Face verification

Papers:

- VGGFace2: A dataset for recognising faces across pose and age
- A Discriminative Feature Learning Approach for Deep Face Recognition
- FaceNet: A Unified Embedding for Face Recognition and Clustering
- SphereFace: Deep Hypersphere Embedding for Face Recognition
- ArcFace: Additive Angular Margin Loss for Deep Face Recognition

General Concept

- **Softmax:** only learns separable features that are not discriminative enough.
  - Softmax + **contrastive loss / center loss**
- **Triplet Loss:** supervise the embedding learning.
  - Center loss: explicitly encourages intra-class compactness.
- **Euclidean margin based loss + softmax:** joint supervision
  - In some sense Euclidean and softmax re incompatible
- **Angular margin:**
- **CosFace:**
- **ArcFace:**

General Concept

- **Face recognition** can be categorized as face identification and fae verification.
- **Face recognition** can be evaluated under close-set or open-set settings.
- **Close-set:** all the testing identities are predefined in the training set. (classification)
- **Open-set:** the testing identities are usually disjoint from the training set. (Discriminative)

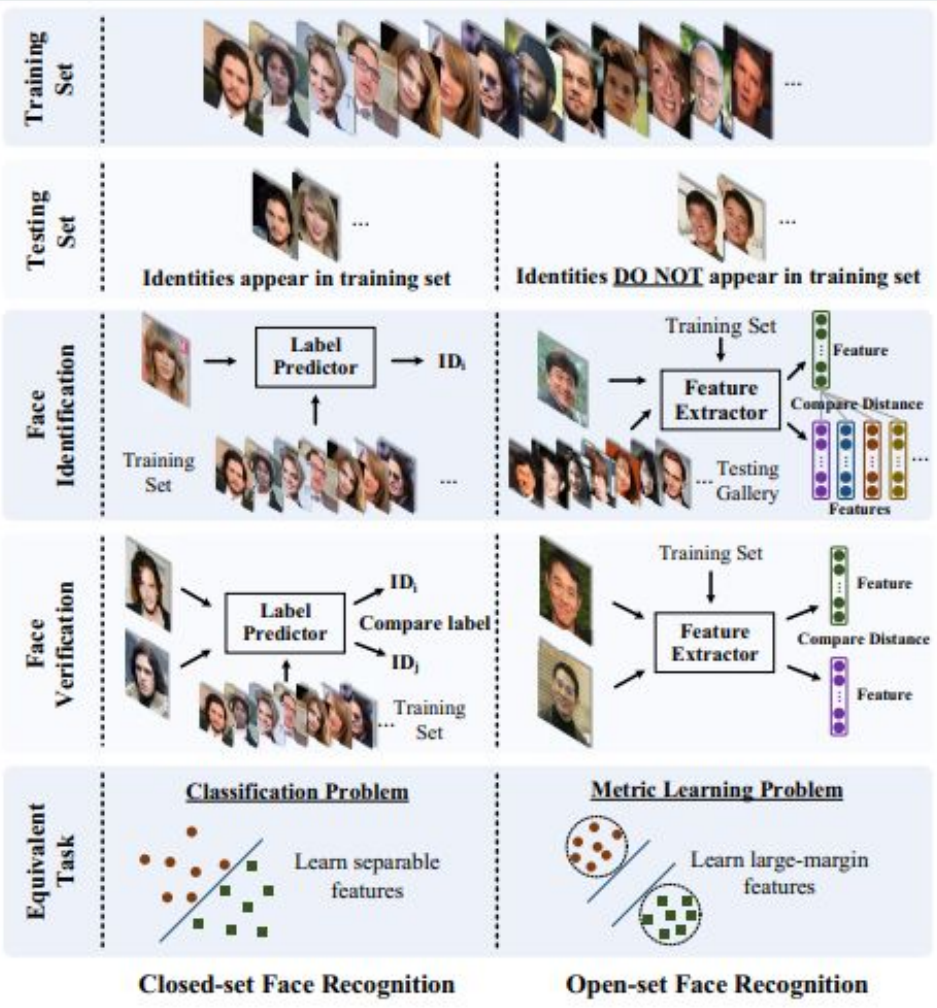


Figure 1: Comparison of open-set and closed-set face recognition.

VGGFace2

A dataset for recognising faces across pose and age.

VGGFace2

- The dataset contains 3.31 million images of 9131 subjects, with an average of 362.6 images for each subject.
  - variations in pose, age, illumination, ethnicity and profession (e.g. actors, athletes, politicians).
- The dataset was collected with three goals in mind:
  - large number of identities + large number of images for each identity;

Datasets	# of subjects	# of images	# of images per subject	manual identity labelling	pose	age	year
LFW [10]	5,749	13,233	1/2.3/530	-	-	-	2007
YTF [23]	1,595	3,425 videos	-	-	-	-	2011
CelebFaces+ [20]	10,177	202,599	19.9	-	-	-	2014
CASIA-WebFace [25]	10,575	494,414	2/46.8/804	-	-	-	2014
IJB-A [13]	500	5,712 images, 2,085 videos	11.4	-	-	-	2015
IJB-B [22]	1,845	11,754 images, 7,011 videos	36.2	-	-	-	2017
VGGFace [16]	2,622	2.6 M	1,000/1,000/1,000	-	-	Yes	2015
MegaFace [12]	690,572	4.7 M	3/7/2469	-	-	-	2016
MS-Celeb-1M [7]	100,000	10 M	100	-	-	-	2016
UMDFaces [5]	8,501	367,920	43.3	Yes	Yes	Yes	2016
UMDFaces-Videos [4]	3,107	22,075 videos	-	-	-	-	2017
VGGFace2 (**this paper**)	9,131	3.31 M	80/362.6/843	Yes	Yes	Yes	2018

VGGFace2 - Dataset Collection

A. Stages:

1. Obtaining and selecting a name list
   - candidates with insufficient images
   - Attribute information such as ethnicity and kinship is obtained from DBPedia
2. Obtaining images for each identity
   - Downloaded 1k images for each subject.
   - Age variation (sideview - 200, very young - 200) = 1400 images
3. Face detection
   - Face detection extended by factor of 0.3 for a better trade-off between precision and recall.

Stage	Aim	Type	# of subject	total # of images	Annotation effort
1	Name list selection	M	500K	50.00 million	3 months
2	Image downloading	A	9244	12.94 million	-
3	Face detection	A	9244	7.31 million	-
4	Automatic filtering by classification	A	9244	6.99 million	-
5	Near duplicate removal	A	9244	5.45 million	-
6	Final automatic and manual filtering	A/M	9131	3.31 million	21days

VGGFace2 - Dataset Collection

4. Automatic filtering by classification: to remove outlier faces for each identity automatically
    - The top 100 retrieved images of each identity are used as positives,
    - and the top 100 of all other identities are used as negative for training.
    - Removing images under the threshold of 0.5
5. Near duplicate removal
    - Near duplicate images
6. Final automatic and manual filtering
    - Existent errors:
        i. Outliers
        ii. Face mixtures

- Detecting overlapped subjects.
  - Subject overlapping:’ Will I Am’ & ‘William’
  - Noisy classes
  - Subjects with less samples
- Removing outlier images for a subject.
  - Resulted in purity of 96%
  - retrain the model based on a dataset classified into three sets
    - H (high score range [1, 0.95]))
    - I (intermediate score range (0.95, 0.8])
    - L (low score range (0.8, 0.5])
- Pose and age annotations
  - Training two networks
    - Head pose (roll, pitch, yaw)
    - Apparent age.

EXPERIMENTS -

Training dataset	VGGFace		MS1M		VGGFace2	
	young	mature	young	mature	young	mature
young	0.5231	0.4338	0.4983	0.4005	0.6256	0.5524
mature	0.4394	0.5518	0.4099	0.5276	0.5607	0.6637

Table V: Face probing across ages. Similarity scores are evaluated across age templates. A higher value is better.

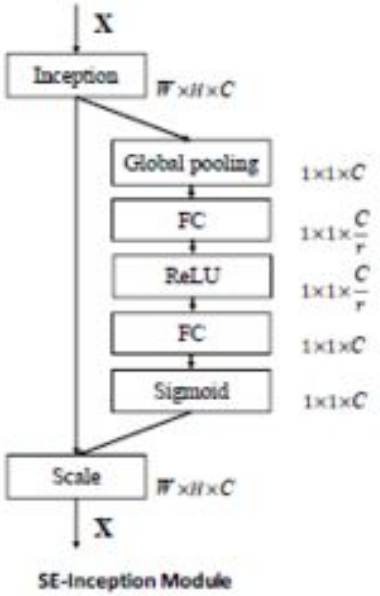
- **Experimental setup**
  - ResNet-50 and SE-RestNet-50 are used as the backbone architectures
    - The Squeeze-and-Excitation (SE) blocks [9] adaptively recalibrate channel-wise feature responses by explicitly modelling channel relationships
  - Networks are learned from scratch - VGGFace, Ms-Celeb-1M, and VGGFace2
  - pre-trained on Ms-Celeb-1M, and fine-tuned on VGGFace2
- **Experiments on the new dataset**
- **Experiments on IJB-A**

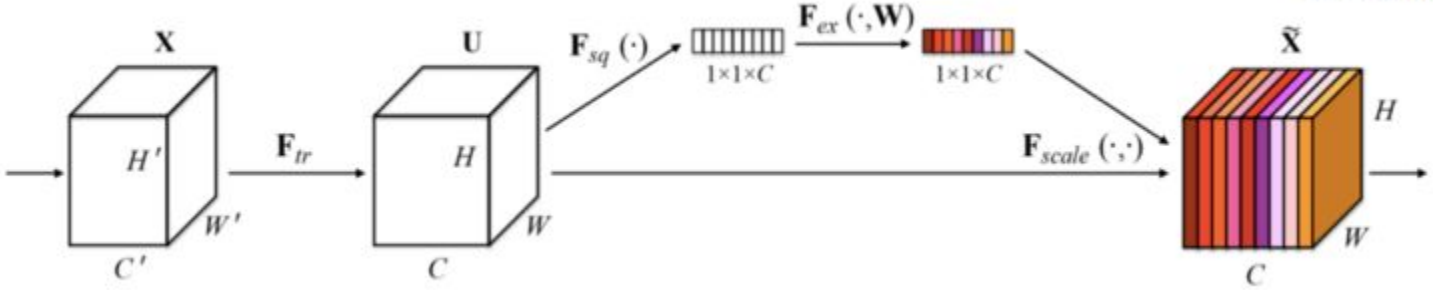
Training dataset	VGGFace			MS1M			VGGFace2		
	front	three-quarter	profile	front	three-quarter	profile	front	three-quarter	profile
front	0.5781	0.5679	0.4821	0.5661	0.5582	0.4715	0.6876	0.6821	0.6222
three-quarter	0.5706	0.5957	0.5345	0.5628	0.5766	0.5036	0.6859	0.6980	0.6481
profile	0.4859	0.5379	0.5682	0.4776	0.5064	0.5094	0.6264	0.6515	0.6488

Table IV: Face probing across poses. Similarity scores are evaluated across pose templates. A higher value is better.

Squeeze-and-Excitation networks: (SENets)

- [illegible] learity -





A Discriminative Feature Learning Approach for Deep Face Recognition

Softmax loss + center loss: inter-class dispensation and intra-class compactness as much as possible
The CNNs are trained under the supervision of the softmax loss and center loss, with a hyper parameter to balance the two supervision signals.

Center loss

- Used to enhance the discriminative of deeply learned features.
- Trainable and easy to optimize.
- Simultaneously learns the center and Penalized the distances
- Efficiently pulls the deep features of the same class to their centers.
- Minimize the intra-class distance of the deep features.

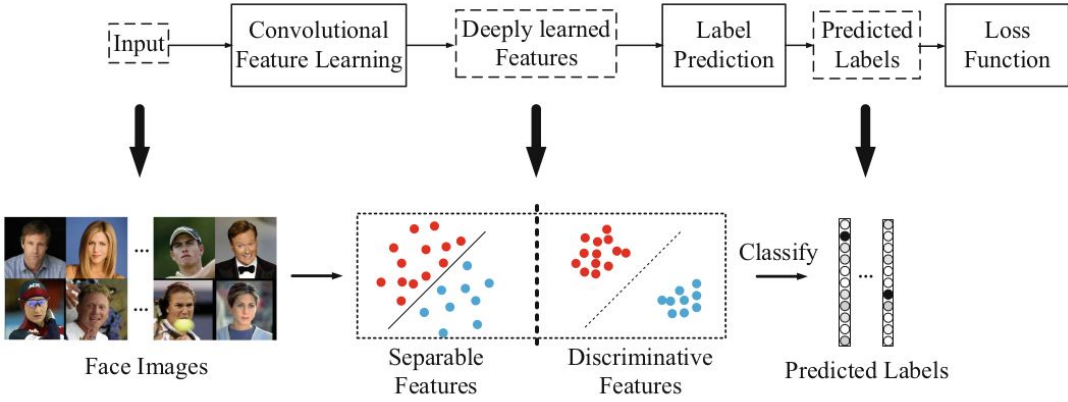


Fig. 1. The typical framework of convolutional neural networks.

- **Softmax pool** only encourages the separability of features.
- **Contrastive loss and triplet loss:** increases the computational complexity due to growing of triplets.
- **Center loss:** Same requirement of Softmax pool, needs no complex recombination of the training samples

Softmax Loss

- A toy example on MINST

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

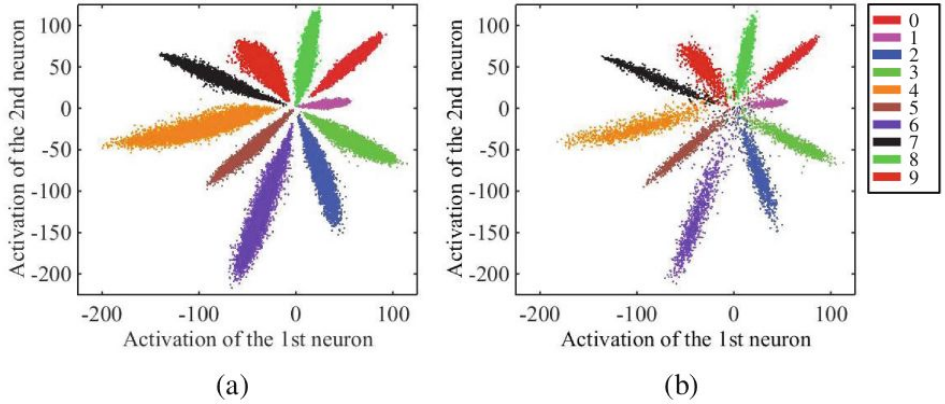


Fig. 2. The distribution of deeply learned features in (a) training set (b) testing set, both under the supervision of softmax loss, where we use 50K/10K train/test splits. The points with different colors denote features from different classes. **Best viewed in color.** (Color figure online)

Table 1. The CNNs architecture we use in toy example, called LeNets++. Some of the convolution layers are followed by max pooling. $(5, 32)_{/1,2} \times 2$ denotes 2 cascaded convolution layers with 32 filters of size 5×5, where the stride and padding are 1 and 2 respectively. $2_{/2,0}$ denotes the max-pooling layers with grid of 2×2, where the stride and padding are 2 and 0 respectively. In LeNets++, we use the Parametric Rectified Linear Unit (PReLU) [12] as the nonlinear unit.

	Stage 1		Stage 2		Stage 3		Stage 4
Layer	Conv	Pool	Conv	Pool	Conv	Pool	FC
LeNets	$(5, 20)_{/1,0}$	$2_{/2,0}$	$(5, 50)_{/1,0}$	$2_{/2,0}$			500
LeNets++	$(5, 32)_{/1,2} \times 2$	$2_{/2,0}$	$(5, 64)_{/1,2} \times 2$	$2_{/2,0}$	$(5, 128)_{/1,2} \times 2$	$2_{/2,0}$	**2**

Center Loss

- Instead of updating the centers with the respect to the entire training set, we perform the update based on the mini-batch.
- To avoid the large perturbation caused by few mislabeled samples, we use a scaler ***α*** to control the learning rate.
- Center loss are trainable and can be optimized by standard SGD.
- A scalar λ is used for balancing the two loss function
- **Joint loss**

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

$$\frac{\partial \mathcal{L}_C}{\partial \boldsymbol{x}_i} = \boldsymbol{x}_i - \boldsymbol{c}_{y_i}$$

$$\Delta \boldsymbol{c}_j = \frac{\sum_{i=1}^{m} \delta(y_i = j) \cdot (\boldsymbol{c}_j - \boldsymbol{x}_i)}{1 + \sum_{i=1}^{m} \delta(y_i = j)}$$

δ = 1 condition satisfied
δ = 0 otherwise

α is restricted in [0,1]

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

$$= -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T \boldsymbol{x}_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T \boldsymbol{x}_i + b_j}} + \frac{\lambda}{2}\sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2^2$$

Joint Supervision

-

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

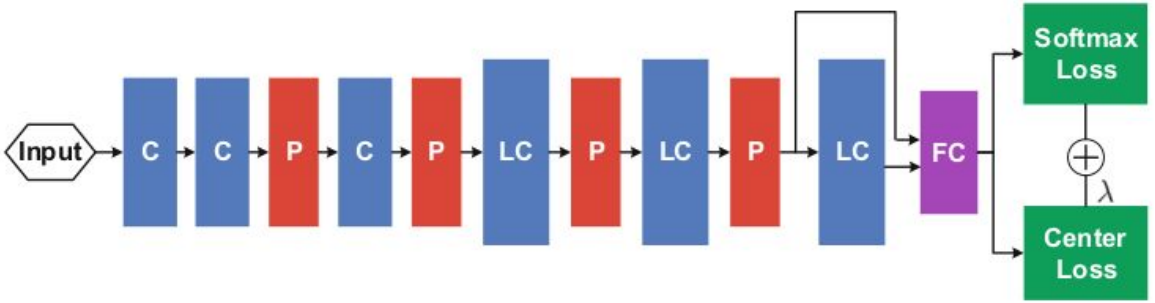


Fig. 4. The CNN architecture using for face recognition experiments. Joint supervision is adopted. The filter sizes in both convolution and local convolution layers are 3×3 with stride 1, followed by PReLU [12] nonlinear units. Weights in three local convolution layers are locally shared in the regions of 4×4, 2×2 and 1×1 respectively. The number of the feature maps are 128 for the convolution layers and 256 for the local convolution layers. The max-pooling grid is 2×2 and the stride is 2. The output of the 4th pooling layer and the 3th local convolution layer are concatenated as the input of the 1st fully connected layer. The output dimension of the fully connected layer is 512. **Best viewed in color.** (Color figure online)

λ influence

Joint Supervision

- **Softmax loss:** deeply learned features contain large intra-class variation
- **Center loss:** deeply learned features and center degrad to zeros.

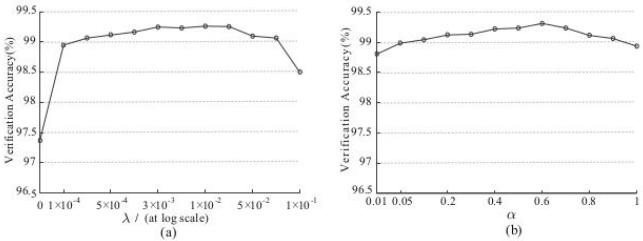


Fig. 5. Face verification accuracies on LFW dataset, respectively achieve by (a) models with different λ and fixed $\alpha = 0.5$. (b) models with different α and fixed $\lambda = 0.003$.

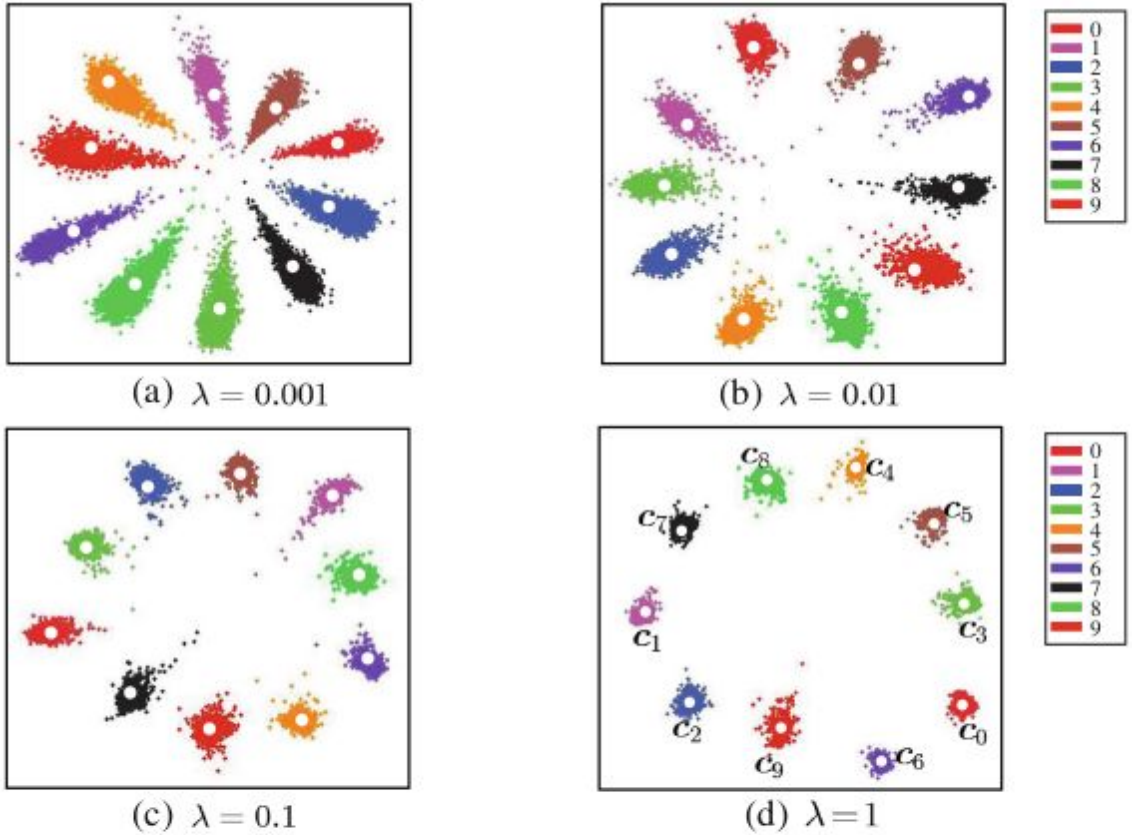


Fig. 3. The distribution of deeply learned features under the joint supervision of softmax loss and center loss. The points with different colors denote features from different classes. Different λ lead to different deep feature distributions ($\alpha = 0.5$). The white dots ($\boldsymbol{c}_0$, $\boldsymbol{c}_1$,...,$\boldsymbol{c}_9$) denote 10 class centers of deep features. **Best viewed in color.** (Color figure online)

Joint Supervision Result

Table 2. Verification performance of different methods on LFW and YTF datasets

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 %**

Joint Supervision Result

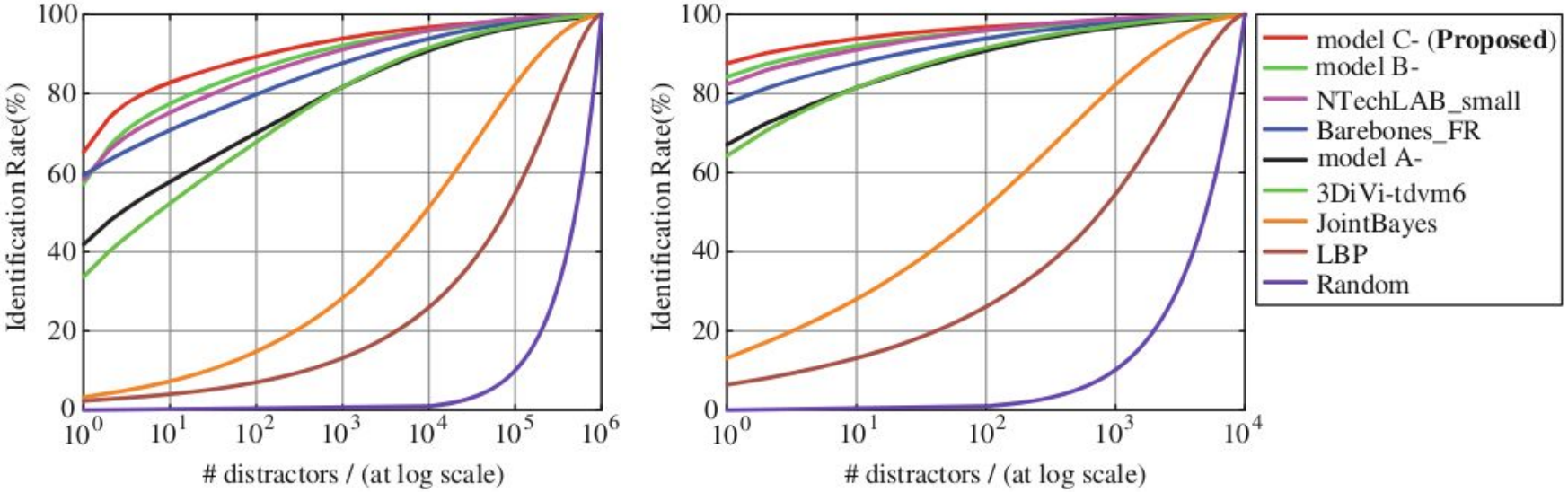


Fig. 8. CMC curves of different methods (under the protocol of small training set) with (a) 1M and (b) 10 K distractors on Set 1. The results of other methods are provided by MegaFace team.

FaceNet

Directly learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity.

FaceNet

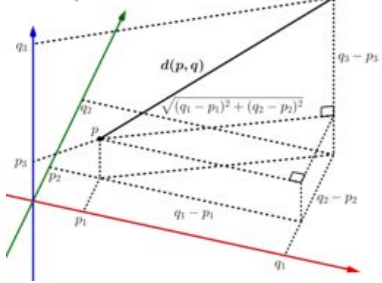
directly learns a mapping from face images to a compact Euclidean space (L2) where distances directly correspond to a measure of face similarity

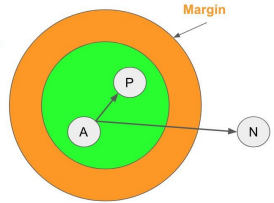
- Recognition becomes a k-NN classification
- Clustering can be achieved using k-means or agglomerative clustering.

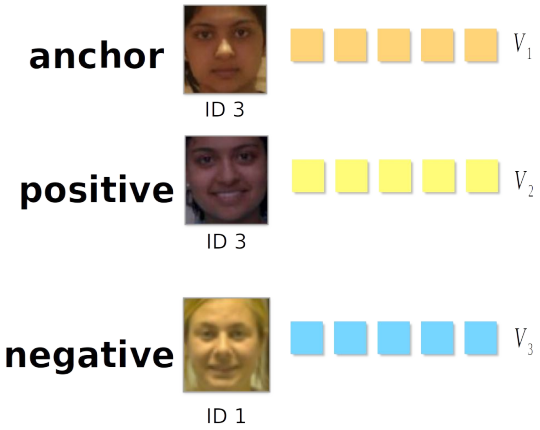
Trained on a deep convolutional network to directly optimize the embedding itself, rather than an intermediate bottleneck layer

- using **triplets**
  - roughly aligned matching / non-matching face patches generated using a novel online triplet mining method

$$d(p,q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}.$$

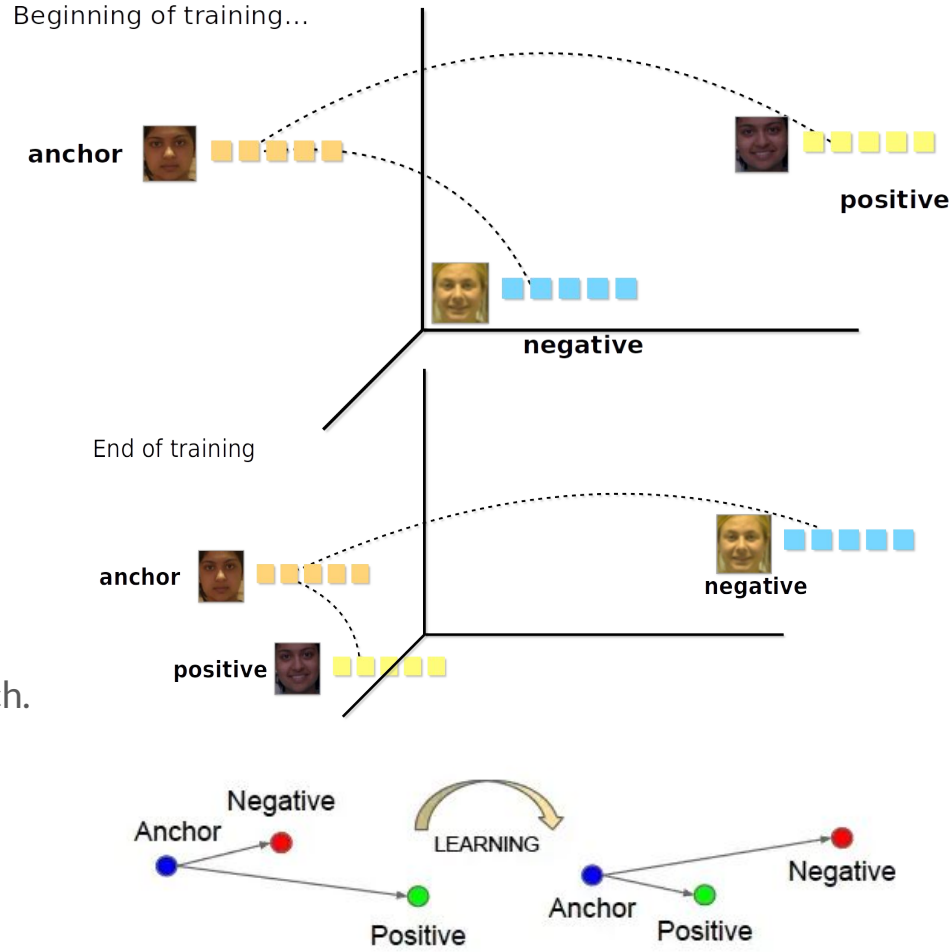






Triplets

- Introduced negative exemplar mining strategy
- compact 128-D embedding using a triplet-based loss function
- The thumbnails are tight crops of the face area
  - scale and translation is performed.
  - no 2D or 3D alignment
- Triplet Loss function + Triplet mining.
- Matching (positive - negative):
  - Hard mining triplet selection approach.
  - Semi Hard triplet mining.
    - Data labeling issue,
    - Mini batch training



Architecture

1. Euclidean Space
2. Triplet Loss Function
   - Hard mining triplet selection approach.
3. Semi Hard triplet mining.
   - Data labeling issue,
   - Mini batch training

➔ K-fold training:

- 200k images
- 100k x 100k image pair

➔ using fully end-to-end network.

➔ CNN SGD + AdaGrad

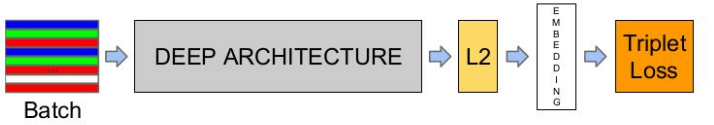


Figure 2. **Model structure.** Our network consists of a batch input layer and a deep CNN followed by L_2 normalization, which results in the face embedding. This is followed by the triplet loss during training.

$$\|x_i^a - x_i^p\|_2^2 + \alpha < \|x_i^a - x_i^n\|_2^2, \ \forall (x_i^a, x_i^p, x_i^n) \in \mathcal{T} . \quad (1)$$

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

Argmax $|| f(x_i^a) - f(x_i^p) ||_2^2$ -- Eq(1)

Argmin $|| f(x_i^a) - f(x_i^n) ||_2^2$ -- Eq(2)

$$\|f(x_i^a) - f(x_i^p)\|_2^2 < \|f(x_i^a) - f(x_i^n)\|_2^2 . \quad (3)$$

$$\mathrm{TA}(d) = \{(i,j) \in \mathcal{P}_{\text{same}}, \text{with } D(x_i, x_j) < d\} . \quad (4)$$

$$\mathrm{FA}(d) = \{(i,j) \in \mathcal{P}_{\text{diff}}, \text{with } D(x_i, x_j) \leq d\} \quad (5)$$

$$\mathrm{VAL}(d) = \frac{|\mathrm{TA}(d)|}{|\mathcal{P}_{\text{same}}|}, \quad \mathrm{FAR}(d) = \frac{|\mathrm{FA}(d)|}{|\mathcal{P}_{\text{diff}}|} . \quad (6)$$

FaceNet

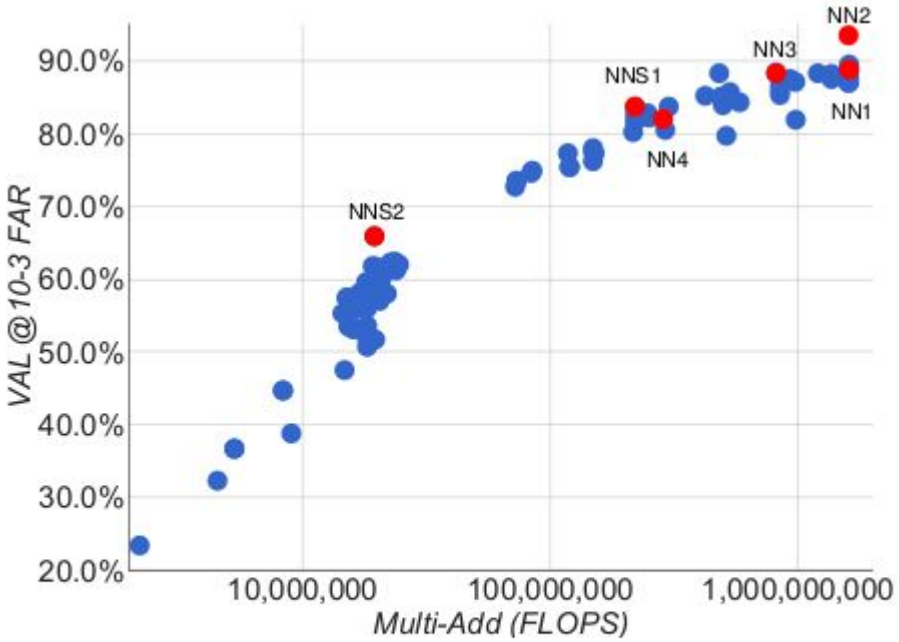


Figure 4. **FLOPS *vs.* Accuracy trade-off.** Shown is the trade-off between FLOPS and accuracy for a wide range of different model sizes and architectures. Highlighted are the four models that we focus on in our experiments.

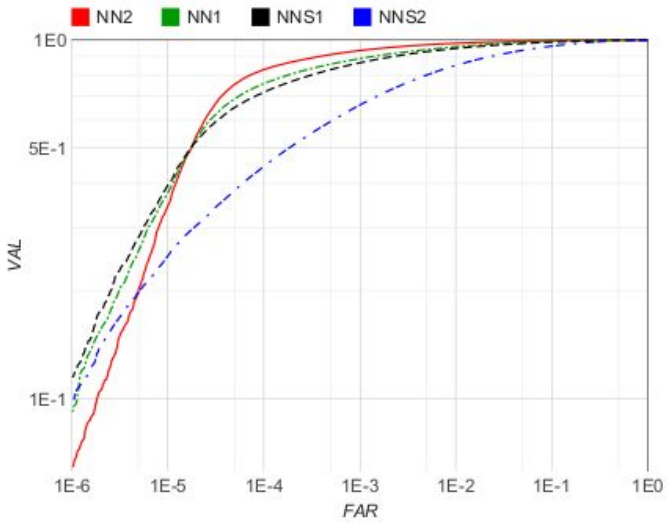


Figure 5. **Network Architectures.** This plot shows the complete ROC for the four different models on our personal photos test set from section 4.2. The sharp drop at 10E-4 FAR can be explained by noise in the groundtruth labels. The models in order of performance are: **NN2:** 224×224 input Inception 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.

Result

Performance:

- Youtube Faces DB
  - classification accuracy of 95.12% ±0.39
- LFW
  - classification accuracy of 98.87% ±0.15

Figure 7. **Face Clustering.** Shown is an exemplar cluster for one user. All these images in the users personal photo collection were clustered together.

Figure 6. **LFW errors.** This shows all pairs of images that were incorrectly classified on LFW.

SphereFace: Deep Hypersphere Embedding for Face Recognition

deep face recognition (FR) problem under open-set protocol, where ideal face features are expected to have smaller maximal intra-class distance than minimal inter-class distance under a suitably chosen metric space

SphareFace

- angular margin
- Transform feature space into hypersphere and compute the distances as the angles between the feature vectors.
- Angular margin directly links to discriminative on a manifold.
- Each pixel is normalized by subtracting 127.5 and then being divided by 128.

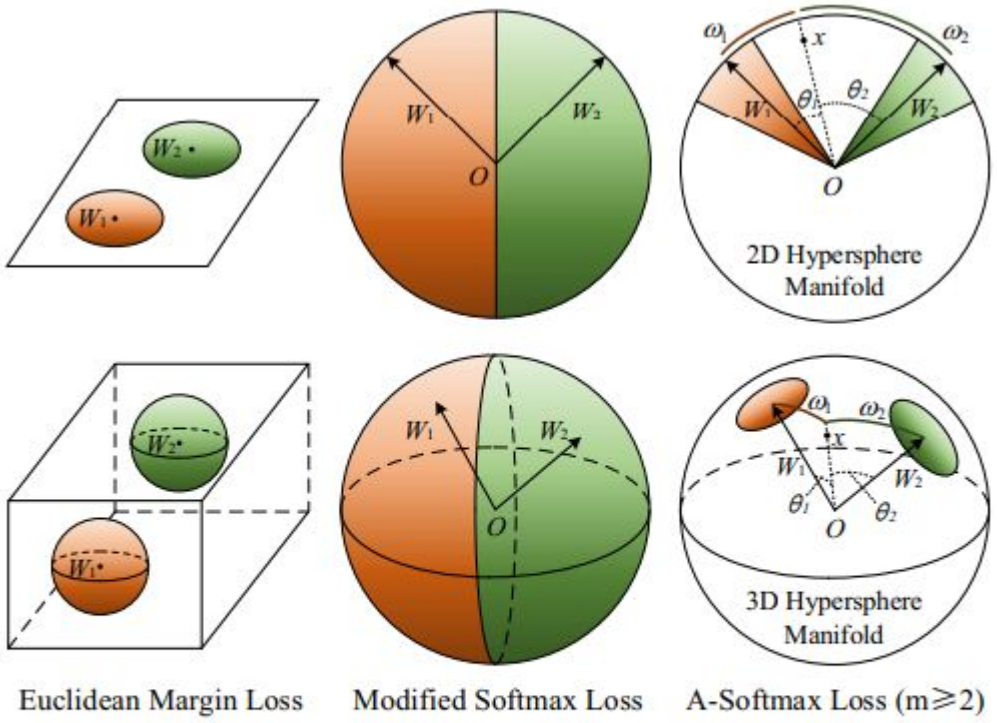


Figure 3: Geometry Interpretation of Euclidean margin loss (e.g. contrastive loss, triplet loss, center loss, etc.), modified softmax loss and A-Softmax loss. The first row is 2D feature constraint, and the second row is 3D feature constraint. The orange region indicates the discriminative constraint for class 1, while the green region is for class 2.

Softmax loss - A Softmax loss

$$L = \frac{1}{N}\sum_i L_i = \frac{1}{N}\sum_i -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right) \tag{3}$$

$$(\boldsymbol{W}_1 - \boldsymbol{W}_2)\boldsymbol{x} + b_1 - b_2 = 0, \quad \|\boldsymbol{x}\|(\cos(\theta_1) - \cos(\theta_2)) = 0,$$

$$\|\boldsymbol{x}\|(\cos(m\theta_1) - \cos(\theta_2)) = 0 \text{ and } \|\boldsymbol{x}\|(\cos(\theta_1) - \cos(m\theta_2)) = 0,$$

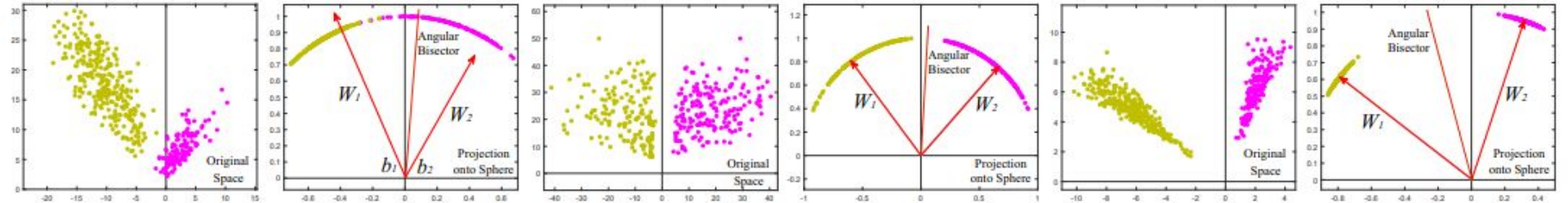
$$p_1 = \frac{\exp(\boldsymbol{W}_1^T\boldsymbol{x} + b_1)}{\exp(\boldsymbol{W}_1^T\boldsymbol{x} + b_1) + \exp(\boldsymbol{W}_2^T\boldsymbol{x} + b_2)} \tag{1}$$

$$p_2 = \frac{\exp(\boldsymbol{W}_2^T\boldsymbol{x} + b_2)}{\exp(\boldsymbol{W}_1^T\boldsymbol{x} + b_1) + \exp(\boldsymbol{W}_2^T\boldsymbol{x} + b_2)} \tag{2}$$

$$L_i = -\log\left(\frac{e^{\boldsymbol{W}_{y_i}^T\boldsymbol{x}_i + b_{y_i}}}{\sum_j e^{\boldsymbol{W}_j^T\boldsymbol{x}_i + b_j}}\right) \qquad \theta_{j,i}(0 \le \theta_{j,i} \le \pi) \tag{4}$$

$$= -\log\left(\frac{e^{\|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\|\cos(\theta_{y_i,i}) + b_{y_i}}}{\sum_j e^{\|\boldsymbol{W}_j\|\|\boldsymbol{x}_i\|\cos(\theta_{j,i}) + b_j}}\right)$$

$$L_{\text{modified}} = \frac{1}{N}\sum_i -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\cos(\theta_{y_i,i})}}{\sum_j e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right) \tag{5}$$



(a) Original Softmax Loss (b) Original Softmax Loss (c) Modified Softmax Loss (d) Modified Softmax Loss (e) A-Softmax Loss (f) A-Softmax Loss

Figure 2: Comparison among softmax loss, modified softmax loss and A-Softmax loss. In this toy experiment, we construct a CNN to learn 2-D features on a subset of the CASIA face dataset. In specific, we set the output dimension of FC1 layer as 2 and visualize the learned features. Yellow dots represent the first class face features, while purple dots represent the second class face features. One can see that features learned by the original softmax loss can not be classified simply via angles, while modified softmax loss can. Our A-Softmax loss can further increase the angular margin of learned features.

A-Softmax Loss

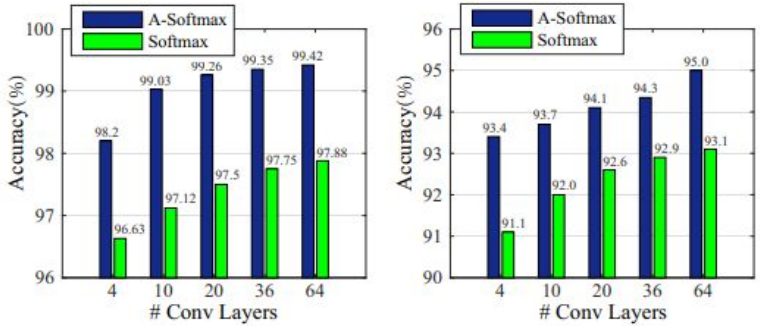


Figure 6: Accuracy (%) on LFW and YTF with different number of convolutional layers. Left side is for LFW, while right side is for YTF.

Loss Function	Decision Boundary
Softmax Loss	$(\boldsymbol{W}_1-\boldsymbol{W}_2)\boldsymbol{x}+b_1-b_2=0$
Modified Softmax Loss	$\|\boldsymbol{x}\|(\cos\theta_1-\cos\theta_2)=0$
A-Softmax Loss	$\|\boldsymbol{x}\|(\cos m\theta_1-\cos\theta_2)=0$ for class 1 $\|\boldsymbol{x}\|(\cos\theta_1-\cos m\theta_2)=0$ for class 2

Table 1: Comparison of decision boundaries in binary case. Note that, θ_i is the angle between $\boldsymbol{W}_i$ and $\boldsymbol{x}$.

$$\cos(\theta_1)>\cos(\theta_2) \quad \cos(m\theta_1)>\cos(\theta_2) \qquad m\geq 2$$

$$L_{\text{ang}}=\frac{1}{N}\sum_i-\log\Big(\frac{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})}}{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})}+\sum_{j\neq y_i}e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\Big) \tag{6}$$

$$L_{\text{ang}}=\frac{1}{N}\sum_i-\log\Big(\frac{e^{\|\boldsymbol{x}_i\|\psi(\theta_{y_i,i})}}{e^{\|\boldsymbol{x}_i\|\psi(\theta_{y_i,i})}+\sum_{j\neq y_i}e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\Big) \tag{7}$$

$$\psi(\theta_{y_i,i})=(-1)^k\cos(m\theta_{y_i,i})-2k,$$

$$\theta_{y_i,i}\in\Big[\frac{k\pi}{m},\frac{(k+1)\pi}{m}\Big] \text{ and } k\in[0,m-1]. \quad m\geq 1$$

A-Softmax Loss

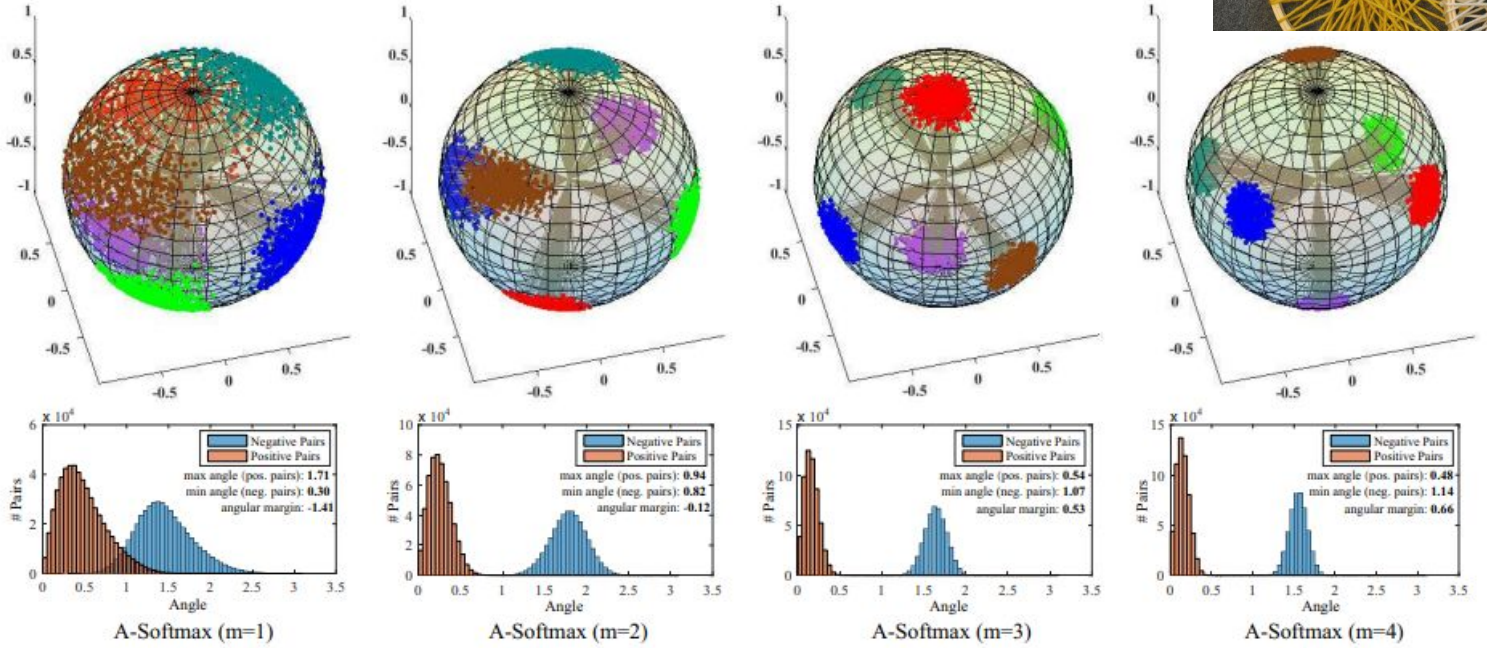
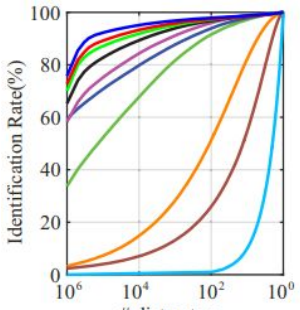
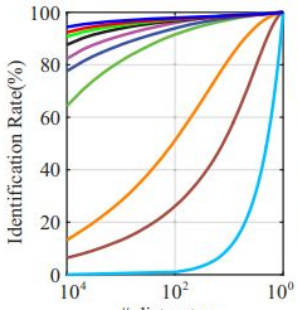


Figure 5: Visualization of features learned with different m. The first row shows the 3D features projected on the unit sphere. The projected points are the intersection points of the feature vectors and the unit sphere. The second row shows the angle distribution of both positive pairs and negative pairs (we choose class 1 and class 2 from the subset to construct positive and negative pairs). Orange area indicates positive pairs while blue indicates negative pairs. All angles are represented in radian. Note that, this visualization experiment uses a 6-class subset of the CASIA-WebFace dataset.

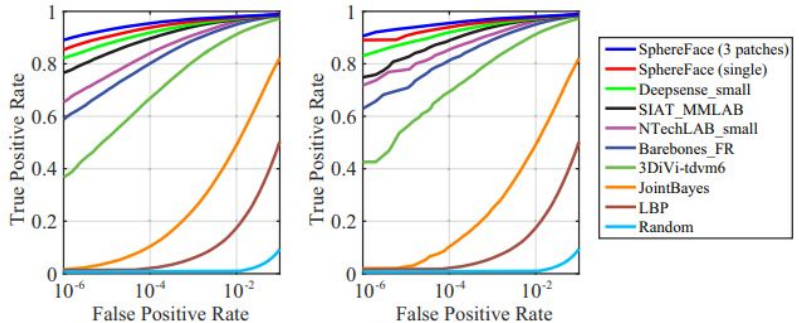
Result



CMC curve on 1M scale



CMC curve on 10K scale



ROC curve on 1M scale

ROC curve on 10K scale

Method	Models	Data	LFW	YTF
DeepFace [30]	3	4M*	97.35	91.4
FaceNet [22]	1	200M*	**99.65**	95.1
Deep FR [20]	1	2.6M	98.95	**97.3**
DeepID2+ [27]	1	300K*	98.70	N/A
DeepID2+ [27]	25	300K*	99.47	93.2
Baidu [15]	1	1.3M*	99.13	N/A
Center Face [34]	1	0.7M*	99.28	94.9
Yi et al. [37]	1	WebFace	97.73	92.2
Ding et al. [2]	1	WebFace	98.43	N/A
Liu et al. [16]	1	WebFace	98.71	N/A
Softmax Loss	1	WebFace	97.88	93.1
Softmax+Contrastive [26]	1	WebFace	98.78	93.5
Triplet Loss [22]	1	WebFace	98.70	93.4
L-Softmax Loss [16]	1	WebFace	99.10	94.0
Softmax+Center Loss [34]	1	WebFace	99.05	94.4
SphereFace	1	WebFace	**99.42**	**95.0**

Table 4: Accuracy (%) on LFW and YTF dataset. * denotes the outside data is private (not publicly available). For fair comparison, all loss functions (including ours) we implemented use 64-layer CNN architecture in Table 2.

ArcFace: Additive Angular Margin Loss for Deep Face Recognition

main challenges in feature learning using Deep Convolutional Neural Networks (DCNNs) for large scale face recognition is the design of appropriate loss functions that can enhance the discriminative power

CosFace

Large Margin Cosine Loss.

$$f_j = W_j^T x = \|W_j\| \|x\| \cos\theta_j, \tag{2}$$

$$L_{lmc} = \frac{1}{N} \sum_i -\log \frac{e^{s(\cos(\theta_{y_i,i}) - m)}}{e^{s(\cos(\theta_{y_i,i}) - m)} + \sum_{j \neq y_i} e^{s\cos(\theta_{j,i})}}, \tag{4}$$

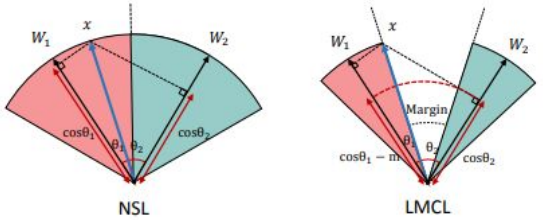


Figure 3. A geometrical interpretation of LMCL from feature perspective. Different color areas represent feature space from distinct classes. LMCL has a relatively compact feature region compared with NSL.

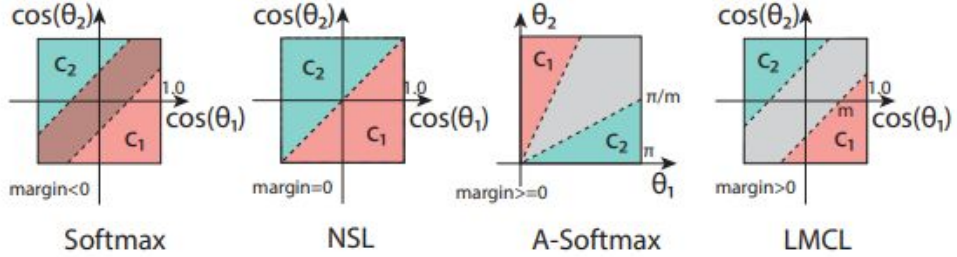


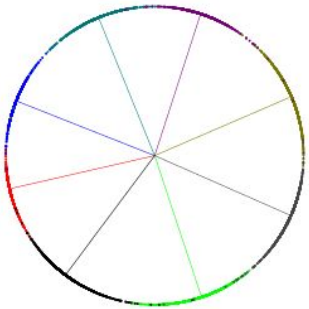
Figure 2. The comparison of decision margins for different loss functions the binary-classes scenarios. Dashed line represents decision boundary, and gray areas are decision margins.

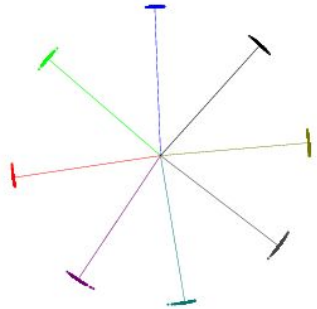
ArcFace Loss

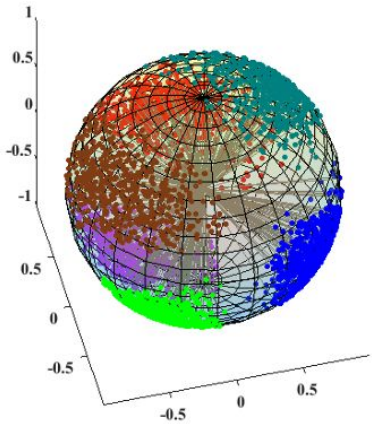
- enhance intra-class compactness and inter-class discrepancy, we consider four kinds of Geodesic Distance (GDis) constraint.
  - Margin-Loss,
  - Intra-Loss,
  - Inter-Loss,
  - Triplet-Loss
- Easy to implement
- Softmax Loss: the size of linear transformation increases linearly.
- Triplet Loss: For large datasets, leads to a significant increase in the number of iteration.

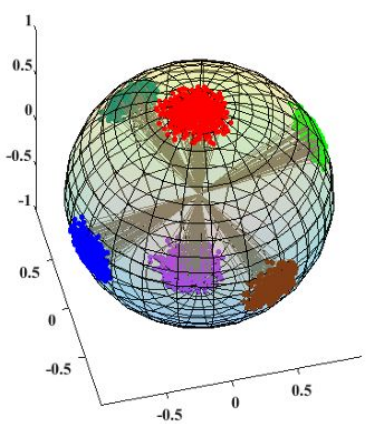
ArcFace Loss

- Toy examples under the softmax and ArcFace loss on
- 8 identities with 2D features. Dots indicate samples and lines refer to the centre direction of each identity. Based on the feature
- normalisation, all face features are pushed to the arc space with
- a fixed radius. The geodesic distance gap between closest classes
- becomes evident as the additive angular margin penalty is incorporated









ArcFace Loss

Intra-Loss is designed to improve the intra-class compactness by decreasing the angle/arc between the sample and the ground truth centre.

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

$$L_2 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\cos\theta_{y_i}}}{e^{s\cos\theta_{y_i}} + \sum_{j=1, j\neq y_i}^{n} e^{s\cos\theta_j}}. \tag{2}$$

$$L_3 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))} + \sum_{j=1, j\neq y_i}^{n} e^{s\cos\theta_j}}. \tag{3}$$

$$L_4 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(m_1\theta_{y_i}+m_2)-m_3)}}{e^{s(\cos(m_1\theta_{y_i}+m_2)-m_3)} + \sum_{j=1, j\neq y_i}^{n} e^{s\cos\theta_j}}. \tag{4}$$

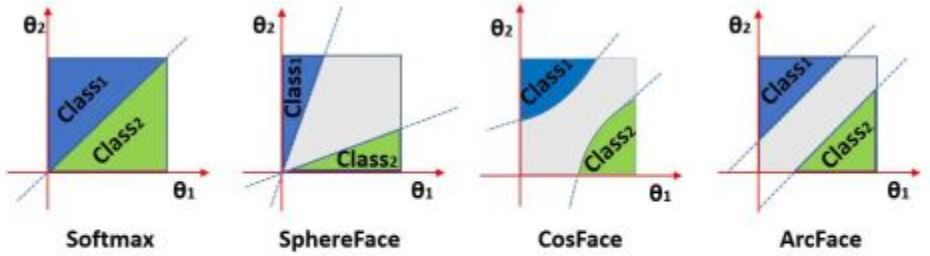


Figure 5. Decision margins of different loss functions under binary classification case. The dashed line represents the decision boundary, and the grey areas are the decision margins.

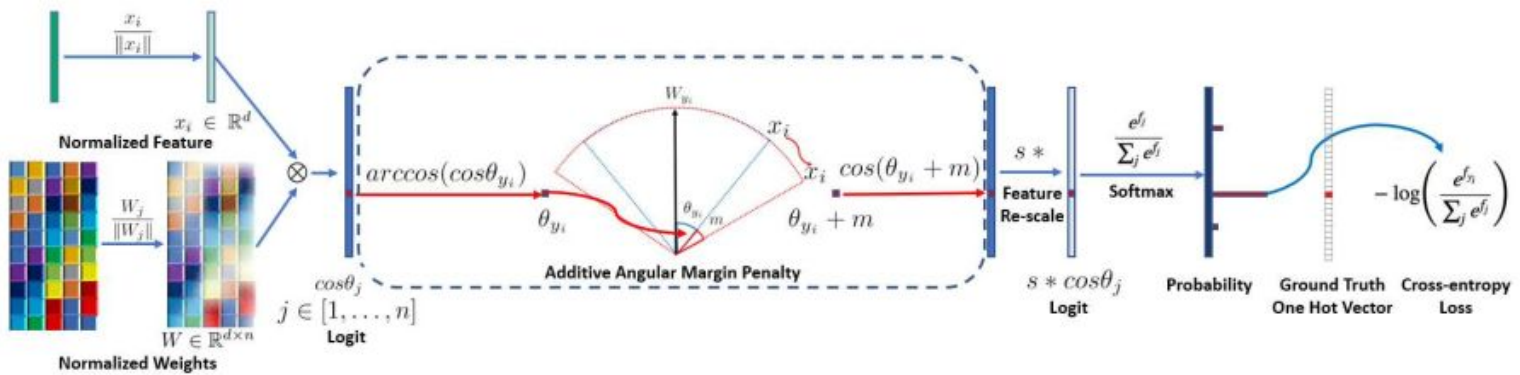


Figure 2. Training a DCNN for face recognition supervised by the ArcFace loss. Based on the feature x_i and weight W normalisation, we get the $\cos\theta_j$ (logit) for each class as $W_j^T x_i$. We calculate the $arccos\theta_{y_i}$ and get the angle between the feature x_i and the ground truth weight W_{y_i}. In fact, W_j provides a kind of centre for each class. Then, we add an angular margin penalty m on the target (ground truth) angle θ_{y_i}. After that, we calculate $\cos(\theta_{y_i} + m)$ and multiply all logits by the feature scale s. The logits then go through the softmax function and contribute to the cross entropy loss.

Algorithm 1 The Pseudo-code of ArcFace on MxNet

Input: Feature Scale s, Margin Parameter m in Eq. 3, Class Number n, Ground-Truth ID gt.

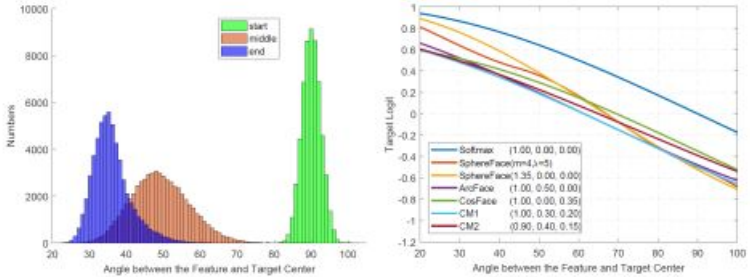
1. x = mx.symbol.L2Normalization (x, mode = 'instance')
2. W = mx.symbol.L2Normalization (W, mode = 'instance')
3. fc7 = mx.sym.FullyConnected (data = x, weight = W, no_bias = True, num_hidden = n)
4. original_target_logit = mx.sym.pick (fc7, gt, axis = 1)
5. theta = mx.sym.arccos (original_target_logit)
6. marginal_target_logit = mx.sym.cos (theta + m)
7. one_hot = mx.sym.one_hot (gt, depth = n, on_value = 1.0, off_value = 0.0)
8. fc7 = fc7 + mx.sym.broadcast_mul (one_hot, mx.sym.expand_dims (marginal_target_logit - original_target_logit, 1))
9. fc7 = fc7 * s

Output: Class-wise affinity score $fc7$.

Numerical Similarity. In Sphere Face Art Face,

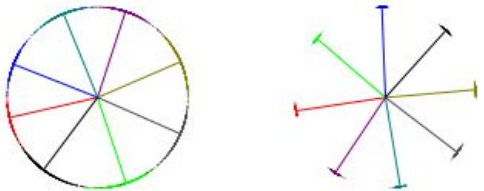
and CosFace , three different kinds of margin

penalty are proposed, e.g. multiplicative angular margin



(a) θ_j Distributions (b) Target Logits Curves

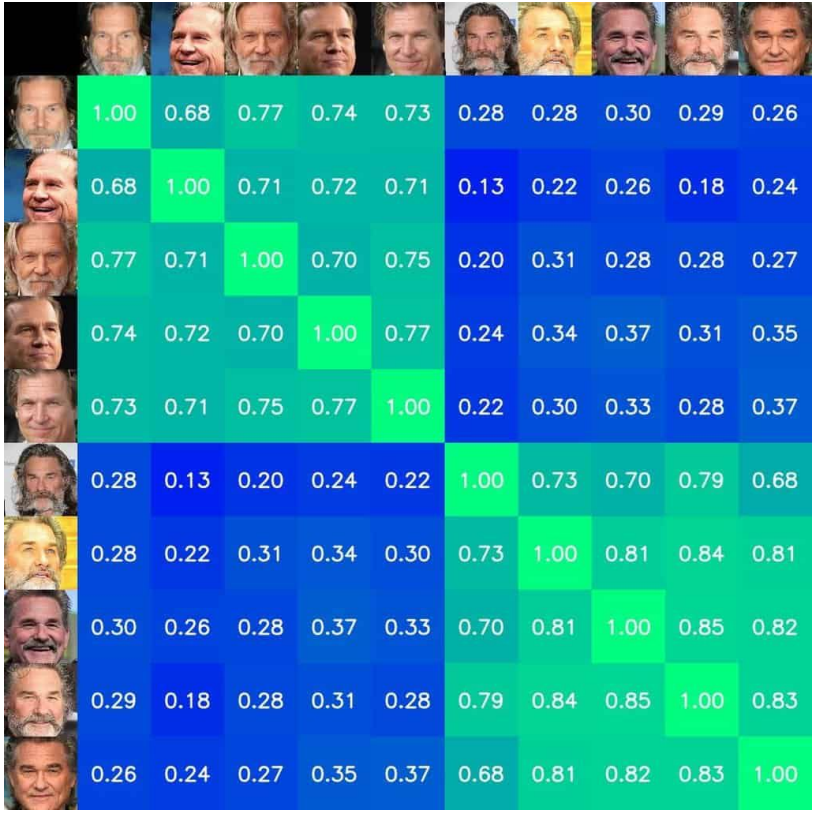
Figure 4. Target logit analysis. (a) θ_j distributions from start to end during ArcFace training. (2) Target logit curves for softmax, SphereFace, ArcFace, CosFace and combined margin penalty $(\cos(m_1\theta + m_2) - m_3)$.



(a) Softmax (b) ArcFace

Figure 3. Toy examples under the softmax and ArcFace loss on 8 identities with 2D features. Dots indicate samples and lines refer to the centre direction of each identity. Based on the feature normalisation, all face features are pushed to the arc space with a fixed radius. The geodesic distance gap between closest classes becomes evident as the additive angular margin penalty is incorporated.

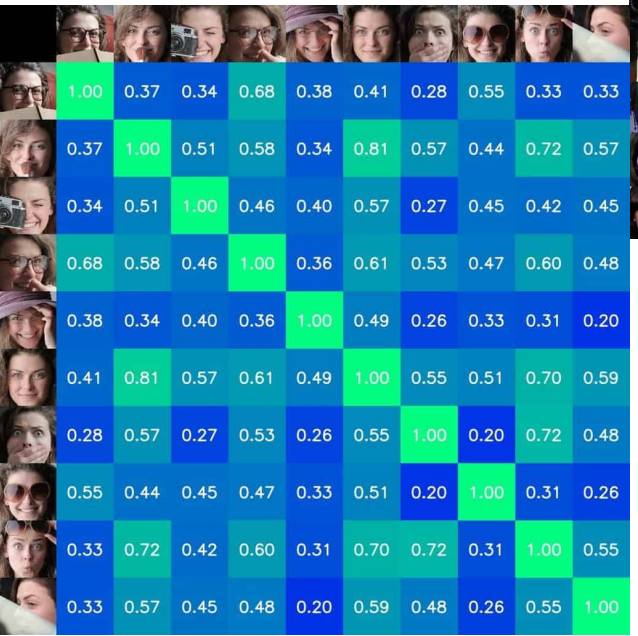
Result

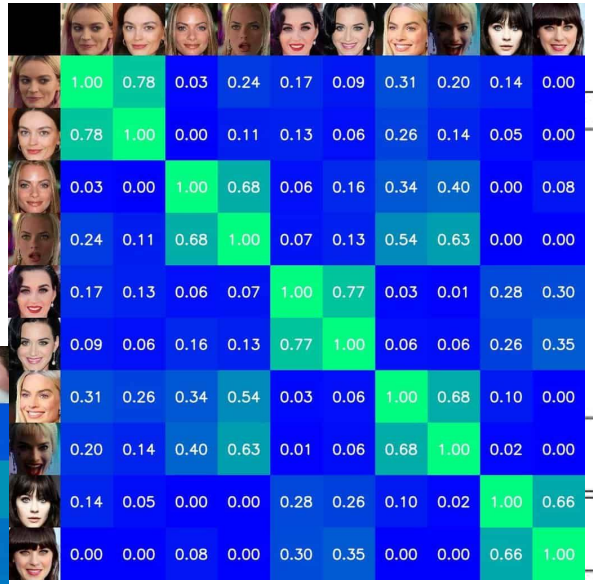


Method	#Image	LFW	YTF
DeepID [30]	0.2M	99.47	93.20
Deep Face [31]	4.4M	97.35	91.4
VGG Face [22]	2.6M	98.95	97.30
FaceNet [27]	200M	99.63	95.10
Baidu [13]	1.3M	99.13	-
Center Loss [36]	0.7M	99.28	94.9
Range Loss [43]	5M	99.52	93.70
Marginal Loss [6]	3.8M	99.48	95.98
SphereFace [15]	0.5M	99.42	95.0
SphereFace+ [14]	0.5M	99.47	-
CosFace [35]	5M	99.73	97.6
MS1MV2, R100, ArcFace	5.8M	**99.83**	**98.02**

Table 4. Verification performance (%) of different methods on LFW and YTF.

Result





Methods	Id (%)	Ver (%)
Softmax [15]	54.85	65.92
Contrastive Loss[15, 30]	65.21	78.86
Triplet [15, 27]	64.79	78.32
Center Loss[36]	65.49	80.14
SphereFace [15]	72.729	85.561
CosFace [35]	77.11	89.88
AM-Softmax [33]	72.47	84.44
SphereFace+ [14]	73.03	-
CASIA, R50, ArcFace	77.50	92.34
CASIA, R50, ArcFace, R	91.75	93.69
FaceNet [27]	70.49	86.47
CosFace [35]	82.72	96.65
MS1MV2, R100, ArcFace	81.03	96.98
MS1MV2, R100, CosFace	80.56	96.56
MS1MV2, R100, ArcFace, R	98.35	98.48
MS1MV2, R100, CosFace, R	97.91	97.91

Table 6. Face identification and verification evaluation of different methods on MegaFace Challenge1 using FaceScrub as the probe set. "Id" refers to the rank-1 face identification accuracy with 1M distractors, and "Ver" refers to the face verification TAR at 10^{-6} FAR. "R" refers to data refinement on both probe set and 1M distractors. ArcFace obtains state-of-the-art performance under both small and large protocols.

Open Questions:

- How often are researchers expanding datasets to enhance training capabilities?
- Are licenses being followed when downloading the images? Do those in the image have the right to remove themselves from the dataset?
- what some other public places are you can get faces. Like would facebook be, ok? They most certainly have a lot of head shots of people.
- When downloading, does it take also the name for future references?
- ArcFace, and SphareFace, as the new images are evaluated without the labels, so, how is that the algorithm calculates the angle to the closest mapped image vector.

Resources:

1. VGGFace2: A dataset for recognising faces across pose and age
2. A Discriminative Feature Learning Approach for Deep Face Recognition
3. FaceNet: A Unified Embedding for Face Recognition and Clustering
4. SphereFace: Deep Hypersphere Embedding for Face Recognition
5. ArcFace: Additive Angular Margin Loss for Deep Face Recognition
6. CosFace: Large Margin Cosine Loss for Deep Face Recognition
7. Squeeze-and-Excitation Networks