Margin Loss: Making Faces More Separable

Riqiang Gao, Fuwei Yang, Wenming Yang, and Qingmin Liao

***Abstract*—The key point of face recognition is creating a discriminative feature representation to ensure intraclass compactness and interclass separability. Softmax loss is widely used in deep learning networks, but it is indirect for face verification. Center loss is effective to improve intraclass compactness, while interclass distances are ignored. In this letter, we propose a novel loss function, termed margin loss, to enlarge distances of interclass and reduce intraclass variations simultaneously. Margin loss aims to focus on samples hard to classify by a distance margin. Different from Softmax loss, margin loss is based on Euclidean distances that can directly measure face similarity. Experiments on different datasets have demonstrated the effectiveness of our method.**

***Index Terms*—Center loss, deep learning, margin loss.**

I. Introduction

FACE recognition (FR) is one of the most popular tasks in computer vision and pattern recognition. The enormous variations in poses, illuminations, occlusions, and expressions make the recognition task challenging. Reducing the intraclass diversifications and enlarging the interclass distances become the critical topics.

Deep learning methods [1]–[6] have attracted wide attention in recent years. Under the large amount of training data and end-to-end framework, discriminative features can be obtained. The DeepID series [2], [7]–[9] demonstrate a set of convolutional neural network architectures that tackle the FR and verification problem simultaneously. The FaceNet [10] presents a unified embedding method for FR and clustering. Wu *et al.* [11] introduce a light CNN framework for face representation on the large-scale data with massive noisy labels. Large-margin Softmax loss [12] is proposed to encourage intraclass compactness and interclass separability between learned features. Wen *et al.* [13] propose center loss that is a discriminative feature learning approach for deep FR. Zhong *et al.* [14] present a method that jointly solve the face alignment and recognition in an end-to-end manner.

Constructing loss function is essential for FR problems. Softmax loss is one of the most common supervision signals in recognition task [2], [5], and there are some improvements of Softmax loss [12], [15]. Range loss [16] is proposed to solve the long-tail training data problem. Contrastive loss [17] is proposed to reduce the intrapersonal variations by pairs-wise training. The features learned from Softmax are generally with large interpersonal distances. However, Softmax layer is indirect and inefficient [10]: The success of network has to depend on the layer representation generalizing well to new images, and the bottleneck layer representation size per image is usually very large. In FaceNet [10], the Softmax loss function is replaced by triplet loss. The triplet loss is more straightforward for verification task since it is based on learning the Euclidean embeddings. Nevertheless, both pairs and triplets selections are hard tasks and the training will get unstable. CenterFace [13] is more stable and easy to convergent since it can be trained in mini-batch. However, center loss cannot be applied independently, and it has to depend on Softmax loss to enlarge interclass distances.

A successful loss function usually focuses on the hard examples. Hinge loss is a loss function that is used for "maximum margin" classification, most notable for support vector machines [18]. Dogan *et al.* [19] propose a unified view on multiclass support vector classification, analyzing various loss functions of multiclass. In [12], Liu *et al.* apply the large-margin mechanism to Softmax loss. Both the Verification signal [17] and Triplet loss [10] define a margin to make training easier.

In this letter, we define a new loss function, named margin loss, to increase interclass differences and reduce the intraclass variations. The class center, with the same dimension of sample feature, is also learned. In the training phase, we enlarge the distances between the sample and relative interclass centers and reduce the intraclass variations simultaneously. Particularly, we set the margin to ensure that our method mainly focuses on the hard samples. That is, we do not optimize the sample that is far away from interclass centers or is close enough to intraclass center in loss function. With the use of margin loss, the Softmax loss can be abandoned in some FR tasks. In summary, our main contributions are described as follows.

First, a new loss function is defined to enlarge interclass differences and reduce intraclass variations. In addition, our algorithm is not only effective because it mainly focuses on hard samples, but also easy to train because no hard simple needs to be selected manually.

Second, since the indirectness and inefficiency of Softmax loss, we can abandon the Softmax loss in the later training phase. We include Softmax loss in the early stage of training to gain a reasonable projection for the training samples.

Third, begin with a toy example of Mnist, extensive experiments on different face databases are performed. In the Mnist

Manuscript received August 21, 2017; revised November 28, 2017; accepted December 23, 2017. Date of publication January 3, 2018; date of current version January 17, 2018. This work was supported in part by the Natural Science Foundation of China under Gant 61471216 and Grant 61771276, in part by the National Key Research and Development Program of China under Grant 2016YFB0101001 and Grant 2017YFC0112500, and in part by the Special Foundation for the Development of Strategic Emerging Industries of Shenzhen under Grant JCYJ20170307153940960 and Grant JCYJ20150831192224146. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Kai Liu. *(Corresponding author: Wenming Yang.)*

The authors are with the Shenzhen Key Laboratory of Information Science and Technology, Shenzhen Engineering Laboratory of IS&DCP and the Department of Electronic Engineering, Graduate School at Shenzhen, Tsinghua University, Beijing 100084, China (e-mail: rqgao15@gmail.com; zjuyfw@163.com; yangelwm@163.com; liaoqm@sz.tsinghua.edu.cn).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2017.2789251

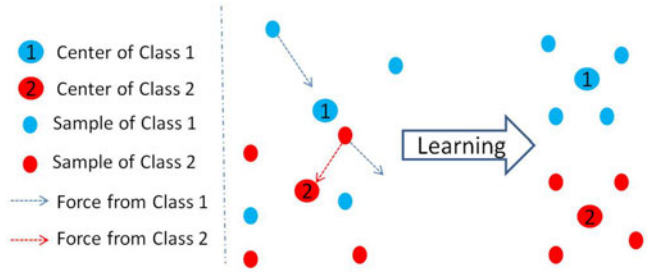


Fig. 1. Motivation of margin loss. We aim to reduce the intraclass variations and enlarge the interclass differences. After learning, the intraclass samples are gathered and the distances of class centers are enlarged.

example, we visualize the results to illustrate the effectiveness of our method.

The rest of the letter is organized as follows. In Section II, we introduce our algorithm in detail. The experiments are presented in Section III, and finally, we conclude our letter in Section IV.

II. Proposed Method

A. Softmax Loss and Center Loss

In the k-class classification, labels can take on k different values. Softmax loss has been widely used in convolutional neural network (CNN). In the training set $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$, $y^{(i)} \in \{1, 2, \ldots, k\}$. The Softmax loss can be described as follows:

$$J_S(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} s\left(y^{(i)} = j\right) \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right] \quad (1)$$

where $\theta_i \in R^{n+1}$ is the parameter of the model. $s(\cdot)$ represents the indicator function. $s(x) = 1$ if x is true, otherwise $s(x) = 0$.

In [13], Wen *et al.* propose a discriminative feature learning method. In order to minimize the intraclass variations, the center loss [13] is defined as follows:

$$J_C(\theta) = \sum_{i=1}^{m} \|x_i - c_{y_i}\|_2^2 \quad (2)$$

where c_{y_i} is the center feature of the y_ith class samples.

B. Margin Loss

We define a loss function, named as margin loss, to enlarge the differences of interclass samples and reduce the intraclass variations. The motivation of margin loss is shown in Fig. 1. Our loss function includes the following considerations.

1) In the FR task, each sample should be kept close to its center (small intraclass variation) and far away from other class centers (large interclass distance) as much as possible.
2) The samples with large enough interclass distance or small enough intraclass distance should be excluded in loss training phase. Otherwise, the training would be unstable and converges slowly. It is crucial to select the hard samples that contribute to effectively.

Thus, margin loss is defined as follows:

$$J_M = \frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[l_{ij}\left(\|x_i - c_j\| - \alpha_{l_{ij}}\right)\right]_+ \quad (3)$$

TABLE I
LeNets++ Structure

Stage 1		Stage 2		Stage 3		Stage 4
Conv	Pool	Conv	Pool	Conv	Pool	FC
(5, 32) × 2	2	(5, 64) × 2	2	(5, 128) × 2	2	2

Notes: (5, 32) × 2 represents that the two cascaded convolution layers with 32 filters of size 5 × 5. All the convolutional strides are 1 and the paddings are 2. The max pooling layers with grid of 2 × 2. All the convolutional strides are 2 and the paddings are 0.

where

$$c_j = \frac{\sum_{i=1}^{m} s(y_i = j) \cdot x_i}{1 + \sum_{i=1}^{m} s(y_i = j)} \quad (4)$$

when the label of x_i equals to j, $l_{ij} = 1$. Otherwise, $l_{ij} = -1$. $\alpha_{l_{ij}}$ is the defined margin. If $l_{ij} = -1$, margin loss only include the samples x_i that satisfies $\|x_i - c_j\| < \alpha_{l_{ij}}$. When $l_{ij} = 1$, x_i is included in margin loss only when $\|x_i - c_j\| \geq \alpha_{l_{ij}}$. Under this condition, margin loss mainly focuses on the hard samples.

In the margin loss, class centers should be updated as the deep features changed. It is impractical and ineffective to update the class centers among the whole training set [13]. Thus, we update the center in mini-batch. In each iteration, the centers would be updated by the samples in the mini-batch. The update of c_j is described as follows:

$$\Delta c_j = \frac{\sum_{i=1}^{m} s(y_i = j) \cdot (c_j - x_i)}{1 + \sum_{i=1}^{m} s(y_i = j)}. \quad (5)$$

We adopt the joint supervision of three loss functions: Softmax loss, center loss, and margin loss. The general formulation of our final loss is given as follows:

$$J = \lambda_0 J_S + \lambda_1 J_C + \lambda_2 J_M \quad (6)$$

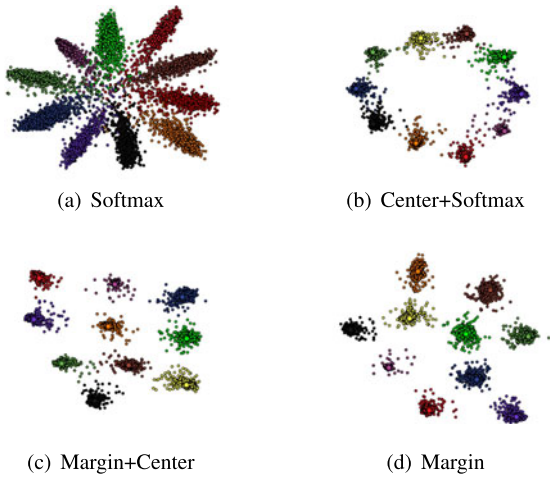
where λ_i is the weight of the relative loss function. To evaluate the effectiveness of our method, six different loss combinations are compared and discussed in Table III.

C. Mnist Example

In this section, we perform our algorithm on the Mnist dataset [20]. For the convenience of comparison, the experimental sets are as same as [13]. The CNN architecture we used in the Mnist example is LeNets++, which is described in Table I. The last hidden layer is restricted to two-dimensional (2-D), which is easy to visualize. The network is consistent with Github of CenterFace [13], and we only change the loss function.

From Fig. 2, we notice that the features from Softmax loss are nearly separable. But they still hold considerable intraclass variations and the interclass distances can be enlarged. The center loss reduces intraclass variations, but it makes no contribution to enlarging interclass distances. On the contrary, margin loss synchronously enlarges the distances of interclass centers and reduces the intraclass variations.

We evaluate the efficiency of our method in Table II. Softmax loss still holds large intraclass variation (reflected on D1), which is adverse to recognition. Center loss reduces the intraclass variation but makes no contribution to enlarging interclass distance (reflected on D2 and D3). When including the margin loss, we achieve small intraclass distances and large interclass distances simultaneously.



(a) Softmax (b) Center+Softmax

(c) Margin+Center (d) Margin

Fig. 2. Comparison of different losses. Compared with (a) and (b) is more discriminative. But the class centers are on a circle. (c) and (d) illustrate that the centers are distributed in the whole 2-D space, indicating margin loss enlarges the interclass distances.

TABLE II
EVALUATION OF THE MNIST EXAMPLE

	D1	D2	D3
S	3.5	5.6	6.4
S+C	1.8	5.9	6.1
C+M	1.7	7.4	7.6
M	1.9	6.9	7.8

Notes: D1 represents the average distance of each sample and its relevant class center. D2 represents the average distance of all the class-center pairs. D3 represents the average distance of each sample between its interclass centers.

D. Discussion of the Algorithm

1) Difference Between Center Loss and Margin Loss: Center loss cannot be applied independently. If we only use center loss, all of the features will converge to the same point. Jointing with the Softmax loss, center loss can learn a discriminative feature. However, center loss is based on the Euclidean distance, which is not in line with Softmax loss. In margin loss, the intraclass variations are reduced and interclass differences are enlarged, and both of them are based on the Euclidean distance.

2) Comparisons With Contrastive Loss and Triplet Loss: Contrastive loss and triplet loss are representative methods for developing discriminative features. In the contrastive loss, the intraclass similarity is enhanced. The triplet loss reduces the intraclass variations and the interclass difference in the same time. However, contrastive loss and triplet loss are easily affected by dramatic data. Similar to Softmax loss and center loss, margin loss can be trained on mini-batch directly.

3) Independence Application: Center loss cannot be used to train model independently. If the CNN network is only supervised by center loss, the deeply learned features and centers would degrade to zeros [13]. Margin loss adds the interclass constraint, which makes the centers distributed separately with large margins. Margin loss can be independently applied in networks since it reduces the intraclass variations and enlarges interclass distance simultaneously.

III. EXPERIMENTS

A. Experiment Settings

In this section, we verify our method on four publicly available databases. CASIA-WebFace [21] and VGGFace [22] are separately used for training. The identification rates on CASIA-WebFace and VGGFace are reported, and face verification tasks are performed on LFW [23], YTF [24], and MegaFace [25].

VGGFace consists of 2.6 million face images of 2622 people. CASIA-webface contains 10 575 subjects and 494 414 images. We separate these two databases for training set and validation set with the proportion of 8:2 (see experiment of Section III-D).

LFW dataset contains 13 233 face images of 5749 individuals. 1680 people have more than one distinct images. YTF dataset includes 3425 videos that come from 1595 different people. These two datasets are widely used to evaluate the performance of face verification algorithm.

MegaFace dataset is very challenging that aims at evaluating face verification and recognition. This challenge contains probe and gallery set. The test set FaceScrub includes 100 000 images of 530 celebrities, and the distractors contains 1 million photos of 690 572 unique users.

The preprocessing works (face and landmark detection) are conducted by multi-task convolutional neural networks [26]. Five landmarks (two eyes, nose, and mouth corners) are used for face alignment.

B. Details of Combinations of Loss Functions

S, C, and M represent the Softmax loss, center loss, and margin loss, respectively, in this letter, and S+C means the model jointly supervised by Softmax loss and center loss. As showed in Table III, we apply six combinations of different losses to verify our method. The Softmax loss (S) is the baseline and $S + C$ is the method presented in [13]. We include the margin loss in this section and change the combination of different losses. The networks of these loss functions are the same and are picked from the related Github of CenterFace for fair comparison.

C. Experiments on LFW, YTF, and MegaFace

In this section, we evaluate our model on three famous face databases: LFW, YTF, and MegaFace. To ensure the reproducibility, our model is trained on CASIA-WebFace. We apply two kinds of distances (Euclidean and Cosine) for verification and Euclidean for identification. The detailed results of LFW and YTF are reported in Table IV and MegaFace's are presented in Table V.

D. Identification on CASIA and VGGFace

Face identification tasks recognize the identity of a test image. In this section, we report the identification rate of CASIA-WebFace and VGGFace. The experimental results are presented in Table VI.

E. Discussion of Experimental Results

1) Comparison of State-of-the-Art Methods: We compare state-of-the-art methods with our algorithm in Tables IV and V. Less training data (0.46 million) and a single network are applied in our algorithm, which does not receive the highest

TABLE III
DESCRIPTION OF LOSSES. $\lambda_0, \lambda_1, \lambda_2$ ARE THE PARAMETERS OF (6)

Loss	S	$S + C$	$S + M$	$S + C + M$	$C + M$(tune)	M(tune)
Description	Softmax loss $\lambda_0 = 1$ $\lambda_1 = 0$ $\lambda_2 = 0$	Softmax + Center loss $\lambda_0 = 1$ $\lambda_1 = 0.008$ $\lambda_2 = 0$	Softmax + Margin loss $\lambda_0 = 1$ $\lambda_1 = 0$ $\lambda_2 = 0.015$	Softmax + Center + Margin loss $\lambda_0 = 1$ $\lambda_1 = 0.008$ $\lambda_2 = 0.01$	Training with Softmax first tuning with Center + Margin loss $\lambda_0 = 0$ $\lambda_1 = 0.008$ $\lambda_2 = 0.01$	Training with Softmax first tuning with Margin loss $\lambda_0 = 0$ $\lambda_1 = 0$ $\lambda_2 = 0.015$

TABLE IV
ACCURACY (%) LFW AND YTF

Method	Trainset	Models	LFW	YTF
DeepFace [5]	4 million	3	97.35	91.4
FaceNet [10]	200 million	1	99.63	95.1
DeepID2 [17]	0.2 million	200	99.15	–
L-Softmax [12]	0.49 million	1	98.71	–
CenterFace [13]	0.7 million	1	99.28	94.9
LightCNN9 [11]	1.5 million	1	98.80	93.4
LightCNN29 [11]	1.5 million	1	99.33	95.5
NormFace [6]	1.5 million	1	99.19	94.72
LN + STN(Pro) [14]	0.46 million	1	99.08	94.7
S (c)	0.46 million	1	96.85	90.32
$S + C$(c)[13]	0.46 million	1	98.37	93.78
$S + M$ (c)	0.46 million	1	98.75	94.12
$S + C + M$ (c)	0.46 million	1	98.93	94.14
$C + M$(tune) (c)	0.46 million	1	98.82	94.27
M(tune) (c)	0.46 million	1	98.58	94.08
S(e)	0.46 million	1	96.62	90.43
$S + C$ (e)[13]	0.46 million	1	98.23	93.52
$S + M$ (e)	0.46 million	1	98.47	93.95
$S + C + M$ (e)	0.46 million	1	98.91	94.04
$C + M$(tune) (e)	0.46 million	1	99.09	94.37
M(tune) (e)	0.46 million	1	99.02	94.22

Note: (c) and (e) represents that the score is computed based on Cosine and Euclidean distance, respectively.

TABLE V
ACCURACY (%) OF MEGAFACE (MF) ON RANK-1 IDENTIFICATION ACCURACY WITH 1 MILLION DISTRACTORS AND VERIFICATION TAR FOR FAR $= 10^{-6}$

Method	Trainset	Rank1 Acc	Ver
Faceall 1600	Large	63.98	63.96
Faceall Norm 1600	Large	64.80	67.12
FaceNet v8	Large	70.50	86.47
NTechLabfacenx-large	Large	73.30	85.08
LightCNN9 [11]	Large	67.11	77.46
LightCNN29 [11]	Large	73.49	84.73
Barebones-FR-cnn	Small	59.36	59.04
Softmax + Contrastive	Small	57.18	69.99
NtechLab-facenx-small	Small	58.22	66.37
L-Softmax Loss [12]	Small	67.13	80.42
S(e)	Small	41.24	41.13
$S + C$ (e)[13]	Small	65.23	76.41
$S + M$ (e)	Small	67.64	76.56
$S + C + M$ (e)	Small	68.35	77.49
$C + M$(tune) (e)	Small	69.68	78.42
M(tune) (e)	Small	68.84	77.37

Note: We apply the Euclidean distance for computation.

TABLE VI
RECOGNITION RATE (%) ON CASIA-WEBFACE AND VGGFACE

Loss Function	Webface	Webface(a)	VGGFace	VGGFace(a)
S(e)	60.4	64.3	54.7	58.3
$S + C$(e)[13]	70.4	74.5	64.7	68.8
$S + M$(e)	71.7	74.9	65.8	69.3
$S + C + M$(e)	73.2	76.1	67.9	72.3
$C + M$(tune)(e)	72.5	74.8	66.4	70.6
M(tune)(e)	71.6	74.5	65.8	69.8

Note: (a) represents that images are aligned.

performance but still competitive. We use the same network and data to compare different losses. Our method beats the softmax loss baseline (e.g., 99.09% beats 96.62% in LFW, 78.42% beats 41.13% in MegaFace) by a large margin and improves the center loss to some degree (e.g., from 98.23% to 99.09% in LFW, from 76.41% to 78.42% in MegaFace). In the identification task (Table VI), $S + C$ performances well and M makes some improvements [e.g., from 65.23% to 69.68%, from 68.8% to 72.3% in VGGFace(a)].

2) Loss Function and Verification Distance: As Fig. 2(a) shows, features from the same identity are inclined to the similar direction with the supervision of Softmax loss, while the center and margin loss are based on the Euclidean distance. And we compare two distances in face verification. When Softmax loss is included in the joint loss function ($\lambda_0 \neq 0$), the verification accuracy based on Cosine distance is higher than that based on the Euclidean distance. On the contrary, the Euclidean distance is more competitive when Softmax loss is abandoned. When the distances of training and verification task are consistent, the performance becomes better.

IV. CONCLUSION

We propose a new loss function termed margin loss for FR network. In this loss function, we enlarge the interclass distances and reduce intraclass variations in the same time. In addition, we define margin in our loss function to focus on the hard samples, which makes our algorithm more effective. The experimental results show our method is competitive in verification and recognition tasks. In this letter, margin loss is based on the Euclidean distance, and verification task-based Euclidean distance reaches the climax when Softmax is abandoned in the later phase. We aim to develop a general version of margin loss to fit different kinds of distances and networks in future work.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.

[2] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 1891–1898.

[3] Y. Wen, Z. Li, and Y. Qiao, "Latent factor guided convolutional neural networks for age-invariant face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 4893–4901.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inform. Process. Syst.*, 2012, pp. 1097–1105.

[5] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 1701–1708.

[6] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: L2 hypersphere embedding for face verification," arXiv:1704.06369, 2017.

[7] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. Adv. Neural Inform. Process. Syst.*, 2014, pp. 1988–1996.

[8] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 2892–2900.

[9] Y. Sun, D. Liang, X. Wang, and X. Tang, "Deepid3: Face recognition with very deep neural networks," arXiv:1502.00873, 2015.

[10] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 815–823.

[11] X. Wu, R. He, Z. Sun, and T. Tan, "A light CNN for deep face representation with noisy labels," arXiv:1511.02683, 2017.

[12] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin Softmax loss for convolutional neural networks," in *Proc. 33rd Int. Conf. Mach. Learning*, 2016, pp. 507–516.

[13] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis*, 2016, pp. 499–515.

[14] Y. Zhong, J. Chen, and B. Huang, "Towards end-to-end face recognition through alignment learning," arXiv:1701.07174, 2017.

[15] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *CVPR*, 2017.

[16] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao, "Range loss for deep face recognition with long-tail," arXiv preprint arXiv: 1611.08976, 2016.

[17] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. Adv. Neural Inform. Process. Syst.*, 2014, pp. 1988–1996.

[18] L. Rosasco, E. De Vito, A. Caponnetto, M. Piana, and A. Verri, "Are loss functions all the same?" *Neural Comput.*, vol. 16, no. 5, pp. 1063–1076, 2004.

[19] U. Dogan, T. Glasmachers, and C. Igel, "A unified view on multi-class support vector classification," *J. Mach. Learning Res.*, vol. 17, no. 45, pp. 1–32, 2016.

[20] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.

[21] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," arXiv:1411.7923, 2014.

[22] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. Brit. Mach. Vis. Conf.*, vol. 1, no. 3, 2015, pp. 6.

[23] G. B. H. E. Learned-Miller, "Labeled faces in the wild: Updates and new reporting procedures," Univ. Massachusetts, Amherst, Tech. Rep. UM-CS-2014-003, May 2014.

[24] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 529–534.

[25] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The megaface benchmark: 1 million faces for recognition at scale," in *Proc. IEEE Conf. Computer Vis. Pattern Recog.*, pp. 4873–4882, 2016.

[26] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.