

Crystal Loss and Quality Pooling for Unconstrained Face Verification and Recognition

Rajeev Ranjan, *Member, IEEE*, Ankan Bansal, Hongyu Xu, *Member, IEEE*,
 Swami Sankaranarayanan, *Member, IEEE*, Jun-Cheng Chen, *Member, IEEE*,
 Carlos D Castillo, *Member, IEEE*, and Rama Chellappa, *Fellow, IEEE*

Abstract—In recent years, the performance of face verification and recognition systems based on deep convolutional neural networks (DCNNs) has significantly improved. A typical pipeline for face verification includes training a deep network for subject classification with softmax loss, using the penultimate layer output as the feature descriptor, and generating a cosine similarity score given a pair of face images or videos. The softmax loss function does not optimize the features to have higher similarity score for positive pairs and lower similarity score for negative pairs, which leads to a performance gap. In this paper, we propose a new loss function, called Crystal Loss, that restricts the features to lie on a hypersphere of a fixed radius. The loss can be easily implemented using existing deep learning frameworks. We show that integrating this simple step in the training pipeline significantly improves the performance of face verification and recognition systems.

Additionally, we focus on the problem of video-based face verification, where the algorithm needs to determine whether a pair of image-sets or videos belong to the same person or not. A compact feature representation is required for every video or image-set, in order to compute the similarity scores. Classical approaches tackle this problem by simply averaging the features extracted from each image/frame of the image-set/video. However, this may lead to sub-optimal feature representations since both good and poor quality faces are weighted equally. To this end, we propose Quality Pooling, which weighs the features based on input face quality. We show that face detection scores can be used as measures of face quality. We also propose Quality Attenuation, which rescales the verification score based on the face quality of a given verification pair. We achieve state-of-the-art performance for face verification and recognition on challenging LFW, IJB-A, IJB-B and IJB-C datasets over a large range of false alarm rates (10^{-1} to 10^{-7}).

Index Terms—Deep Learning, Face Verification, Face Recognition, Loss Functions, Hypersphere Feature Embedding.

arXiv:1804.01159v2 [cs.CV] 4 Feb 2019

1 INTRODUCTION

FACE verification in unconstrained settings is a challenging problem. Despite the excellent performance of recent face verification systems on datasets like Labeled Faces in the Wild (LFW) [20], it is still difficult to achieve similar accuracy on faces with extreme variations in viewpoints, resolution, occlusion and image quality. This is evident from the performance of traditional algorithms on the publicly available IJB-A [23] dataset. Data quality imbalance in the training set is one of the reasons for this performance gap. Existing face recognition training datasets contain large amount of high quality and frontal faces, whereas the unconstrained and difficult faces occur rarely. Most of the DCNN-based methods trained with softmax loss for classification tend to over-fit to the high quality data and fail to correctly classify faces acquired in difficult conditions.

Using the softmax loss function for training face verification system has its own advantages and disadvantages. On the one hand, it can be easily implemented using inbuilt functions from the publicly available deep learning toolboxes such as Caffe [22], Torch [12] and TensorFlow [1]. Unlike triplet loss [43], it does not have any restrictions on the input batch size and converges quickly. The learned features are discriminative enough for efficient face verification without any metric learning. On the other

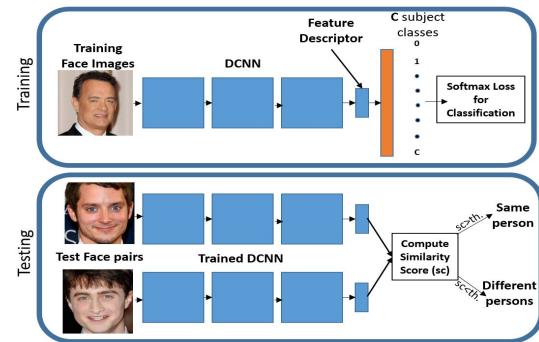


Fig. 1. A general pipeline for training and testing a face verification system using DCNN.

hand, the softmax loss is biased to the sample distribution. Unlike contrastive loss [44] and triplet loss [43] which specifically attend to hard samples, the softmax loss maximizes the conditional probability of all the samples in a given mini-batch. Hence, it is suited to handle high quality faces, ignoring the rare difficult faces in a training mini-batch. We observe that the L_2 -norm of features learned using softmax loss is informative of the quality of the face [35]. Features for good quality frontal faces have a high L_2 -norm while blurry faces with extreme pose have low L_2 -norm (see Figure 2(b)). Moreover, the softmax loss does not optimize the verification requirement of keeping positive pairs closer and negative pairs far from each other. In order to address

R. Ranjan, A. Bansal, H. Xu, S. Sankaranarayanan and R. Chellappa are with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD, 20742 USA e-mail: {rranjan1, ankan, hyxu, swamiv, rama}@umiacs.umd.edu.

J-C Chen and C. D. Castillo are with UMIACS, University of Maryland, College Park, MD, 20742 USA e-mail: {pullpull, carlos}@umiacs.umd.edu.

this limitation, many methods either apply metric learning on top of softmax features [7], [9], [37], [42] or train an auxiliary loss [44], [49], [50] along with the softmax loss to achieve enhanced verification performance.

In this paper, we provide a symptomatic treatment to issues associated with using softmax loss. We propose the Crystal loss function that adds a constraint on the features during training such that their L_2 -norm remains constant. In other words, we restrict the features to lie on a hypersphere of a fixed radius. The proposed Crystal loss has two advantages. Firstly, it provides equal attention to both good and bad quality faces since all the features have the same L_2 -norm now, which is essential for improved performance in unconstrained settings. Secondly, it strengthens the verification features by forcing the same subject features to be closer and features from different subjects to be far from each other in the normalized space. Therefore, it maximizes the margin for the normalized L_2 distance or cosine similarity score between negative and positive pairs. In this way, the proposed Crystal loss overcomes the limitations of the regular softmax loss.

The Crystal loss also retains the advantages of the regular softmax loss. Similar to the softmax loss, it is a one network, one loss system. It doesn't necessarily require any joint supervision as used by many recent methods [37], [44], [49], [50]. It can be easily implemented using inbuilt functions from Caffe [22], Torch [12] and TensorFlow [1], and converges very fast. It introduces just a single scaling parameter to the network. Compared to the regular softmax loss, the Crystal loss gains a significant improvement in the performance. It achieves new state-of-the-art results on IJB-A, IJB-B, IJB-C and LFW datasets, and competitive results on YouTube Face datasets. It surpasses the performance of several state-of-the-art systems, which use multiple networks or multiple loss functions or both. Moreover, the gains from Crystal Loss are complementary to metric learning (eg: TPE [42], joint-Bayes [7]) or auxiliary loss functions (eg: center loss [50], contrastive loss [44]). We show that applying these techniques on top of the Crystal Loss can further improve the verification performance.

We also address the problem of face verification and recognition using videos or image-sets. A video may contain multiple frames with faces of a person of interest. An image-set, sometimes interchangeable with template, may contain multiple images/frames of a person of interest, captured from different sources. In a video-based or template-based face verification problem, we need to determine whether a given pair of videos/templates belong to the same identity. A traditional way to solve this problem is to represent a video or a template using a set of features, each corresponding to its constituent images or frames. This approach is not memory-efficient and does not scale with large number of videos. Additionally, computing similarity scores between two videos for every frame-pair is of a high time complexity. Owing to these limitations, researchers have focused on generating a single feature representation from a given video or a template. A simple approach is to represent the video/template with arithmetic mean of the features of the constituent frames/images. This approach may lead to sub-optimal feature representation since the features for both good as well as bad quality faces get weighted equally. To this end, we propose Quality Pooling, which obtains the weight coefficients using the face detection scores. We show that these probability scores from a face detector could be treated as a measure of the face quality. A good-quality frontal face has a higher detection probability score

compared to a blurry and profile face. Using the precomputed detection score does not require any additional training and improves the performance of video/template-based face verification.

In addition, we focus on improving the face verification performance at low False Accept Rates (FARs). We propose Quality Attenuation, that rescales the similarity score based on maximum of the detection score of the verification pair. It helps in reducing the score for a dissimilar pair if the face quality of both images in the pair is poor, thus increasing the True Accept Rate (TAR) at a given FAR. Experiments on challenging IJB-B and IJB-C datasets show that Quality Attenuation significantly improves the TARs at very low FARs.

In summary, this paper makes the following contributions:

- 1) We propose a simple, novel yet effective Crystal Loss for face verification that restricts the L_2 -norm of the feature descriptor to a constant value α .
- 2) We study the variations in the performance with respect to the scaling parameter α and provide suitable bounds on its value for achieving consistently high performance.
- 3) We propose Quality Pooling, which generates a compact feature representation for a video or template using face detection score.
- 4) We propose Quality Attenuation, which rescales the similarity scores based on the face detection scores of the verification pairs.
- 5) The proposed methods yields consistent and significant improvements on all the challenging face verification datasets namely LFW [20], YouTube Face [28], and IJB-A [23], IJB-B [51] and IJB-C [33]

2 RELATED WORK

In recent years, there has been significant improvements in the accuracy of face verification using deep learning methods [37], [39], [42], [43], [44], [46], [50], [56]. Most of these methods have even surpassed human performance on the LFW [20] dataset. Although these methods use DCNNs, they differ from each other by the type of loss function used for training. For face verification, it is essential for the features of positive subject pair to be closer and features of negative subject pair to be far apart. To solve this problem, researchers have adopted two major approaches.

In the first approach, pairs of face images are input to the training algorithm to learn a feature embedding where positive pairs are closer and negative pairs are far apart. In this direction, Chopra et al. [10] proposed siamese networks with contrastive loss for training. Hu et al. [19] designed a discriminative deep metric with a margin between positive and negative face pairs. FaceNet [43] introduced triplet loss to learn the metric using hard triplet face samples.

In the second approach, the face images along with their subject labels are used to learn discriminative identification features in a classification framework. Most of the recent methods [37], [39], [44], [46], [57] train a DCNN with softmax loss to learn these features which are later used either to directly compute the similarity score for a pair of faces or to train a discriminative metric embedding [7], [42]. Another strategy is to train the network for joint identification-verification task [44], [49], [50]. Xiong et al. [54] proposed transferred deep feature fusion (TDF) which involves two-stage fusion of features trained with different networks and datasets. Template adaptation [13] is applied to further boost the performance.

A recent approach [50] introduced center loss to learn face embeddings which have better discriminative ability. Our proposed method is different from the center loss in the following aspects. First, we use one loss function (i.e., Crystal Loss) whereas [50] uses center loss jointly with the softmax loss during training. Second, center loss introduces $C \times D$ additional parameters during training where C is the number of classes and D is the feature dimension. On the other hand, the Crystal Loss introduces just a single parameter that defines the fixed L_2 -norm of the features. Moreover, the center loss can also be used in conjunction with Crystal Loss, which performs better than center loss trained with regular softmax loss (see Section 6.1.4).

Recently, some algorithms have used feature normalization during training to improve performance. Hasnat et al. [16] uses class-conditional von Mises-Fisher distribution to model the feature representation. SphereFace [30] proposes angular softmax (A-softmax) loss that enables DCNNs to learn angularly discriminative features. Another method called DeepVisage [15] uses a special case of batch normalization [21] technique to normalize the feature descriptor before applying softmax loss. Our proposed method is different as it applies an L_2 -constraint on the feature descriptors enforcing them to lie on a hypersphere of a given radius.

Video-based face recognition has been extensively researched in the past. Some earlier methods [3], [25], [47], [55] represent the video frames or image-sets with appearance subspaces or manifolds. The similarity score for verification is obtained by computing manifold distances. Few other methods represent a video using local features. PEP methods [26] cluster the local features from a part-based representation. VF² [36] aggregates Fisher Vector encodings across different video frames.

Most of the recent deep learning-based methods either use pairwise feature similarity computation for every frame-pair [43], [46] or average frame feature pooling [9], [40], [42] to generate a video representation. The pairwise method is computation and memory expensive, while the average feature pooling treats all the features equally irrespective of the face quality. A recent method performs weighted averaging of the frame-level features, called Neural Aggregation Network (NAN) [57], to predict the averaging coefficients for the set of features. Similar to NAN, our proposed Quality Pooling performs weighted averaging of frame features. But, instead of predicting the coefficients, we generate them using the face detection probability score. We show that the face detection score can be used as a measure of face quality. Hence, features from good quality frames are weighed higher compared to features from poor quality frames. Thus, the proposed method generates a rich feature representation for a video without any additional expense of training a model.

3 MOTIVATION

We first summarize the general pipeline for training a face verification system using DCNN as shown in Figure 1. Given a training dataset with face images and corresponding identity labels, a DCNN is trained as a classification task where the network learns to classify a given face image to its correct identity label. A softmax loss function is used for training the network, given by (1)

$$L_S = -\frac{1}{M} \sum_{i=1}^M \log \frac{e^{W_{y_i}^T f(\mathbf{x}_i) + b_{y_i}}}{\sum_{j=1}^C e^{W_j^T f(\mathbf{x}_i) + b_j}}, \quad (1)$$

where M is the training batch size, \mathbf{x}_i is the i^{th} input face image in the batch, $f(\mathbf{x}_i)$ is the corresponding output of the penultimate layer of the DCNN, y_i is the corresponding class label, and W and b are the weights and bias for the last layer of the network which acts as a classifier.

At test time, feature descriptors $f(\mathbf{x}_g)$ and $f(\mathbf{x}_p)$ are extracted for the pair of test face images \mathbf{x}_g and \mathbf{x}_p respectively using the trained DCNN, and normalized to unit length. Then, a similarity score is computed on the feature vectors which provides a measure of distance or how close the features lie in the embedded space. If the similarity score is greater than a threshold, the face pairs are decided to be of the same person. Usually, the similarity score is computed as the L_2 -distance between the normalized features [37], [43] or by using cosine similarity score s , as given by (2) [7], [40], [42], [50]. Both these similarity measures are equivalent and produce same results.

$$s = \frac{f(\mathbf{x}_g)^T f(\mathbf{x}_p)}{\|f(\mathbf{x}_g)\|_2 \|f(\mathbf{x}_p)\|_2} \quad (2)$$

There are two major issues with this pipeline. First, the training and testing steps for face verification task are decoupled. Training with softmax loss doesn't necessarily ensure the positive pairs to be closer and the negative pairs to be far apart in the normalized or angular space.

Secondly, the softmax classifier is weak in modeling difficult or extreme samples. In a typical training batch with data quality imbalance, the softmax loss gets minimized by increasing the L_2 -norm of the features for easy samples, and ignoring the hard samples. The network thus learns to respond to the quality of the face by the L_2 -norm of its feature descriptor. To validate this claim, we perform a simple experiment on the IJB-A [23] dataset where we divide the templates (groups of images/frames of the same subject) into three different sets based on the L_2 -norm of their feature descriptors. The features were computed using Face-Resnet [50] trained with regular softmax loss. Templates with descriptors' L_2 -norm <90 are assigned to set1. Templates with L_2 -norm >90 but <150 are assigned to set2, while templates with L_2 -norm >150 are assigned to set3. In total, they form six sets of evaluation pairs. Figure 2(a) shows the performance of these six different sets for the IJB-A face verification protocol. It can be clearly seen that pairs having low L_2 -norm for both templates perform very poorly, while pairs with high L_2 -norm perform the best. The difference in performance between each set is quite significant. Figure 2(b) shows some sample templates from set1, set2 and set3 which confirms that the L_2 -norm of the feature descriptor is informative of its quality.

To solve these issues, we enforce the L_2 -norm of the features to be fixed for every face image. Specifically, we add an L_2 -constraint to the feature descriptor such that it lies on a hypersphere of a fixed radius. This approach has two advantages. Firstly, on a hypersphere, minimizing the softmax loss is equivalent to maximizing the cosine similarity for the positive pairs and minimizing it for the negative pairs, which strengthens the verification signal of the features. Secondly, the softmax loss is able to model the extreme and difficult faces better, since all the face features have the same L_2 -norm.

4 PROPOSED METHOD

The proposed Crystal Loss is given by (3)

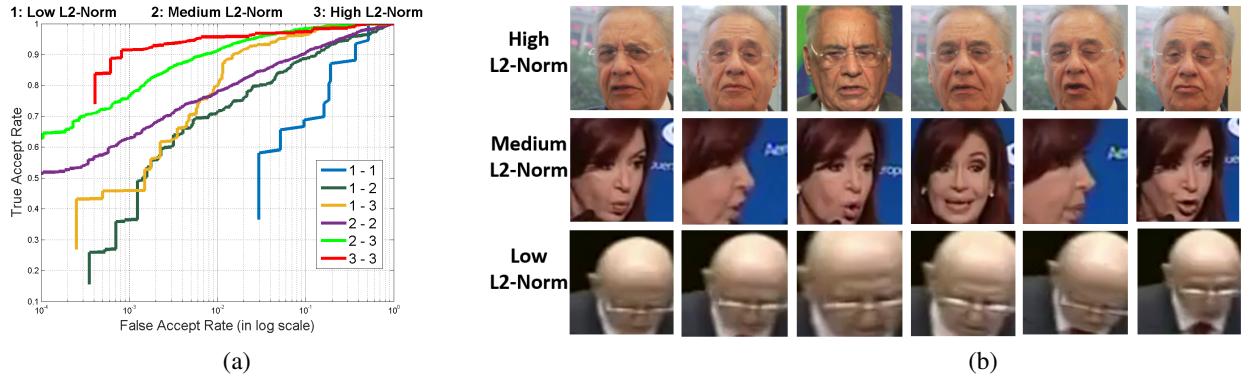


Fig. 2. (a) Face Verification Performance on the IJB-A dataset. The templates are divided into 3 sets based on their L_2 -norm. '1' denotes the set with low L_2 -norm while '3' represents high L_2 -norm. The legend 'x-y' denote the evaluation pairs where one template is from set 'x' and another from set 'y'. (b) Sample template images from IJB-A dataset with high, medium and low L_2 -norm

$$\begin{aligned} \text{minimize} \quad & -\frac{1}{M} \sum_{i=1}^M \log \frac{e^{W_{y_i}^T f(\mathbf{x}_i) + b_{y_i}}}{\sum_{j=1}^C e^{W_j^T f(\mathbf{x}_i) + b_j}} \\ \text{subject to} \quad & \|f(\mathbf{x}_i)\|_2 = \alpha, \quad \forall i = 1, 2, \dots, M, \end{aligned} \quad (3)$$

where \mathbf{x}_i is the input image in a mini-batch of size M , y_i is the corresponding class label, $f(\mathbf{x}_i)$ is the feature descriptor obtained from the penultimate layer of DCNN, C is the number of subject classes, and W and b are the weights and bias for the last layer of the network which acts as a classifier. Equation (3) adds an additional L_2 -constraint to the softmax loss defined in (1). We show the effectiveness of this constraint using MNIST [24] data.

4.1 MNIST Example

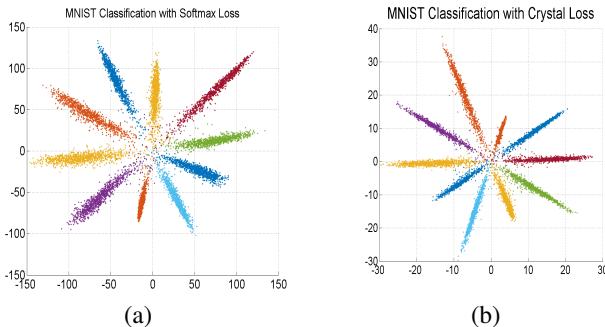


Fig. 3. Vizualization of 2-dimensional features for MNIST digit classification test set using (a) Softmax Loss. (b) Crystal Loss

We study the effect of Crystal Loss on the MNIST dataset [24]. We use a deeper and wider version of LeNet mentioned in [50], where the last hidden layer output is restricted to 2-dimensional space for easy visualization. For the first setup, we train the network end-to-end using the regular softmax loss for digit classification with the number of classes equal to 10. For the second setup, we add an L_2 -normalize layer and a scale layer to the 2-dimensional features which enforces the L_2 -constraint described in (3) (seen Section 4.2 for details). Figure 3 depicts the 2-D features for different classes for the MNIST test set containing 10,000 digit images. Each of the lobes shown in the figure represents 2-D features of unique digits classes. The features for the second setup were obtained before the L_2 -normalization layer.

TABLE 1
Accuracy on MNIST test set in (%)

	Softmax Loss	Crystal Loss
Accuracy	98.88	99.05

We find two clear differences between the features learned using the two setups discussed above. First, the intra-class angular variance is large when using the regular softmax loss, which can be estimated by the average width of the lobes for each class. On the other hand, the features obtained with crystal loss have lower intra-class angular variability, and are represented by thinner lobes. Second, the magnitudes of the features are much higher with the softmax loss (ranging upto 150), since larger feature norms result in a higher probability for a correctly classified class. In contrast, the feature norm has minimal effect on the crystal loss since every feature is normalized to a circle of fixed radius before computing the loss. Hence, the network focuses on bringing the features from the same class closer to each other and separating the features from different classes in the normalized or angular space. Table 1 lists the accuracy obtained with the two setups on MNIST test set. Crystal loss achieves a higher performance, reducing the error by more than 15%. Note that these accuracy numbers are lower compared to a typical DCNN since we are using only 2-dimensional features for classification.

4.2 Implementation Details

Here, we provide the details of implementing the L_2 -constraint described in (3) in the framework of DCNNs. The constraint is enforced by adding an L_2 -normalization layer followed by a scale layer as shown in Figure 5.

This module is added just after the penultimate layer of DCNN which acts as a feature descriptor. The L_2 -normalization layer normalizes the input feature \mathbf{x} to a unit vector given by (4). The scale layer scales the input unit vector to a fixed radius given by the parameter α (5). In total, we just introduce one scalar parameter (α) which can be trained along with the other parameters of the network.

$$\mathbf{y} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2} \quad (4)$$

$$\mathbf{z} = \alpha \cdot \mathbf{y} \quad (5)$$



Fig. 4. Three-dimensional normalized features for three different identities, obtained from (a) network trained with Softmax Loss. (b) network trained with Crystal Loss. The intra-class cosine distance reduces while the inter-class cosine distance increases by using the Crystal Loss.

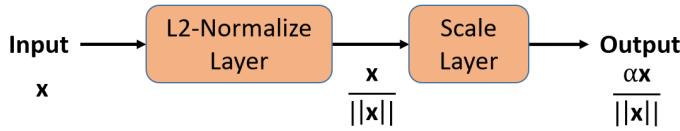


Fig. 5. We add an L_2 -normalization layer and a scale layer to constrain the feature descriptor to lie on a hypersphere of radius α .

The module is fully differentiable and can be used in the end-to-end training of the network. At test time, the proposed module is redundant, since the features are eventually normalized to unit length while computing the cosine similarity. At training time, we backpropagate the gradients through the L2-normalization and the scale layer, as well as compute the gradients with respect to the scaling parameter α using the chain rule as given below.

$$\begin{aligned}\frac{\partial l}{\partial y_i} &= \frac{\partial l}{\partial z_i} \cdot \alpha \\ \frac{\partial l}{\partial \alpha} &= \sum_{j=1}^D \frac{\partial l}{\partial z_j} \cdot y_j \\ \frac{\partial l}{\partial x_i} &= \sum_{j=1}^D \frac{\partial l}{\partial y_j} \cdot \frac{\partial y_j}{\partial x_i} \\ \frac{\partial y_i}{\partial x_i} &= \frac{\|\mathbf{x}\|_2^2 - x_i^2}{\|\mathbf{x}\|_2^3} \\ \frac{\partial y_j}{\partial x_i} &= \frac{-x_i \cdot x_j}{\|\mathbf{x}\|_2^3}\end{aligned}\tag{6}$$

The features learned using Softmax Loss and Crystal Loss are shown in Figure 4. We train two networks, one with Softmax Loss and another with Crystal Loss, using 100 training identities. We restrict the feature dimension to three for better visualization on a sphere. The blue, green and red points depict the L_2 -normalized features for three different identities. It is clear from the figure that Crystal Loss forces the features to have a low intra-class angular variability and higher inter-class angular variability, which improves the face verification accuracy.

4.3 Bounds on Parameter α

The scaling parameter α plays a crucial role in deciding the performance of L_2 -softmax loss. There are two ways to enforce the L_2 -constraint: 1) by keeping α fixed throughout the training, and 2) by letting the network to learn the parameter α . The second way is elegant and always improves over the regular softmax loss. But, the α parameter learned by the network is high which results in a relaxed L_2 -constraint. The softmax classifier aimed at increasing the feature norm for minimizing the overall loss, increases the α parameter instead, allowing it more freedom to fit to the easy samples. Hence, the α learned by the network forms an upper bound for the parameter. Improved performance is obtained by fixing α to a lower constant value.

On the other hand, with a very low value of α , the training algorithm does not converge. For instance, $\alpha = 1$ performs poorly on the LFW [20] dataset, achieving an accuracy of 86.37% (see Figure 11). The reason being that a hypersphere with small radius (α) has limited surface area for embedding features from the same class together and those from different classes far from each other.

Here, we formulate a theoretical lower bound on α . Assuming the number of classes C to be lower than twice the feature dimension D , we can distribute the classes on a hypersphere of dimension D such that the centers of any two classes are at least 90° apart. Figure 6(a) represents this case for $C = 4$ class centers distributed on a circle of radius α . We assume the classifier weights (W_i) to be a unit vector pointing in the direction of their respective class centers. We ignore the bias term. The average softmax probability p for correctly classifying a feature is given by (7)

$$p = \frac{e^{W_i^T X_i}}{\sum_{j=1}^4 e^{W_j^T X_i}} \quad (7)$$

$$= \frac{e^\alpha}{e^\alpha + 2 + e^{-\alpha}}$$

Ignoring the term $e^{-\alpha}$ and generalizing it for C classes, the average probability becomes:

$$p = \frac{e^\alpha}{e^\alpha + C - 2} \quad (8)$$

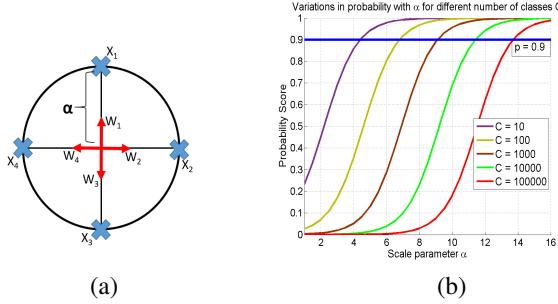


Fig. 6. (a) 2-D visualization of the assumed distribution of features (b) Variation in Softmax probability with respect to α for different number of classes C

Figure 6(b) plots the probability score as a function of the parameter α for various number of classes C . We can infer that to achieve a given classification probability (say $p = 0.9$), we need to have a higher α for larger C . Given the number of classes C for a dataset, we can obtain the lower bound on α to achieve a probability score of p by using (9).

$$\alpha_{low} = \log \frac{p(C - 2)}{1 - p} \quad (9)$$

4.4 Relation to von Mises-Fisher Distribution

The distribution of features learned using Crystal Loss can be characterized as a special case of von Mises-Fisher distribution [16]. In directional statistics, von Mises-Fisher distribution is a probability distribution on a hypersphere, whose probability density function is represented using (10)

$$f_p(\mathbf{x}, \mu, \kappa) = C_p \exp(\kappa \mu^T \mathbf{x}), \quad (10)$$

where $\kappa \geq 0$ is the concentration parameter, $\|\mu\|_2 = 1$, $\|\mathbf{x}\|_2 = 1$, and C_p is the normalization constant dependent on κ and the feature dimension p . Keeping the concentration parameter κ same for all the C classes, the log maximum a posteriori estimate for the parameters of von Mises-Fisher distribution results in the formulation of Crystal Loss (\mathbf{L}) as shown in (11)

$$\begin{aligned} \mathbf{L} &= \text{maximize } \log \frac{f_p(\mathbf{x}_i, \mu_i, \kappa)}{\sum_{j=1}^C f_p(\mathbf{x}_j, \mu_j, \kappa)} \\ &= \text{minimize } -\log \frac{\exp(\kappa \mu_i^T \mathbf{x}_i)}{\sum_{j=1}^C \exp(\kappa \mu_j^T \mathbf{x}_j)} \end{aligned} \quad (11)$$

The concentration parameter κ corresponds to the scale factor in the Crystal Loss. The κ value decides the spread of the features on the hypersphere, as shown in Figure 7¹. A low value of κ results in high intra-class angular variability, while a high value of κ decreases the inter-class angular distance. Hence, an optimal value of κ or the scale factor for Crystal Loss is required (see Section 4.3) so that features from same class are close together and features from different classes are far from each other in angular space. We do not normalize the classifier weight vectors since it significantly slows down the training process for large number of classes.

1. https://en.wikipedia.org/wiki/Von_MisesFisher_distribution

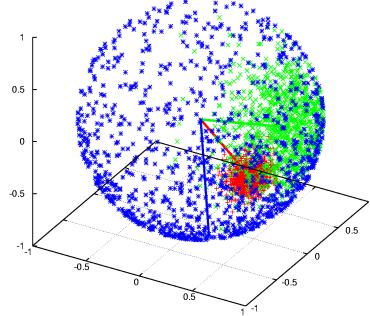


Fig. 7. Visualization of features on a sphere sampled from von Mises-Fisher distribution. The blue, green and red color represents features for different concentration parameters $\kappa = 1$, $\kappa = 10$ and $\kappa = 100$ respectively.

5 QUALITY POOLING AND ATTENUATION

In this section, we propose Quality Pooling to flatten the images/frames in a template/video, and Quality Attenuation to rescale the similarity score for a verification pair. Both these methods use the precomputed face detection score obtained from a face detector. We observe that the face detection score is an indicator of the quality of a face image. A high resolution and frontal face has a higher detection probability compared to a blurry face or faces in extreme pose (see Figure 8). We use the Single Shot Detector [29] trained for face detection task [38] to generate the detection probabilities.

5.1 Quality Pooling

Given a video/template T containing set of frames/images $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_k\}$, let the corresponding feature vectors be denoted by $\{\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \dots, \mathbf{f}_k\}$. The feature descriptor \mathbf{r} for the video/template T is given by (12)

$$\mathbf{r} = \sum_{i=1}^k c_i \mathbf{f}_i, \quad (12)$$

where c_i is the coefficient for the weighted sum corresponding to the feature of i^{th} frame/template. We can compute the coefficients as shown in (13)

$$c_i = \frac{e^{\lambda l_i}}{\sum_{j=1}^k e^{\lambda l_j}}, \quad (13)$$

where λ is a hyperparameter, and l_i is the logit corresponding to the face detection probability p_i , and is given by (14)

$$l_i = \min\left(\frac{1}{2} \log \frac{p_i}{1 - p_i}, 7\right). \quad (14)$$

The logits are upper bounded by 7 to avoid exponentially large values when the detection probability score is close to 1.0. The variation in Quality Pooling performance with the hyperparameter λ is discussed in Section 6.4. We use the value of $\lambda = 0.3$ in our experiments. Algorithm 1 provides the pseudo-code of the Quality Pooling method for generating a compact feature representation.



Fig. 8. The face detection probability score for images of three different templates (one template each row) from IJB-A dataset [23]. The scores reflect the face quality of the images. Higher scores correspond to good quality images, while lower scores are predicted for blurry and extreme pose faces.

Algorithm 1 Quality Pooling

```

1:  $k \leftarrow$  number of frames in a video
2: for  $i = 1$  to  $k$  do
3:    $p_i \leftarrow$  get_detection_score(frame)
4:    $f_i \leftarrow$  get_identity_descriptor(frame)
5: end
6: feature_descriptor  $\leftarrow 0$ 
7:  $q \leftarrow 0.3$ 
8: for  $i = 1$  to  $k$  do
9:    $l_i \leftarrow \min\left(\frac{1}{2} \log \frac{p_i}{1-p_i}, 7\right)$ 
10:   $c_i \leftarrow \frac{e^{\lambda l_i}}{\sum_j^k e^{\lambda l_j}}$ 
11:  feature_descriptor  $\leftarrow$  feature_descriptor +  $c_i f_i$ 
12: end
  
```

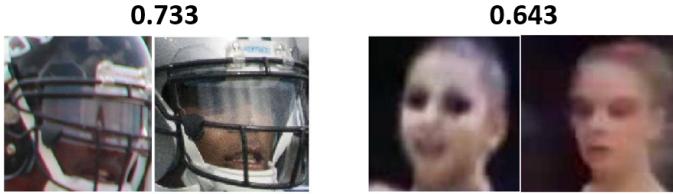


Fig. 9. The non-match verification pairs from IJB-B dataset. Although the pairs have poor face quality, their cosine similarity scores (shown above the images) are high.

5.2 Quality Attenuation

We observe that feature descriptors for high quality faces are more discriminative, compared to those for low quality faces. Hence, the similarity scores generated for verification pairs containing both low quality faces are unreliable. This causes a non-match pair to be assigned with a high similarity score, which can significantly reduce the TAR at very low FARs of 10^{-6} , 10^{-7} , etc. Figure 9 shows a couple of non-match verification pairs with poor face quality, for which the network generates high similarity scores. In order to reduce the similarity scores for low quality face pairs, we scale down the similarity score by a factor of γ if maximum face detection probability of one of the template/video in the verification pair is less than a given threshold (set to 0.75 in our experiments). Algorithm 2 provides the pseudo-code of the proposed Quality Attenuation method.

Algorithm 2 Quality Attenuation

```

1:  $k_1 \leftarrow$  number of images in template  $T_1$ 
2:  $k_2 \leftarrow$  number of images in template  $T_2$ 
3: for  $i = 1$  to  $k_1$  do
4:    $p_i \leftarrow$  get_detection_score( $T_{1i}$ )
5: end
6:  $\text{lomax}_1 \leftarrow \max\{p_1, p_2, p_3, \dots p_{k_1}\}$ 
7: for  $i = 1$  to  $k_2$  do
8:    $p_i \leftarrow$  get_detection_score( $T_{2i}$ )
9: end
10:  $\text{lomax}_2 \leftarrow \max\{p_1, p_2, p_3, \dots p_{k_2}\}$ 
11:  $\text{score} \leftarrow$  get_similarity_score( $T_1, T_2$ )
12: if ( $\text{lomax}_1$  or  $\text{lomax}_2$ )  $\leq$  det_threshold then
13:    $\text{score} \leftarrow \frac{\text{score}}{\gamma}$ 
14: end
  
```

6 RESULTS

We use the publicly available Face-Resnet [50] DCNN for our experiments. Figure 10 shows the architecture of the network. It contains 27 convolutional layers and 2 fully-connected layers. The dimension of the feature descriptor is 512. It utilizes the widely used residual skip-connections [18]. We add an L_2 -normalization layer and a scale layer after the fully-connected layer to enforce the L_2 -constraint on the descriptor. All our experiments are carried out in Caffe [22].

6.1 Baseline experiments

In this subsection, we experimentally validate the usefulness of the L_2 -softmax loss for face verification. We form two subsets of training dataset from the MS-Celeb-1M [14] dataset: 1) MS-small containing 0.5 million face images with the number of subjects being 13403, and 2) MS-large containing 3.7 million images of 58207 subjects. The dataset was cleaned using the clustering algorithm presented in [27]. We train the Face-Resnet network with softmax loss as well as Crystal loss for various α . While training with MS-small, we start with a base learning rate of 0.1 and decrease it by $1/10^{th}$ after 16K and 24K iterations, upto a maximum of 28K iterations. For training on MS-large, we use the same learning rate but decrease it after 50K and 80K iterations upto a maximum of 100K iterations. A training batch size of 256 was used. Both softmax and Crystal loss functions consume

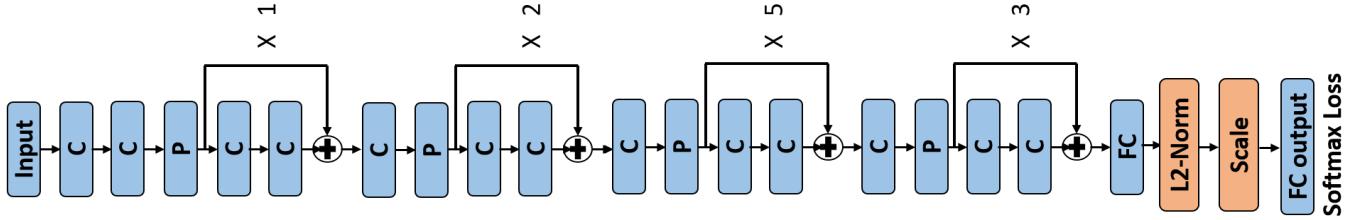


Fig. 10. The Face-Resnet architecture [50] used for the experiments. **C** denotes Convolution Layer followed by PReLU [17] while **P** denotes Max Pooling Layer. Each pooling layer is followed by a set of residual connections, the count for which is denoted alongside. After the fully-connected layer (**FC**), we add an L_2 -Normalize layer and Scale Layer which is followed by the softmax loss.

the same amount of training time which is around 9 hours for MS-small and 32 hours for MS-large training set respectively, on two TITAN X GPUs. We set the learning multiplier and decay multiplier for the scale layer to 1 for trainable α , and 0 for fixed α during the network training. We evaluate our baselines on the widely used LFW dataset [20] for the unrestricted setting, and the challenging IJB-A dataset [23] for the 1:1 face verification protocol. The faces were cropped and aligned to the size of 128×128 in both training and testing phases by implementing the face detection and alignment algorithm presented in [40].

6.1.1 Experiment with small training set

Here, we compare the network trained on MS-small dataset using the proposed Crystal loss, against the one trained with regular softmax loss. Figure 11 shows that the softmax loss attains an accuracy of 98.1% whereas the proposed Crystal loss achieves the best accuracy of 99.28%, thereby reducing the error by more than 62%. It also shows the variations in performance with the scale parameter α . The performance is poor when α is below a certain threshold and stable with α higher than the threshold. This behavior is consistent with the theoretical analysis presented in Section 4.3. From the figure, the performance of Crystal loss is better for $\alpha > 12$ which is close to its lower bound computed using equation 9 for $C = 13403$ with a probability score of 0.9.

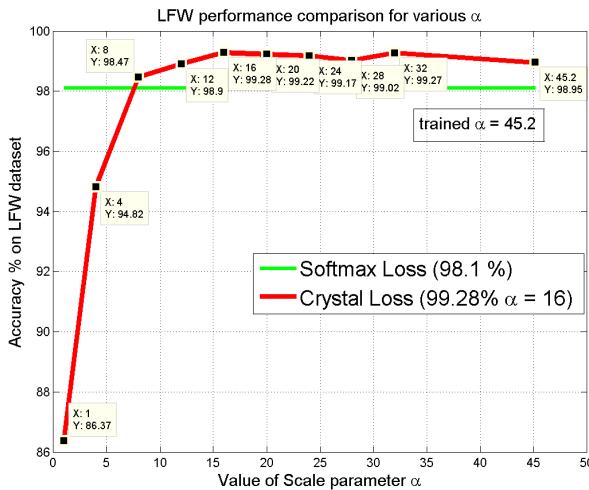


Fig. 11. The red curve shows the variations in LFW accuracy with the parameter α for Crystal loss. The green line is the accuracy using softmax loss.

A similar trend is observed for 1:1 verification protocol on IJB-A [23] as shown in Table 2, where the numbers denote True Accept Rate (TAR) at False Accept Rates (FAR) of 0.0001, 0.001, 0.01 and 0.1. Our proposed approach improves the TAR@FAR=0.0001 by 19% compared to the baseline softmax loss. The performance is consistent with α ranging between 16 to 32. Another point to note is that by allowing the network to learn the scale parameter α by itself results in a slight decrease in performance, which shows that having a tighter constraint is a better choice.

TABLE 2
TAR on IJB-A 1:1 Verification Protocol @FAR

	0.0001	0.001	0.01	0.1
Softmax Loss	0.553	0.730	0.881	0.957
Crystal Loss ($\alpha=8$)	0.257	0.433	0.746	0.953
Crystal Loss ($\alpha=12$)	0.620	0.721	0.875	0.970
Crystal Loss ($\alpha=16$)	0.734	0.834	0.924	0.974
Crystal Loss ($\alpha=20$)	0.740	0.820	0.922	0.973
Crystal Loss ($\alpha=24$)	0.744	0.831	0.912	0.974
Crystal Loss ($\alpha=28$)	0.740	0.834	0.922	0.975
Crystal Loss ($\alpha=32$)	0.727	0.831	0.921	0.972
Crystal Loss (α trained)	0.698	0.817	0.914	0.971

6.1.2 Experiment with large training set

We train the network on the MS-large dataset for this experiment. Figure 12 shows the performance on the LFW dataset. Similar to the small training set, the Crystal loss significantly improves over the baseline, reducing the error by 60% and achieving an accuracy of 99.6%. Similarly, it improves the TAR@FAR=0.0001 on IJB-A by more than 10% (Table 3). The performance of Crystal loss is consistent with α in the range 40 and beyond. Unlike, the small set training, the self-trained α performs equally good compared to fixed α of 40 and 50. The theoretical lower bound on α is not of much use in this case since improved performance is achieved for $\alpha > 30$. We can deduce that as the number of subjects increases, the lower bound on α is less reliable, and the self-trained α is more reliable with performance. This experiment clearly suggests that the proposed Crystal loss is consistent across the training and testing datasets.

TABLE 3
TAR on IJB-A 1:1 Verification Protocol @FAR

	0.0001	0.001	0.01	0.1
Softmax Loss	0.730	0.851	0.926	0.972
Crystal Loss ($\alpha=30$)	0.775	0.871	0.938	0.978
Crystal Loss ($\alpha=40$)	0.827	0.900	0.951	0.982
Crystal Loss ($\alpha=50$)	0.832	0.906	0.952	0.981
Crystal Loss (α trained)	0.832	0.903	0.950	0.980

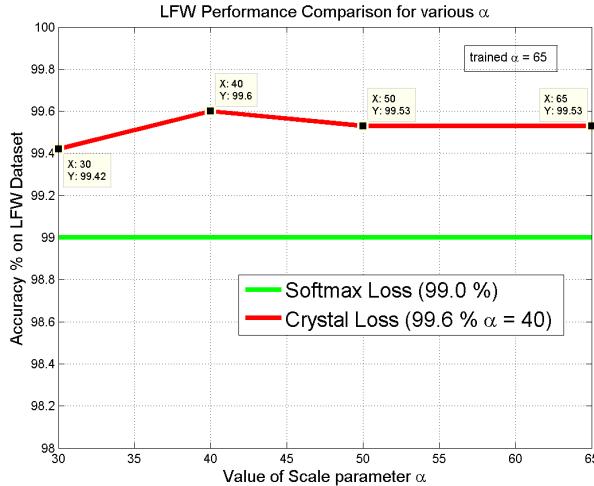


Fig. 12. The red curve shows the variations in LFW accuracy with the parameter α for Crystal loss. The green line is the accuracy using the Softmax loss.

6.1.3 Experiment with a different DCNN

To check the consistency of our proposed Crystal loss, we apply it on the All-In-One Face [40] instead of the Face-Resnet. We use the recognition branch of the All-In-One Face to fine-tune on the MS-small training set. The recognition branch of All-In-One Face consists of 7 convolution layers followed by 3 fully-connected layers and a softmax loss. We add an L_2 -normalize and a scale layer after the 512 dimension feature descriptor. Figure 13 shows the comparison of Crystal loss and the Softmax loss on LFW dataset. Similar to the Face-Resnet, All-In-One Face with Crystal loss improves over the Softmax performance, reducing the error by 40%, and achieving an accuracy of 98.82%. The improvement obtained by using All-In-One Face is smaller compared to the Face-Resnet. This shows that residual connections and depth of the network generate better feature embedding on a hypersphere. The performance variation with scaling parameter α is similar to that of Face-Resnet, indicating that the optimal scale parameter does not depend on the choice of the network.

TABLE 4
Accuracy on LFW (%)

Softmax loss	98.10
Center loss [50] + Softmax loss	99.23
Crystal loss	99.28
Center loss [50] + Crystal loss	99.33

6.1.4 Experiment with auxiliary loss

Similar to softmax loss, the Crystal loss can be coupled with auxiliary losses such as center loss, contrastive loss, triplet loss, etc. to further improve the performance. Here we study the performance variation of Crystal loss when coupled with the center loss. We use the MS-small dataset for training the networks. Table 4 lists the accuracy obtained on the LFW dataset by different loss functions. The softmax loss performs the worst. The center loss improves the performance significantly when trained in conjunction with the softmax loss, and is comparable to the Crystal loss. Training center loss with Crystal loss gives the best performance of 99.33%

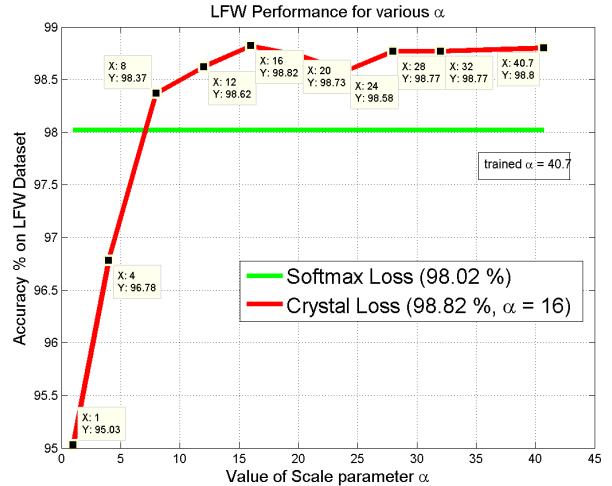


Fig. 13. The red curve shows the variations in LFW accuracy with the parameter α for Crystal loss. The green line is the accuracy using the Softmax loss.

accuracy. This shows that Crystal loss is as versatile as the softmax loss and can be used efficiently with other auxiliary loss functions.

6.2 Experiments on LFW and YTF Datasets

TABLE 5
Verification accuracy (in %) of different methods on LFW and YTF datasets.

Method	Images	#nets	One loss	LFW	YTF
Deep Face [46]	4M	3	No	97.35	91.4
DeepID-2+ [44]	-	25	No	99.47	93.2
FaceNet [43]	200M	1	Yes	99.63	95.12
VGG Face [37]	2.6M	1	No	98.95	97.3
Baidu [28]	1.3M	1	No	99.13	-
Wen et al. [50]	0.7M	1	No	99.28	94.9
NAN [57]	3M	1	No	-	95.72
DeepVisage [15]	4.48M	1	Yes	99.62	96.24
SphereFace [30]	0.5M	1	Yes	99.42	95.0
Softmax(FR)	3.7M	1	Yes	99.0	93.82
CrL (FR)	3.7M	1	Yes	99.60	95.54
CrL (R101)	3.7M	1	Yes	99.67	96.02
CrL (RX101)	3.7M	1	Yes	99.78	96.08

We compare our algorithm with recently reported face verification methods on LFW [20], YouTube Face [52] and IJB-A [23] datasets. We crop and align the images for all these datasets by implementing the algorithm mentioned in [40]. We train the Face-Resnet (FR) with Crystal loss (CrL) as well as Softmax loss using the MS-large training set. Additionally, we train ResNet-101(R101) [18] and ResNeXt-101(RX101) [53] deep networks for face recognition using MS-large training set with Crystal loss. Both R101 and RX101 models were initialized with parameters pre-trained on ImageNet [41] dataset. A fully-connected layer of dimension 512 was added before the Crystal loss classifier. The scaling parameter was kept fixed with a value of $\alpha = 50$. Experimental results on different datasets show that Crystal loss works efficiently with deeper models.

The LFW dataset [20] contains 13,233 web-collected images from 5749 different identities. We evaluate our model following

TABLE 6
Face Identification and Verification Evaluation on IJB-A dataset

Method	IJB-A Verification (TAR@FAR)				IJB-A Identification			
	0.0001	0.001	0.01	0.1	FPIR=0.01	FPIR=0.1	Rank=1	Rank=10
GOTS [23]	-	0.2(0.008)	0.41(0.014)	0.63(0.023)	0.047(0.02)	0.235(0.03)	0.443(0.02)	-
B-CNN [11]	-	-	-	-	0.143(0.027)	0.341(0.032)	0.588(0.02)	-
LFSFS [48]	-	0.514(0.06)	0.733(0.034)	0.895(0.013)	0.383(0.063)	0.613(0.032)	0.820(0.024)	-
VGG-Face [37]	-	0.604(0.06)	0.805(0.03)	0.937(0.01)	0.46(0.07)	0.67(0.03)	0.913(0.01)	0.981(0.005)
DCNN _{manual} +metric [8]	-	-	0.787(0.043)	0.947(0.011)	-	-	0.852(0.018)	0.954(0.007)
Pose-Aware Models [31]	-	0.652(0.037)	0.826(0.018)	-	-	-	0.840(0.012)	0.946(0.007)
Chen et al. [7]	-	-	0.838(0.042)	0.967(0.009)	0.577(0.094)	0.790(0.033)	0.903(0.012)	0.977(0.007)
Deep Multi-Pose [2]	-	-	0.876	0.954	0.52	0.75	0.846	0.947
Masi et al. [32]	-	0.725	0.886	-	-	-	0.906	0.977
Triplet Embedding [42]	-	0.813(0.02)	0.90(0.01)	0.964(0.005)	0.753(0.03)	0.863(0.014)	0.932(0.01)	0.977(0.005)
Template Adaptation [13]	-	0.836(0.027)	0.939(0.013)	0.979(0.004)	0.774(0.049)	0.882(0.016)	0.928(0.01)	0.986(0.003)
All-In-One Face [40]	-	0.823(0.02)	0.922(0.01)	0.976(0.004)	0.792(0.02)	0.887(0.014)	0.947(0.008)	0.988(0.003)
NAN [57]	-	0.881(0.011)	0.941(0.008)	0.979(0.004)	0.817(0.041)	0.917(0.009)	0.958(0.005)	0.986(0.003)
PPN [6]	0.775	0.852	0.901	-	-	-	0.914	0.938
TDFP [54]	0.875(0.013)	0.919(0.006)	0.961(0.007)	0.988(0.003)	0.878(0.035)	0.941(0.010)	0.964(0.006)	0.992(0.002)
TDFP [54]+TPE [42]	0.877(0.018)	0.921(0.005)	0.961(0.007)	0.989(0.003)	0.881(0.039)	0.940(0.009)	0.964(0.007)	0.992(0.003)
model-A	0.914(0.016)	0.948(0.006)	0.971(0.004)	0.985(0.002)	0.917(0.048)	0.960(0.005)	0.974(0.004)	0.989(0.002)
model-B	0.914(0.018)	0.949(0.005)	0.969(0.003)	0.984(0.002)	0.918(0.043)	0.959(0.005)	0.972(0.004)	0.988(0.003)
model-C	0.907(0.018)	0.947(0.004)	0.968(0.003)	0.983(0.002)	0.917(0.043)	0.958(0.005)	0.972(0.004)	0.988(0.003)

the standard protocol of unrestricted with labeled outside data. We test on 6,000 face pairs and report the experiment results in Table 5. Along with the accuracy values, we also compare with the number of images, networks and loss functions used by the methods for their overall training. The proposed method attains state-of-the-art performance with the RX101 model, achieving an accuracy of 99.78%. Unlike other methods which use auxiliary loss functions such as center loss and contrastive loss along with the primary softmax loss, our method uses a single loss training paradigm which makes it easier and faster to train.

YouTube Face (YTF) [52] dataset contains 3425 videos of 1595 different people, with an average length of 181.3 frames per video. It contains 10 folds of 500 video pairs. We follow the standard verification protocol and report the average accuracy on splits with cross-validation in Table 5. We achieve the accuracy of 96.08% using Crystal loss with RX101 network. Our method outperforms many recent algorithms and is only behind DeepVisage [15] which uses larger number of training samples, and VGG Face [37] which further uses a discriminative metric learning on YTF.

6.3 Experiments on IJB Datasets

We evaluate the proposed models on three challenging IARPA Janus Benchmark datasets, namely IJB-A [23], IJB-B [51] and IJB-C [33]. We use Universe face dataset, a combination of curated MS-Celeb-1M [14], UMDFaces [5] and UMDFaces-Videos [4] datasets, for training the network. We remove the subject overlaps from all the three IJB-A, IJB-B and IJB-C datasets. In total, the training data contains 58,020 subjects and 5,714,444 images. We use ResNet-101 [18] architecture with Crystal Loss for training. The scale factor α was set to 50. Since the Crystal loss can be coupled with any other auxiliary loss, we use the Triplet Probabilistic Embedding (TPE) [42] to learn a 128-dimensional embedding using the images from UMDFaces [5] dataset. In order to showcase the effect of Quality Pooling and Quality Attenuation, we evaluate the following three models on IJB-A, IJB-B and IJB-C datasets for the tasks of 1:1 Verification and 1:N Identification:

- **model-A** - ResNet-101 trained with Crystal Loss ($\alpha=50$), TPE [42], media average pooling.

- **model-B** - ResNet-101 trained with Crystal Loss ($\alpha=50$), TPE [42], Quality Pooling ($\lambda=0.3$).

- **model-C** - ResNet-101 trained with Crystal Loss ($\alpha=50$), TPE [42], Quality Pooling ($\lambda=0.3$), Quality Attenuation ($\gamma=1.1$)

6.3.1 IJB-A Dataset

The IJB-A dataset [23] contains 500 subjects with a total of 25,813 images including 5,399 still images and 20,414 video frames. It contains faces with extreme viewpoints, resolution and illumination which makes it more challenging than the commonly used LFW dataset. The dataset is divided into 10 splits, each containing 333 randomly sampled subjects for training and remaining 167 subjects for testing. Given a template containing multiple faces of the same individual, we generate a common vector representation. Table 6 lists the performance of recent DCNN-based methods on the IJB-A dataset. We achieve state-of-the-art result for both verification and the identification protocols. The **model-A** attains a record TAR of 0.914 @ FAR = 0.0001. Our method performs significantly better than existing methods in most of the other metrics as well. The Quality Pooling and Quality Attenuation do not show much improvement for in performance, since they are most effective at very low FARs.

6.3.2 IJB-B Dataset

The IJB-B dataset [51] is an extension to the publicly available IJB-A [23] dataset. It contains 1,845 unique subjects with a total of 21,798 still images and 55,026 video frames collected in unconstrained settings. The dataset is more challenging and diverse than IJB-A, with protocols designed to test detection, identification, verification and clustering of faces. Unlike the IJB-A dataset, it does not contain any training splits. The verification protocol contains of 8,010,270 between S1 and S2 gallery templates and 1:N Mixed Media probe templates. In total, they result in 10,270 genuine comparisons and 8,000,000 impostor comparisons. This allows us to evaluate the performance at very low FARs of 10^{-5} and 10^{-6} . We compare our proposed methods

TABLE 7
1:1 Face Verification Evaluation on IJB-B dataset

Method	IJB-B 1:1 Verification (True Accept rate (in %) @ False Accept Rate)					
	0.000001	0.00001	0.0001	0.001	0.01	0.1
GOTS [51]	-	-	16.0	33.0	60.0	-
VGGFace [50]	-	-	55.0	72.0	86.0	-
FPN [6]	-	-	83.2	91.6	96.5	-
model-A	48.39	80.43	89.84	94.44	97.23	98.92
model-B	50.79	82.76	90.44	94.61	97.12	98.82
model-C	50.04	82.54	90.29	94.49	97.09	98.77

TABLE 8
1:N Face Identification Evaluation on IJB-B dataset

Method	IJB-B 1:N Identification				
	TPIR (%) @ FPIR		Retrieval Rate (%)		
Method	0.001	0.01	0.1	Rank=1	Rank=10
GOTS [51]	-	-	-	42.0	62.0
VGGFace [50]	-	-	-	78.0	89.0
FPN [6]	-	-	-	91.1	96.5
model-A	43.44	82.75	91.55	93.56	97.28
model-B	39.55	83.59	91.91	93.69	97.19
model-C	38.92	82.96	91.67	93.59	97.18

with Government-off-the-shelf (GOTS [51]), VGGFace [50] and FacePoseNet (FPN [6]). Table 7 lists the performance of various methods on 1:1 Verification protocol of IJB-B dataset. We achieve significant improvement over the previous state-of-the-art methods, with TAR of 50.79% at FAR = 10^{-6} . Table 8 provides the results for 1:N identification protocol on IJB-B dataset. We evaluate both open-set and closed-set protocol. Since, the dataset contains two set of galleries S1 and S2, we report the average performance of both the gallery sets. We achieve a True Positive Identification Rate (TPIR) of 43.44% at False Positive Identification Rate (FPIR) of 0.1% in the open-set protocol, and a Rank-1 accuracy of 93.69% in the closed-set protocol. The proposed methods perform significantly better than previous state-of-the-art algorithms. The results show that Quality Pooling performs better than the naive media averaging of templates. We do not see much improvement with Quality Attenuation, since its more applicable to very low FARs and FPIRs.

6.3.3 IJB-C Dataset

The IJB-C dataset [33] is an extension to the publicly available IJB-B [51] dataset. It contains 3,531 unique subjects with a total of 31,334 still images and 117,542 video frames collected in unconstrained settings. Similar to IJB-B dataset, the protocols are designed to test detection, identification, verification and clustering of faces. The dataset also contains end-to-end protocols to evaluate the algorithm's ability to perform end-to-end face recognition. The verification protocol contains 19,557 genuine comparisons and 15,638,932 impostor comparisons. This allows us to evaluate the performance at very low FARs of 10^{-6} and 10^{-7} .

We compare our proposed methods with Government-off-the-shelf (GOTS [33]), VGGFace [37] and FaceNet [43]. We also report the performance of three Janus systems, namely Janus1, Janus2 and UMD, on IJB-C [33] dataset. The results of Janus performers were obtained in a private communication as of Jan-

uary 16, 2018. The UMD system performs score-level fusion of **model-C** with three other networks: 1) ResNet-101 [18] trained on MSCeleb-1M dataset with Crystal Loss, 2) Inception-ResNet-V2 [45] trained on Universe face dataset with Softmax Loss, and 3) All-In-One Face [40] trained on MSCeleb-1M dataset using Softmax Loss. Quality Pooling and Quality Attenuation is applied to all the networks involved in fusion. The score-level fusion from different networks improves the performance in most of the metrics.

Table 9 lists the performance of the proposed methods on 1:1 Verification protocol of IJB-C dataset. We achieve state-of-the-art results with TARs of 71.37% and 81.15% at FARs = 10^{-7} and 10^{-6} respectively. Table 10 provides the results for 1:N identification protocol on IJB-C dataset, where the average performance on gallery sets G1 and G2 are reported. We achieve a TPIR of 78.54% at FPIR of 0.1% in the open-set protocol, and a Rank-1 accuracy of 94.73% in the closed-set protocol. The proposed methods outperform previously state-of-the-art algorithms by a large margin. Quality Pooling shows improvement for most of the metrics, whereas Quality Attenuation improves low FARs of 10^{-6} and 10^{-7} for the verification protocol and low FPIR of 0.1% for the identification protocol.

6.4 Effect of Quality Pooling and Quality Attenuation

Here, we provide the ablation study of using the Quality Pooling and Quality Attenuation modules for IJB-C [33] 1:1 verification protocol. Table 11 shows the variations in performance for different λ parameter setting of Quality Pooling. $\lambda = 0$ corresponds to the simple media averaging technique used in [42]. We observe that the performance improves by increasing λ upto a value of 0.2, after which it decreases consistently for all FARs. This proves that the DCNN feature descriptors are more reliable for good quality faces, which makes Quality Pooling perform better than naive media averaging.

Table 12 shows the variations in performance for different Quality Attenuation parameter γ . A γ value of 1.0 corresponds to no attenuation in similarity score. From the table, we see that Quality Attenuation with $\gamma = 1.1$ significantly improves the performance at very low FARs of 10^{-7} and 10^{-6} with negligible decrease in performance at high FARs. Thus, Quality Attenuation is quite helpful in the face recognition system where false positives are highly undesirable.

7 DISCUSSION

We present some observations about Crystal Loss, Quality Pooling and Quality Attenuation based on our experiments. Crystal Loss improves the performance over Softmax Loss by a large margin,

TABLE 9
Face Verification Evaluation on IJB-C dataset

Method	IJB-C 1:1 Verification (True Accept rate (in %) @ False Accept Rate)						
	0.0000001	0.000001	0.00001	0.0001	0.001	0.01	0.1
GOTS [33]	-	3.00	6.61	14.67	33.04	61.99	80.93
FaceNet [43]	-	20.95	33.30	48.69	66.45	81.76	92.45
VGGFace [37]	-	32.20	43.69	59.75	74.79	87.13	95.64
model-A	65.96	76.46	86.25	91.91	95.72	97.83	99.14
model-B	66.55	77.91	87.75	92.50	95.78	97.75	99.09
model-C	71.37	81.15	87.35	92.29	95.63	97.66	99.06
Janus1	-	62.93	76.37	87.13	93.90	97.43	99.01
Janus2	-	49.95	76.21	85.99	92.31	95.73	97.15
UMD	-	80.82	87.62	92.52	95.75	97.63	99.03

TABLE 10
1:N Face Identification Evaluation on IJB-C dataset

Method	IJB-C 1:N Identification				
	TPIR (%) @ FPIR		Retrieval Rate (%)		
0.001	0.01	0.1	Rank=1	Rank=10	
GOTS [33]	2.66	5.78	15.60	37.85	60.24
FaceNet [43]	20.58	32.40	50.98	69.22	81.36
VGGFace [37]	26.18	45.06	62.75	78.60	89.20
model-A	78.42	86.09	91.91	94.56	97.53
model-B	78.16	87.38	92.27	94.64	97.51
model-C	78.54	87.01	92.10	94.57	97.48
Janus1	69.54	81.67	89.83	94.03	97.75
Janus2	36.03	75.74	85.70	89.77	94.95
UMD	78.28	87.58	92.38	94.73	97.44

TABLE 11
Effect of Quality Pooling on IJB-C 1:1 Verification

TAR@FAR	Quality Pooling parameter λ					
	0	0.1	0.2	0.3	0.4	0.5
10^{-7}	65.96	67.54	67.98	66.55	64.12	61.73
10^{-6}	76.46	78.68	78.95	77.91	77.28	76.26
10^{-5}	86.25	87.61	88.00	87.75	87.36	86.95
10^{-4}	91.91	92.53	92.54	92.50	92.16	91.82
10^{-3}	95.72	95.83	95.85	95.78	95.55	95.36
10^{-2}	97.83	97.90	97.83	97.75	97.64	97.54
10^{-1}	99.14	99.15	99.13	99.09	99.04	99.00

TABLE 12
Effect of Quality Attenuation on IJB-C 1:1 Verification

TAR@FAR	Quality Attenuation parameter γ				
	1.0	1.1	1.2	1.3	1.4
10^{-7}	66.55	71.37	70.68	70.47	70.46
10^{-6}	77.91	81.15	81.03	80.39	80.07
10^{-5}	87.75	87.35	86.70	86.17	85.59
10^{-4}	92.50	92.29	91.99	91.48	90.84
10^{-3}	95.78	95.63	95.45	95.21	94.99
10^{-2}	97.75	97.66	97.57	97.48	97.37
10^{-1}	99.09	99.06	99.02	98.99	98.95

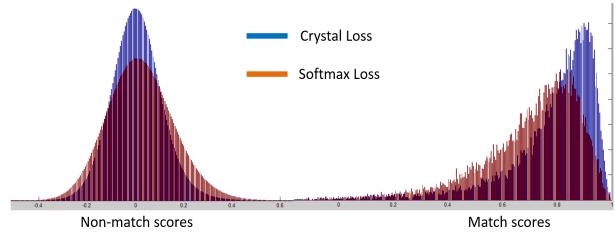


Fig. 14. Similarity Score distribution with Crystal Loss and Softmax Loss for face verification pairs of IJB-C [33] dataset.

since it specifically constrains the intra-class features to be close to each other in angular space. This is evident from Fig. 14 which shows the distribution of similarity scores for verification pairs of IJB-C [33]. We observe that the match scores and non-match scores are more separated with Crystal Loss compared to Softmax Loss, which makes it valuable for face verification and identification.

Quality Pooling and Quality Attenuation use face detection score to determine the authenticity of a given feature representation. The results show that the DCNN features are not invariant to the face size, pose or resolution. Similarity score from a high quality face pair is more reliable.

We also visualize the top 15 false positive verification pairs generated by **model-C** for IJB-C [33] dataset, that contributes to FAR of 10^{-6} , in Fig. 15. We observe that most of the templates contain faces with ethnicity other than Caucasians. This shows that just like humans, the DCNN models suffer from the “other-race effect” [34], as most of the training data are biased towards Caucasian faces. One way to solve this problem is to incorporate sufficient amount of images from all the ethnicities in the training dataset.

8 CONCLUSION

In this paper, we proposed Crystal Loss that adds a simple, yet effective, L_2 -constraint to the regular softmax loss for training a face verification system. The constraint enforces the features to lie on a hypersphere of a fixed radius characterized by parameter α . We also provided bounds on the value of α for achieving a consistent performance. Additionally, we proposed Quality Pooling for generating better feature representation of a face video or template. Quality Attenuation also helps in improving the performance at very low FARs. Experiments on LFW, YTF and IJB datasets show that the proposed methods provide significant and consistent



Fig. 15. Top 15 False Positive verification pairs generated using **model-C** on IJB-C [33] dataset at false accept rate of 10^{-6}

improvements and achieve state-of-the-art results on IJB-A [23], IJB-B [51] and IJB-C [33] datasets for both face verification and face identification tasks. In conclusion, Crystal loss is a valuable replacement for the existing softmax loss, for the task of face recognition. In the future, we would further explore the possibility of exploiting the geometric structure of the feature encoding using manifold-based metric learning.

ACKNOWLEDGMENTS

This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced

Research Projects Activity (IARPA), via IARPA R&D Contract No. 2014-14071600012. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

REFERENCES

- [1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. [1](#), [2](#)
- [2] W. AbdAlmageed, Y. Wu, S. Rawls, S. Harel, T. Hassner, I. Masi, J. Choi, J. Lekust, J. Kim, P. Natarajan, et al. Face recognition using deep multi-pose representations. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016. [10](#)
- [3] O. Arandjelovic, G. Shakhnarovich, J. Fisher, R. Cipolla, and T. Darrell. Face recognition with image sets using manifold density divergence. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 581–588. IEEE, 2005. [3](#)
- [4] A. Bansal, C. Castillo, R. Ranjan, and R. Chellappa. The do’s and don’ts for cnn-based face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2545–2554, 2017. [10](#)
- [5] A. Bansal, A. Nanduri, C. Castillo, R. Ranjan, and R. Chellappa. Umdfaces: An annotated face dataset for training deep networks. *arXiv preprint arXiv:1611.01484*, 2016. [10](#)
- [6] F.-J. Chang, A. T. Tran, T. Hassner, I. Masi, R. Nevatia, and G. Medioni. Faceposenet: Making a case for landmark-free face alignment. In *2017 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 1599–1608. IEEE, 2017. [10](#), [11](#)
- [7] J.-C. Chen, V. M. Patel, and R. Chellappa. Unconstrained face verification using deep cnn features. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016. [2](#), [3](#), [10](#)
- [8] J.-C. Chen, R. Ranjan, A. Kumar, C.-H. Chen, V. M. Patel, and R. Chellappa. An end-to-end system for unconstrained face verification with deep convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 118–126, 2015. [10](#)
- [9] J.-C. Chen, R. Ranjan, S. Sankaranarayanan, A. Kumar, C.-H. Chen, V. M. Patel, C. D. Castillo, and R. Chellappa. Unconstrained still/video-based face verification with deep convolutional neural networks. *International Journal of Computer Vision*, pages 1–20, 2017. [2](#), [3](#)
- [10] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005. [2](#)
- [11] A. R. Chowdhury, T.-Y. Lin, S. Maji, and E. Learned-Miller. One-to-many face recognition with bilinear cnns. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016. [10](#)
- [12] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011. [1](#), [2](#)
- [13] N. Crosswhite, J. Byrne, O. M. Parkhi, C. Stauffer, Q. Cao, and A. Zisserman. Template adaptation for face verification and identification. *arXiv preprint arXiv:1603.03958*, 2016. [2](#), [10](#)
- [14] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*, pages 87–102. Springer, 2016. [7](#), [10](#)
- [15] A. Hasnat, J. Bohné, S. Gentric, and L. Chen. Deepvisage: Making face recognition simple yet with powerful generalization skills. *arXiv preprint arXiv:1703.08388*, 2017. [3](#), [9](#), [10](#)
- [16] M. A. Hasnat, J. Bohné, J. Milgram, S. Gentric, and L. Chen. von mises-fisher mixture model-based deep learning: Application to face verification. [3](#), [6](#)
- [17] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. [8](#)
- [18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. [7](#), [9](#), [10](#), [11](#)
- [19] J. Hu, J. Lu, and Y.-P. Tan. Discriminative deep metric learning for face verification in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1875–1882, 2014. [2](#)
- [20] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007. [1](#), [2](#), [5](#), [8](#), [9](#)
- [21] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. [3](#)
- [22] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. [1](#), [2](#), [7](#)
- [23] B. F. Klare, B. Klein, E. Tabor斯基, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1931–1939, 2015. [1](#), [2](#), [3](#), [7](#), [8](#), [9](#), [10](#), [12](#)
- [24] Y. LeCun, C. Cortes, and C. J. Burges. The mnist database of handwritten digits, 1998. [4](#)
- [25] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman. Video-based face recognition using probabilistic appearance manifolds. In *Computer vision and pattern recognition, 2003. proceedings. 2003 ieee computer society conference on*, volume 1, pages I–I. IEEE, 2003. [3](#)
- [26] H. Li, G. Hua, X. Shen, Z. Lin, and J. Brandt. Eigen-pep for video face recognition. In *Asian Conference on Computer Vision*, pages 17–33. Springer, 2014. [3](#)
- [27] W.-A. Lin, J.-C. Chen, and R. Chellappa. A proximity-aware hierarchical clustering of faces. *arXiv preprint arXiv:1703.04835*, 2017. [7](#)
- [28] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang. Targeting ultimate accuracy: Face recognition via deep embedding. *arXiv preprint arXiv:1506.07310*, 2015. [2](#), [9](#)
- [29] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. [6](#)
- [30] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. *arXiv preprint arXiv:1704.08063*, 2017. [3](#), [9](#)
- [31] I. Masi, S. Rawls, G. Medioni, and P. Natarajan. Pose-aware face recognition in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4838–4846, 2016. [10](#)
- [32] I. Masi, A. T. Trn, T. Hassner, J. T. Leksut, and G. Medioni. Do we really need to collect millions of faces for effective face recognition? In *European Conference on Computer Vision*, pages 579–596. Springer, 2016. [10](#)
- [33] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney, et al. Iarpa janus benchmark-c: Face dataset and protocol. [2](#), [10](#), [11](#), [12](#), [13](#)
- [34] A. J. Otoole, K. A. Deffenbacher, D. Valentini, and H. Abdi. Structural aspects of face recognition and the other-race effect. *Memory & Cognition*, 22(2):208–224, 1994. [12](#)
- [35] C. J. Parde, C. Castillo, M. Q. Hill, Y. I. Colon, S. Sankaranarayanan, J.-C. Chen, and A. J. O’Toole. Deep convolutional neural network features and the original image. *arXiv preprint arXiv:1611.01751*, 2016. [1](#)
- [36] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman. A compact and discriminative face track descriptor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1693–1700, 2014. [3](#)
- [37] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, volume 1, page 6, 2015. [2](#), [3](#), [9](#), [10](#), [11](#), [12](#)
- [38] R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. [6](#)
- [39] R. Ranjan, S. Sankaranarayanan, A. Bansal, N. Bodla, J.-C. Chen, V. M. Patel, C. D. Castillo, and R. Chellappa. Deep learning for understanding faces: Machines may be just as good, or better, than humans. *IEEE Signal Processing Magazine*, 35(1):66–83, 2018. [2](#)
- [40] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa. An all-in-one convolutional neural network for face analysis. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 17–24. IEEE, 2017. [3](#), [8](#), [9](#), [10](#), [11](#)
- [41] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. [9](#)
- [42] S. Sankaranarayanan, A. Alavi, C. D. Castillo, and R. Chellappa. Triplet probabilistic embedding for face verification and clustering. In *Biometrics Theory, Applications and Systems (BTAS), 2016 IEEE 8th International Conference on*, pages 1–8. IEEE, 2016. [2](#), [3](#), [10](#), [11](#)
- [43] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015. [1](#), [2](#), [3](#), [9](#), [11](#), [12](#)
- [44] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are

- sparse, selective, and robust. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2892–2900, 2015. 1, 2, 9
- [45] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017. 11
- [46] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014. 2, 3, 9
- [47] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2273–2286, 2011. 3
- [48] D. Wang, C. Otto, and A. K. Jain. Face search at scale: 80 million gallery. *arXiv preprint arXiv:1507.07242*, 2015. 10
- [49] Y. Wen, Z. Li, and Y. Qiao. Latent factor guided convolutional neural networks for age-invariant face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4893–4901, 2016. 2
- [50] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, pages 499–515. Springer, 2016. 2, 3, 4, 7, 8, 9, 10, 11
- [51] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. Adams, T. Miller, N. Kalka, A. K. Jain, J. A. Duncan, K. Allen, et al. Iarpa janus benchmark-b face dataset. In *CVPR Workshop on Biometrics*, 2017. 2, 10, 11, 12
- [52] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 529–534. IEEE, 2011. 9, 10
- [53] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431*, 2016. 9
- [54] L. Xiong, J. Karlekar, J. Zhao, J. Feng, S. Pranata, and S. Shen. A good practice towards top performance of face recognition: Transferred deep feature fusion. *arXiv preprint arXiv:1704.00438*, 2017. 2, 10
- [55] H. Xu, J. Zheng, A. Alavi, and R. Chellappa. Learning a structured dictionary for video-based face recognition. In *2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016*, pages 1–9, 2016. 3
- [56] H. Xu, J. Zheng, A. Alavi, and R. Chellappa. Template regularized sparse coding for face verification. In *23rd International Conference on Pattern Recognition, ICPR 2016*, pages 1448–1454, 2016. 2
- [57] J. Yang, P. Ren, D. Chen, F. Wen, H. Li, and G. Hua. Neural aggregation network for video face recognition. *arXiv preprint arXiv:1603.05474*, 2016. 2, 3, 9, 10

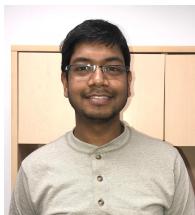


Hongyu Xu received the BEng degree in Electrical Engineering from the University of Science and Technology of China in 2012 and the MS degree in Electrical and Computer Engineering from University of Maryland, College Park in 2016. He is currently a research assistant in the Institute for Advanced Computer Studies at the University of Maryland, College Park. His research interests include object detection, face recognition, object classification, and domain adaptation.

Swami Sankaranarayanan is a Ph.D. candidate at University of Maryland College Park. He received his M.S degree from TU Delft, in 2012. His research interests include face analysis and adversarial machine learning.

Jun-Cheng Chen is a postdoctoral research fellow at the University of Maryland Institute for Advanced Computer Studies (UMIACS). He received the Ph.D. degree from University of Maryland, College Park in 2016. His current research interests include computer vision and machine learning with applications to face recognition and facial analysis. He was a recipient of ACM Multimedia best technical full paper award, 2006.

Carlos D. Castillo is an assistant research scientist at the University of Maryland Institute for Advanced Computer Studies (UMIACS). He received the Ph.D. degree from University of Maryland, College Park in 2012. His current research interests include stereo matching, multi-view geometry, face detection, alignment and recognition.



Rajeev Ranjan received the B.Tech. degree in Electronics and Electrical Communication Engineering from Indian Institute of Technology Kharagpur, India, in 2012. He is currently a Research Assistant at University of Maryland College Park. His research interests include face detection, face recognition and machine learning. He received Best Poster Award at IEEE BTAS 2015. He is a recipient of UMD Outstanding Invention of the Year award 2015, in the area of Information Science. He received the 2016 Jimmy Lin Award for Invention.

Ankan Bansal is a Ph.D. degree student at the University of Maryland, College Park. He received his B.Tech. and M.Tech. degrees in Electrical Engineering from the Indian Institute of Technology, Kanpur, in 2015. His research interests include multi-modal learning, action understanding, and face analysis.



Rama Chellappa is a Distinguished University Professor, a Minta Martin Professor of Engineering and Chair of the ECE department at the University of Maryland. Prof. Chellappa received the K.S. Fu Prize from the International Association of Pattern Recognition (IAPR). He is a recipient of the Society, Technical Achievement and Meritorious Service Awards from the IEEE Signal Processing Society and four IBM faculty Development Awards. He also received the Technical Achievement and Meritorious Service Awards from the IEEE Computer Society. At UMD, he received college and university level recognitions for research, teaching, innovation and mentoring of undergraduate students. In 2010, he was recognized as an Outstanding ECE by Purdue University. Prof. Chellappa served as the Editor-in-Chief of PAMI. He is a Golden Core Member of the IEEE Computer Society, served as a Distinguished Lecturer of the IEEE Signal Processing Society and as the President of IEEE Biometrics Council. He is a Fellow of IEEE, IAPR, OSA, AAAS, ACM and AAAI and holds four patents.