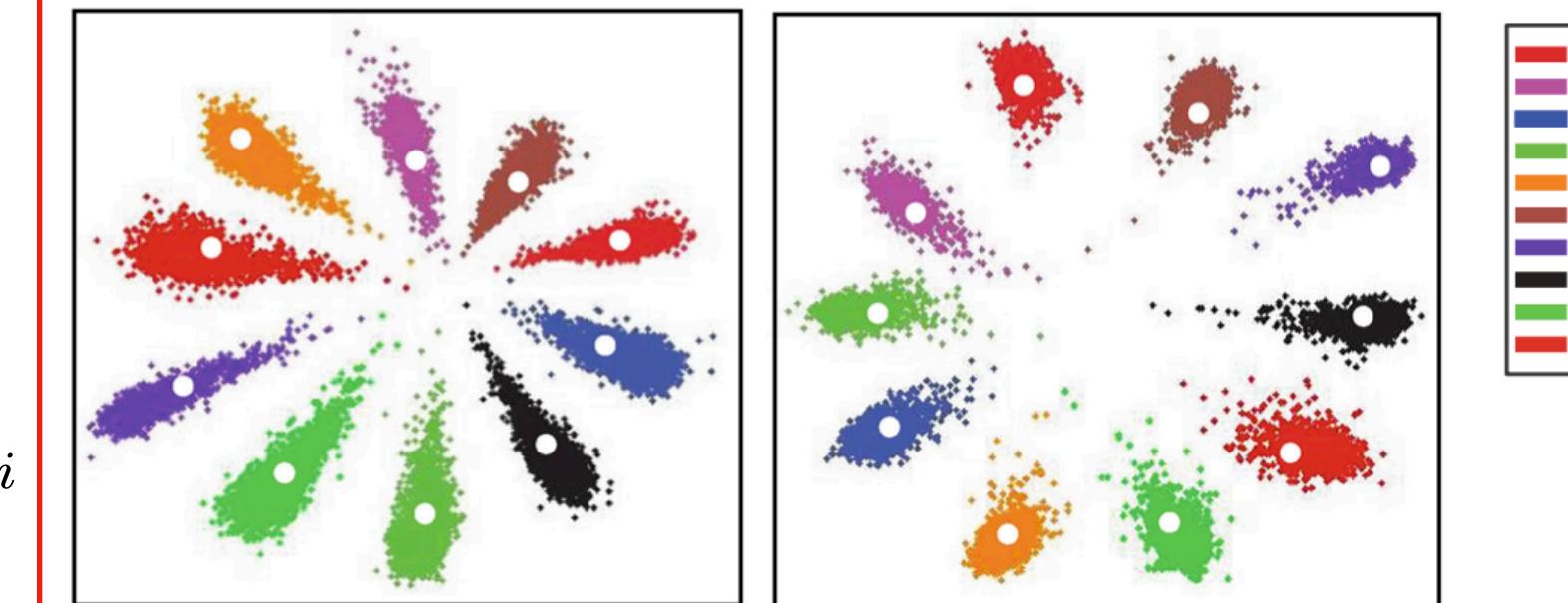
Wen, Yandong, et al. "A discriminative feature learning approach for deep face recognition." *European conference on computer vision*. Springer, Cham, 2016.

The Center Loss

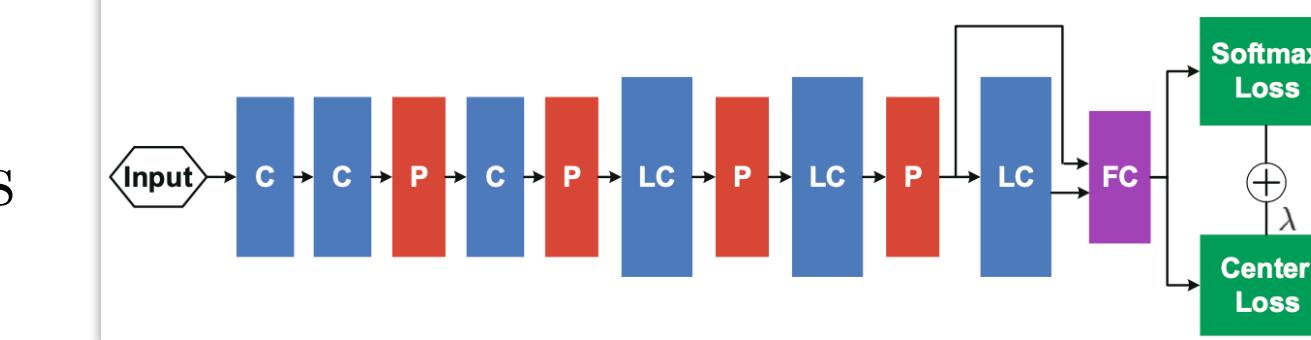
$$\mathcal{L}_C = \frac{1}{2} \sum_{i=1}^m \|\mathbf{x}_i - \mathbf{c}_{y_i}\|_2^2$$

$\mathbf{c}_{y_i} \in \mathbb{R}^d \rightarrow y_i$ th class center of deep features

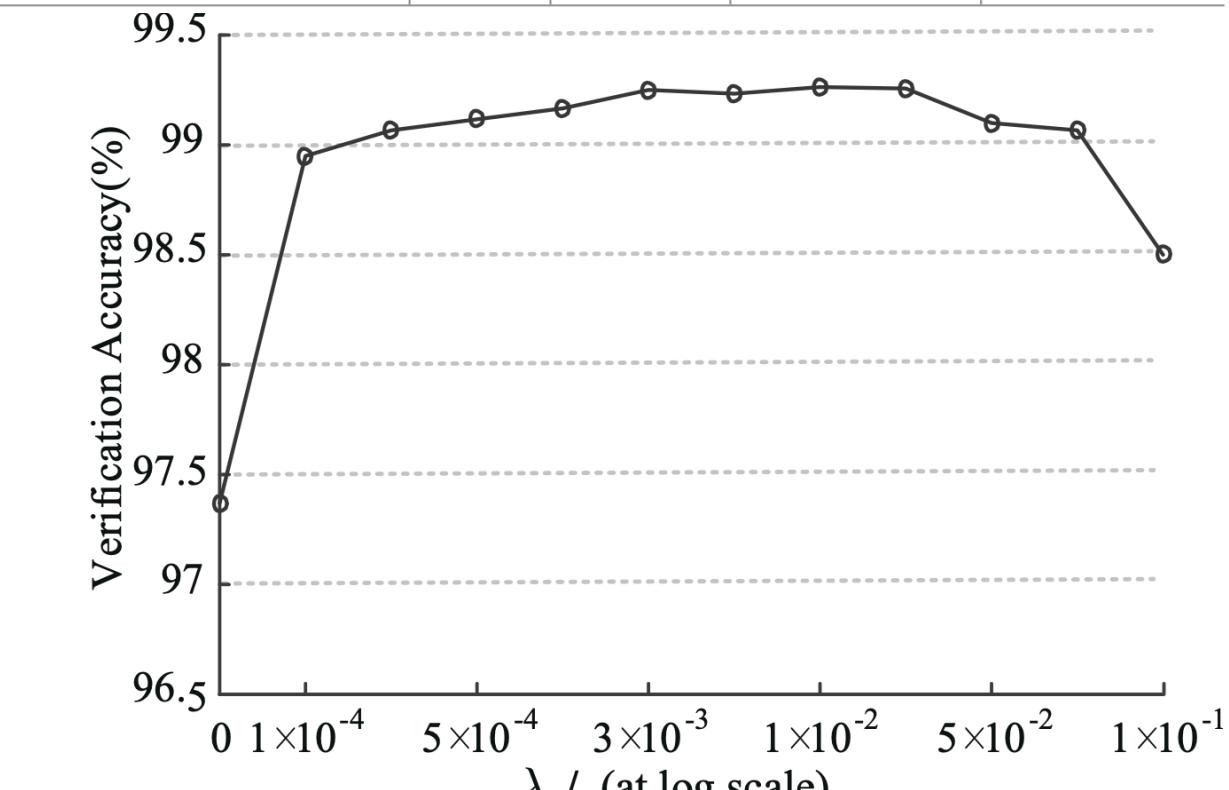
$$\mathcal{L} = \mathcal{L}_S + \lambda \mathcal{L}_C$$

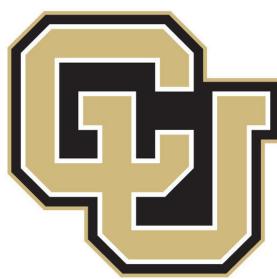


C: The convolution layer
P: The max-pooling layer
LC: The local convolution layer
FC: The fully connected layer



Method	Images	Networks	Acc. on LFW	Acc. on YTF
DeepFace [34]	4M	3	97.35 %	91.4 %
DeepID-2+ [32]	-	1	98.70 %	-
DeepID-2+ [32]	-	25	99.47 %	93.2 %
FaceNet [27]	200M	1	99.63 %	95.1 %
Deep FR [25]	2.6M	1	98.95 %	97.3 %
Baidu [21]	1.3M	1	99.13 %	-
model A	0.7M	1	97.37 %	91.1 %
model B	0.7M	1	99.10 %	93.8 %
model C (Proposed)	0.7M	1	99.28 %	94.9 %





Boulder

In Defense of the Triplet Loss for Person Re-Identification



[YouTube Video](#)

Metric embedding learning: Learn a function $f_\theta : \mathbb{R}^F \rightarrow \mathbb{R}^D$ which maps semantically similar points from the data manifold in \mathbb{R}^F onto metrically close points in \mathbb{R}^D .

$$D : \mathbb{R}^D \rightarrow \mathbb{R}$$

↳ a metric function measuring distances in the embedding space

$$D(a, b) = \|a - b\|_2 \rightarrow \text{Euclidean distance}$$

$$D_{i,j} := D(f_\theta(x_i), f_\theta(x_j))$$

Triplet Loss

$$\mathcal{L}_{\text{tri}}(\theta) = \sum_{\substack{a,p,n \\ y_a = y_p \neq y_n}} [m + D_{a,p} - D_{a,n}]_+$$

$x_a \rightarrow \text{anchor}$, $x_p \rightarrow \text{positive}$ (same class/person), $x_n \rightarrow \text{negative}$ (another class)



Mining hard triplets is crucial for learning!

It is common to only mine moderate negatives and/or moderate positives.

Batch Hard Loss

$$\mathcal{L}_{\text{BH}}(\theta; X) = \sum_{i=1}^P \sum_{a=1}^K \left[m + \underbrace{\max_{p=1 \dots K} D(f_\theta(x_a^i), f_\theta(x_p^i))}_{\text{hardest positive}} - \underbrace{\min_{\substack{n=1 \dots K \\ j \neq i}} D(f_\theta(x_a^i), f_\theta(x_n^j))}_{\text{hardest negative}} \right]_+$$

$X \rightarrow \text{mini-batch}$

$x_j^i \rightarrow j\text{-th image of the } i\text{-th person in the batch}$

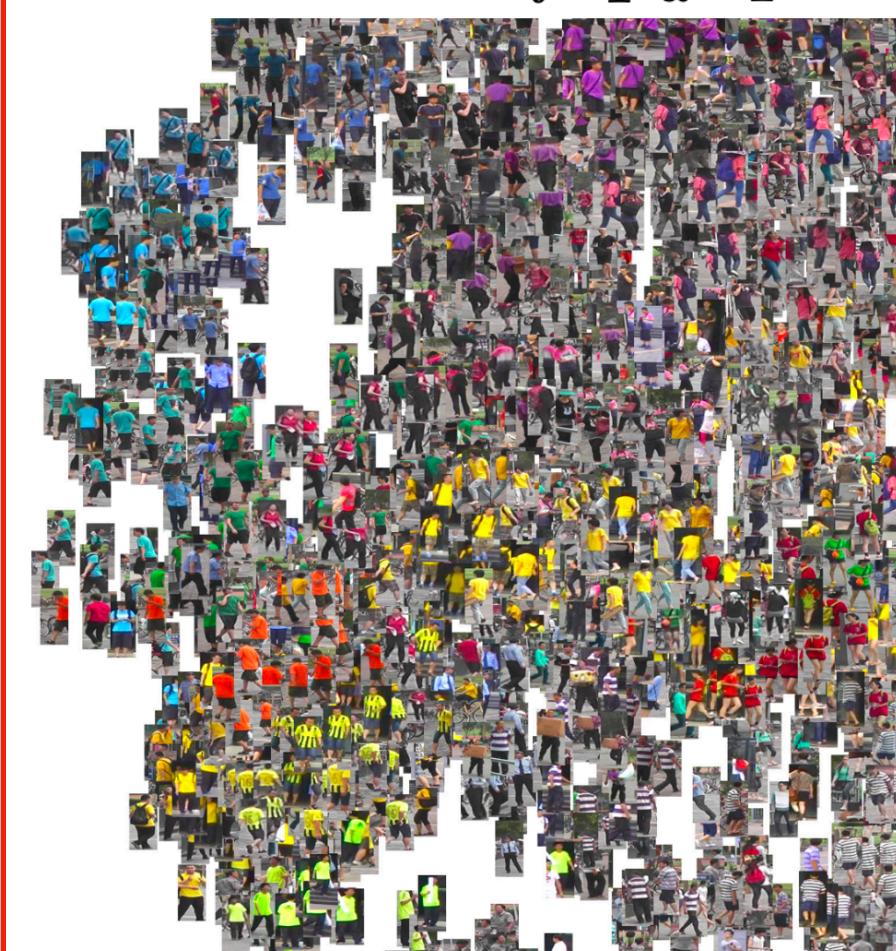
$P \rightarrow \text{number of classes (person identities) in a batch}$

$K \rightarrow \text{number of images of each class (person)}$

$PK \rightarrow \text{batch size}$
For each sample a in the batch, select the hardest positive and the hardest negative samples within the batch.

Generalized Lifted Embedding Loss

$$\mathcal{L}_{\text{LG}}(\theta; X) = \sum_{i=1}^P \sum_{a=1}^K \left[\log \underbrace{\sum_{\substack{p=1 \\ p \neq a}}^K e^{D(f_\theta(x_a^i), f_\theta(x_p^i))}}_{\text{all anchors}} + \log \underbrace{\sum_{\substack{j=1 \\ j \neq i}}^P \sum_{n=1}^K e^{m - D(f_\theta(x_a^i), f_\theta(x_n^j))}}_{\text{all negatives}} \right]_+ + \log \underbrace{\sum_{n=1}^K e^{m - D(f_\theta(x_a^i), f_\theta(x_n^i))}}_{\text{all positives}}$$



A small crop of the Barnes-Hut t-SNE of the learned embeddings for the Market-1501 test-set. The triplet loss learns semantically meaningful features.

It is possible to replace the hinge function by a smooth approximation using the softplus function $\ln(1 + \exp(\cdot))$

MARS validation split	margin 0.1		margin 0.2		margin 0.5		margin 1.0		soft margin	
	mAP	rank-1								
Triplet (\mathcal{L}_{tri})	40.80	59.23	41.71	60.78	43.51	60.87	43.61	61.63	48.40	66.37
Triplet (\mathcal{L}_{tri}) + OHM	16.6*	36.6*	61.40	82.95	32.0*	57.1*	41.45	59.42	46.63	65.43
Batch hard (\mathcal{L}_{BH})	65.09	83.51	65.27	84.55	65.12	83.39	63.78	82.48	65.77	84.69
Batch hard ($\mathcal{L}_{\text{BH} \neq 0}$)	63.10	83.04	64.19	83.42	63.71	82.29	64.06	84.50	-	-
Batch all (\mathcal{L}_{BA})	59.43	79.24	60.48	79.99	60.30	79.52	62.08	80.55	61.04	80.65
Batch all ($\mathcal{L}_{\text{BA} \neq 0}$)	63.29	83.65	64.31	83.37	64.41	83.98	64.06	82.90	-	-
Lifted 3-pos. (\mathcal{L}_{LG})	64.00	82.71	63.87	82.86	63.61	84.55	64.02	84.17	-	-
Lifted 1-pos. (\mathcal{L}_{L}) [32]	61.95	81.35	63.68	81.73	63.01	82.48	62.28	82.34	-	-

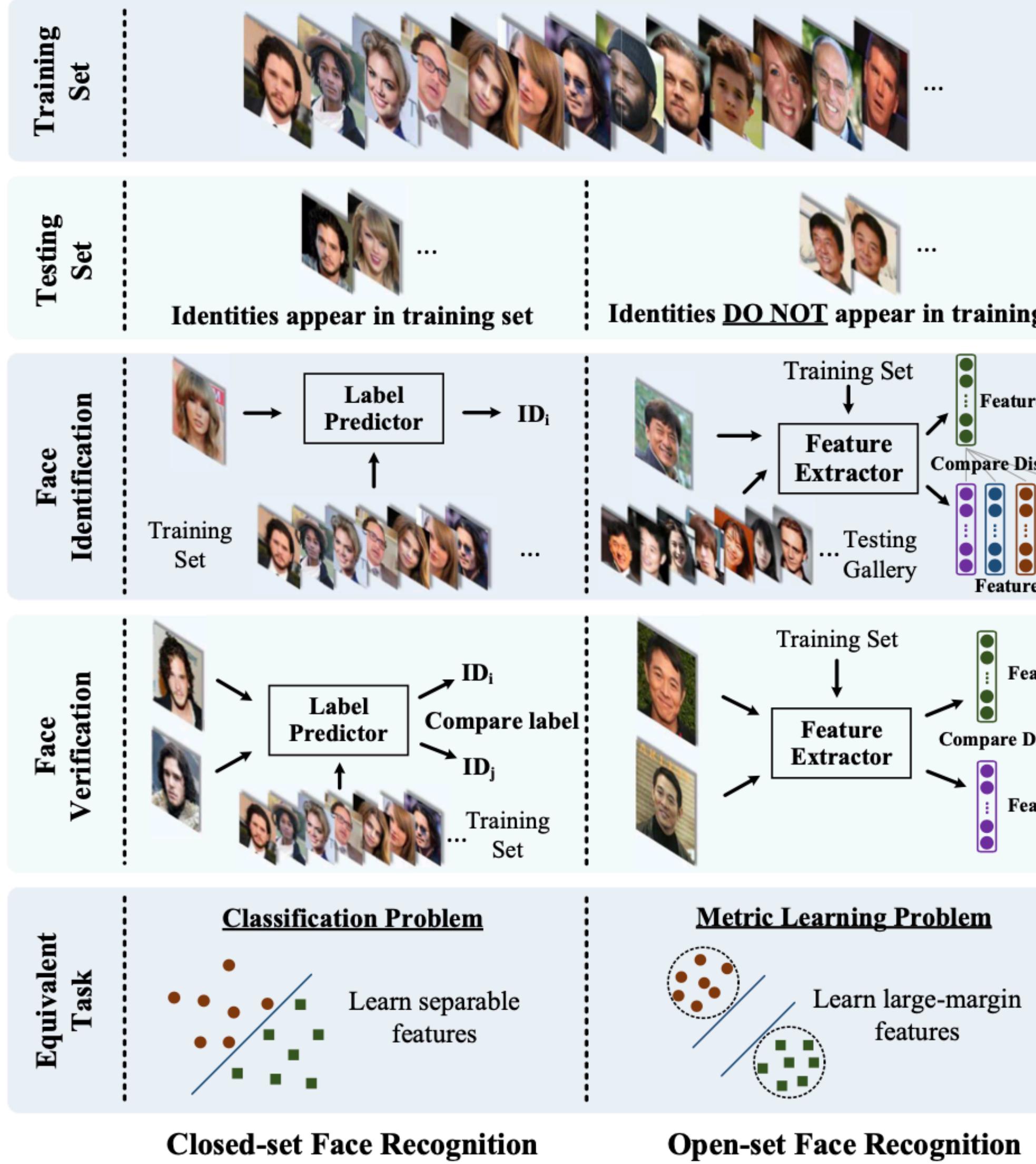


Boulder

SphereFace: Deep Hypersphere Embedding for Face Recognition



[YouTube Video](#)



Revisiting the Softmax Loss

$$L = \frac{1}{N} \sum_i L_i \quad \theta_{j,i} \rightarrow \text{angle between } W_j \text{ and } x_i$$

$$L_i = -\log \left(\frac{e^{\|W_{y_i}\|^T x_i + b_{y_i}}}{\sum_j e^{\|W_j\|^T x_i + b_j}} \right) = -\log \left(\frac{e^{\|W_{y_i}\| \|x_i\| \cos(\theta_{y_i,i}) + b_{y_i}}}{\sum_j e^{\|W_j\| \|x_i\| \cos(\theta_{j,i}) + b_j}} \right)$$

$$\text{Set } \|W_j\| = 1 \text{ and } b_j = 0$$

$$L_{\text{modified}} = \frac{1}{N} \sum_i -\log \left(\frac{e^{\|x_i\| \cos(\theta_{y_i,i})}}{\sum_j e^{\|x_i\| \cos(\theta_{j,i})}} \right)$$

- CASIA-WebFace dataset (Training)
- LFW: Labeled Face in the Wild (Testing)
- YTF: Youtube Faces (Testing)
- MegaFace Challenge 1 (Testing)

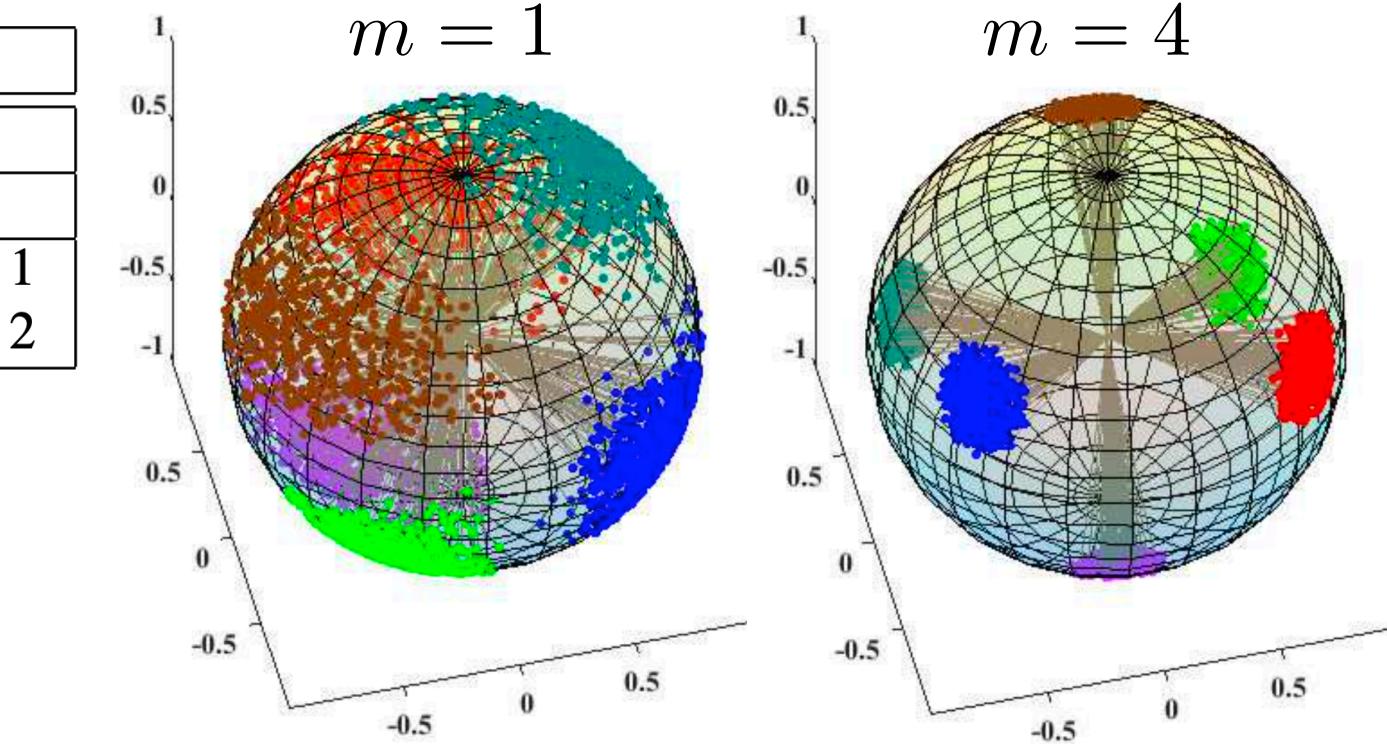
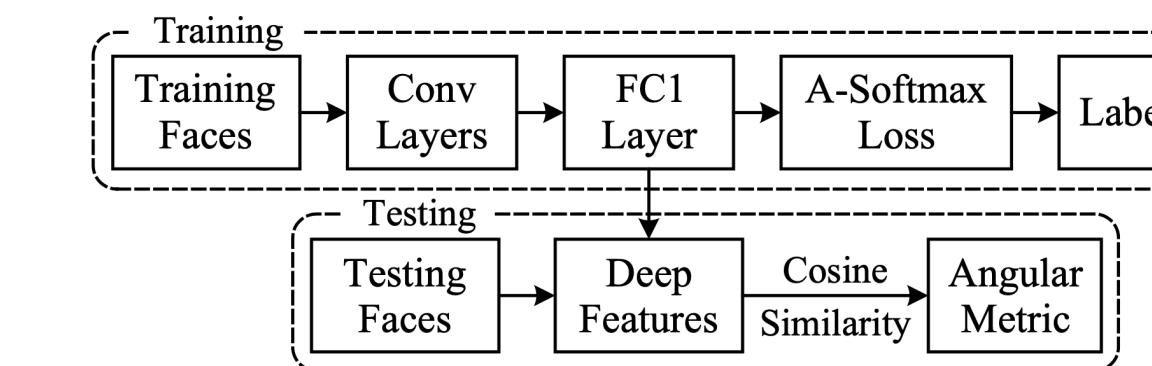
Angular Softmax (A-Softmax) loss

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

$$\psi(\theta_{y_i,i}) = (-1)^k \cos(m\theta_{y_i,i}) - 2k \quad \text{for } \theta_{y_i,i} \in [\frac{k\pi}{m}, \frac{(k+1)\pi}{m}] \text{ and } k \in [0, m-1]$$

monotonically decreasing angle function

Loss Function	Decision Boundary
Softmax Loss	$(W_1 - W_2)x + b_1 - b_2 = 0$
Modified Softmax Loss	$\ x\ (\cos \theta_1 - \cos \theta_2) = 0$
A-Softmax Loss	$\ x\ (\cos m\theta_1 - \cos \theta_2) = 0 \text{ for class 1}$ $\ x\ (\cos \theta_1 - \cos m\theta_2) = 0 \text{ for class 2}$





Boulder



[YouTube Video](#)

ArcFace: Additive Angular Margin Loss for Deep Face Recognition

- softmax classifier
- triplet loss

softmax drawbacks

- size of the linear transformation matrix $W \in \mathbb{R}^{d \times n}$ increases linearly with the identities number n
- good for closed-set classification problems but not discriminative enough for the open-set face recognition problem

triplet loss drawbacks

- combinatorial explosion in the number of face triplets
- semi-hard-example-mining is a quite difficult problem for efficient model training

$$L_1 = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^n e^{W_j^T x_i + b_j}}$$

softmax loss

$x_i \in \mathbb{R}^d \rightarrow$ deep feature of the i -th sample

$y_i \rightarrow$ class of the i -th sample

$W_j \in \mathbb{R}^d \rightarrow j$ -th column of $W \in \mathbb{R}^{d \times n}$

$b_j \in \mathbb{R}^n \rightarrow$ bias

N & $n \rightarrow$ batch size & class number

set $b_j = 0$

$$W_j^T x_i = \|W_j\| \|x_i\| \cos \theta_j$$

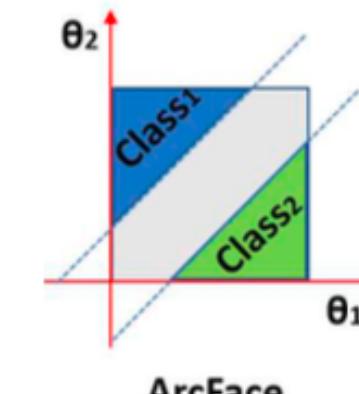
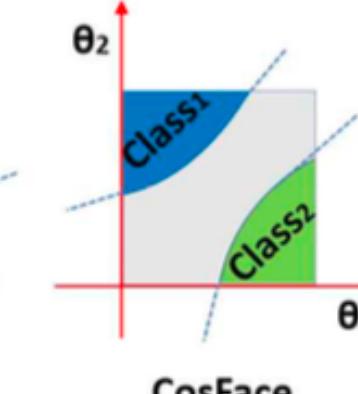
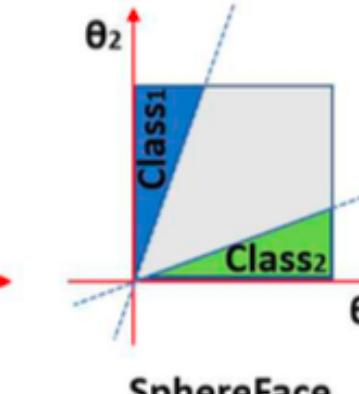
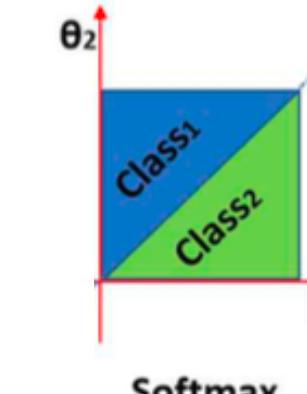
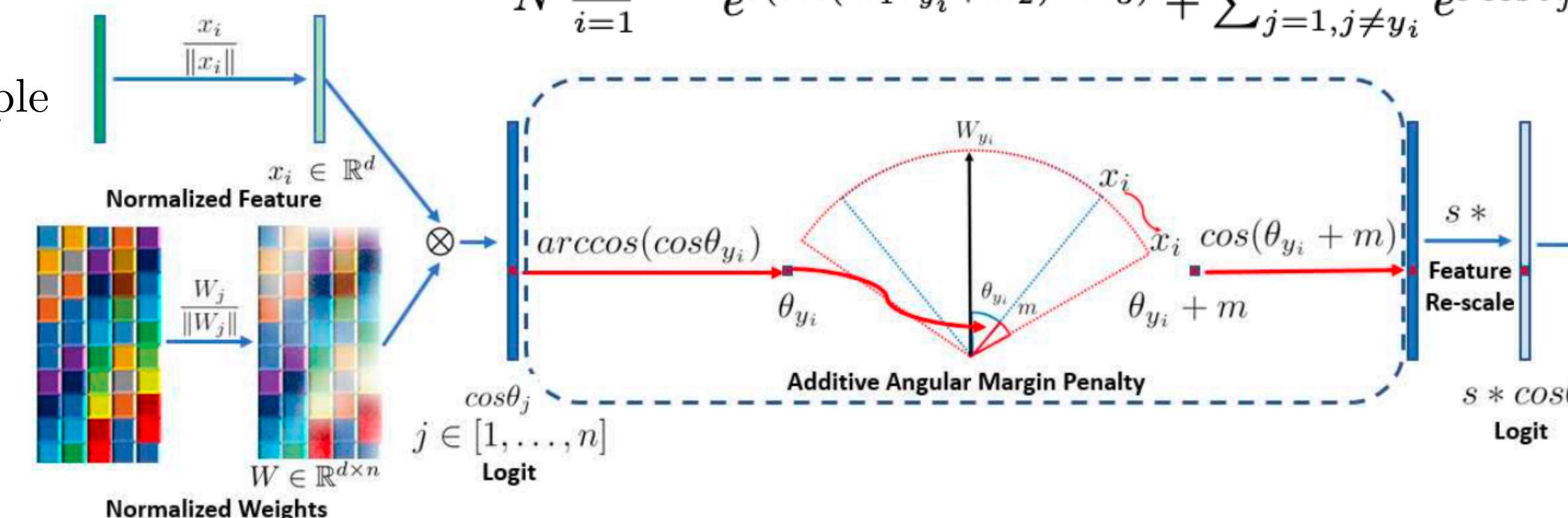
$\|W_j\| = 1$ by l_2 normalization
 $\|x_i\| = s$ by l_2 normalization
 \curvearrowleft hyper-sphere with radius s

$$L_2 = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s \cos \theta_{y_i}}}{e^{s \cos \theta_{y_i}} + \sum_{j=1, j \neq y_i}^n e^{s \cos \theta_j}}$$

$$L_3 = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s(\cos(\theta_{y_i} + m))}}{e^{s(\cos(\theta_{y_i} + m))} + \sum_{j=1, j \neq y_i}^n e^{s \cos \theta_j}}$$

$m \rightarrow$ additive angular margin penalty between x_i and W_{y_i} to simultaneously enhance the intra-class compactness and inter-class discrepancy

$$L_4 = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s(\cos(m_1 \theta_{y_i} + m_2) - m_3)}}{e^{s(\cos(m_1 \theta_{y_i} + m_2) - m_3)} + \sum_{j=1, j \neq y_i}^n e^{s \cos \theta_j}}$$

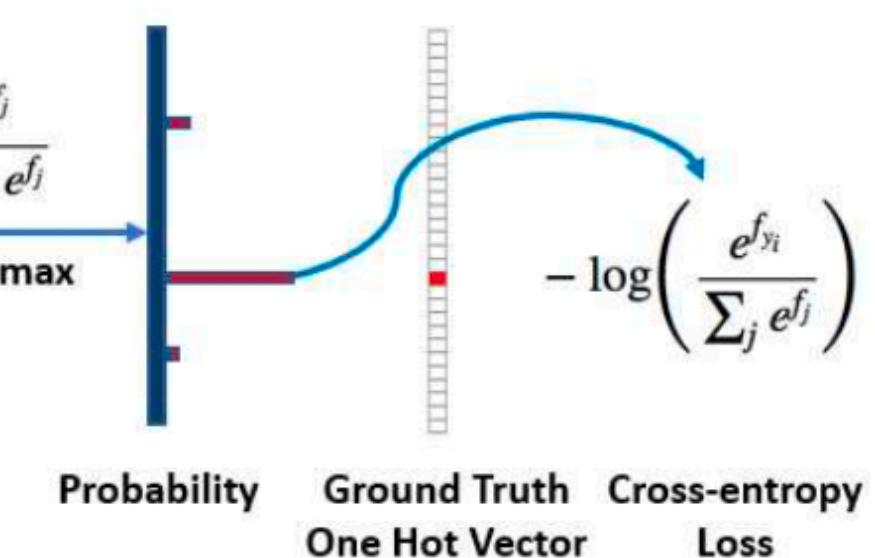
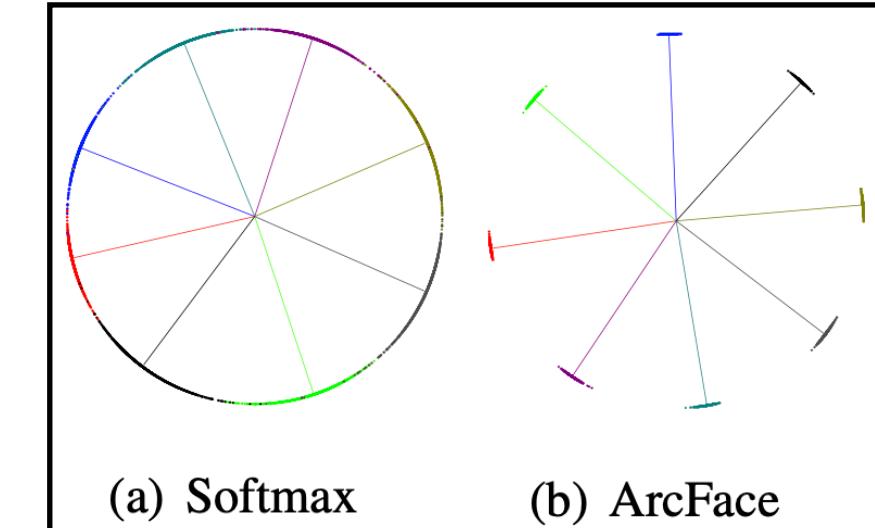


triplet-loss $\leftarrow \arccos(x_i^{pos} x_i) + m \leq \arccos(x_i^{neg} x_i)$

$$\text{Intra-Loss} \leftarrow L_5 = L_2 + \frac{1}{\pi N} \sum_{i=1}^N \theta_{y_i}$$

$$L_6 = L_2 - \frac{1}{\pi N(n-1)} \sum_{i=1}^N \sum_{j=1, j \neq y_i}^n \arccos(W_{y_i}^T W_j)$$

Inter-Loss





Boulder



Questions?

[YouTube Playlist](#)
