Ring loss: Convex Feature Normalization for Face Recognition

Yutong Zheng, Dipan K. Pal and Marios Savvides
Department of Electrical and Computer Engineering
Carnegie Mellon University
{yutongzh, dipanp, marioss}@andrew.cmu.edu

Abstract

We motivate and present Ring loss, a simple and elegant feature normalization approach for deep networks designed to augment standard loss functions such as Softmax. We argue that deep feature normalization is an important aspect of supervised classification problems where we require the model to represent each class in a multi-class problem equally well. The direct approach to feature normalization through the hard normalization operation results in a non-convex formulation. Instead, Ring loss applies soft normalization, where it gradually learns to constrain the norm to the scaled unit circle while preserving convexity leading to more robust features. We apply Ring loss to large-scale face recognition problems and present results on LFW, the challenging protocols of IJB-A Janus, Janus CS3 (a super-set of IJB-A Janus), Celebrity Frontal-Profile (CFP) and MegaFace with 1 million distractors. Ring loss outperforms strong baselines, matches state-of-the-art performance on IJB-A Janus and outperforms all other results on the challenging Janus CS3 thereby achieving state-of-the-art. We also outperform strong baselines in handling extremely low resolution face matching.

1. Introduction

Deep learning has demonstrated impressive performance on a variety of tasks. Arguably the most important task, that of supervised classification, has led to many advancements. Notably, the use of deeper structures [21, 23, 7] and more powerful loss functions [6, 19, 26, 24, 15] have resulted in far more robust feature representations. There has also been more attention on obtaining better-behaved gradients through normalization of batches or weights [9, 1, 18].

One of the most important practical applications of deep networks with supervised classification is face recognition. Robust face recognition poses a huge challenge in the form of very large number of classes with relatively few samples per class for training with significant nuisance transformations. A good understanding of the challenges in this task results
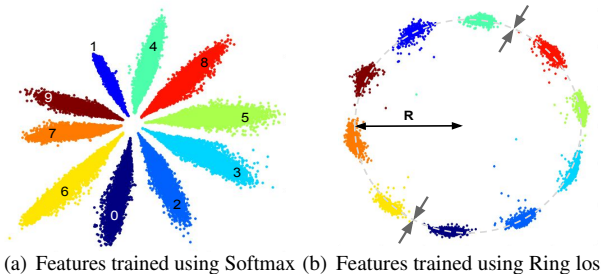

(a) Features trained using Softmax (b) Features trained using Ring loss

Figure 1: Sample MNIST features trained using (a) Softmax and (b) Ring loss on top of Softmax. Ring loss uses a convex norm constraint to gradually enforce normalization of features to a learned norm value R. This results in features of equal length while mitigating classification margin imbalance between classes. Softmax achieves 98.97 % accuracy on MNIST, whereas Ring loss achieves 99.34 % demonstrating the superior performance of the network learned normalized features.

in a better understanding of the core problems in supervised classification, and in general representation learning. However, despite the impressive attention on face recognition tasks over the past few years, there are still many gaps towards such an understanding. Notably, the need and practice of feature normalization. Normalization of features has recently been discovered to provide significant improvement in performance which implicitly results in a cosine embedding [17, 25]. However, direct normalization in deep networks explored in these works results in a non-convex formulation resulting in local minima generated by the loss function itself. It is important to preserve convexity in loss functions for more effective minimization of the loss given that the network optimization itself is non-convex. In a separate thrust of work, cosine similarity has also been very recently explored for supervised classification [16, 4]. Nonetheless, a concrete justification and principled motivation for the *need* for normalizing the features itself is also lacking.

Contributions. In this work, we propose Ring loss, a simple and elegant approach to normalize all sample features through a convex augmentation of the primary loss function (such as Softmax). The value of the target norm is also learnt during training. Thus, the only hyperparameter in Ring loss is the loss weight w.r.t to the primary loss func-
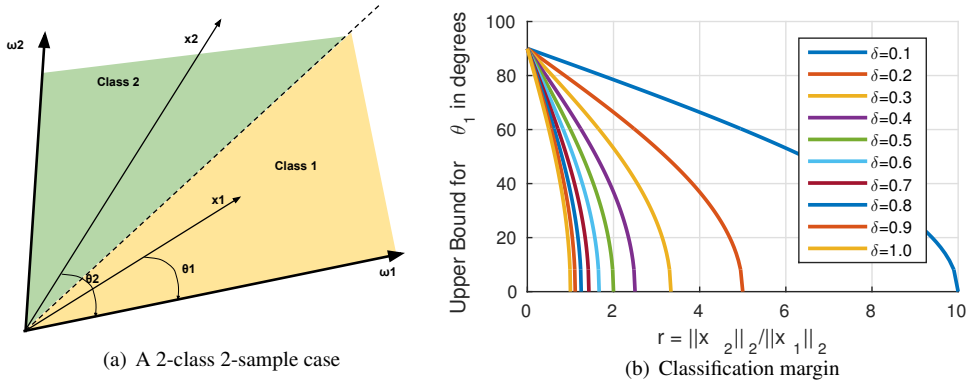
(a) A 2-class 2-sample case

(b) Classification margin

Figure 2: (a) A simple case of binary classification. The shaded regions (yellow, green) denote the classification margin (for class 1 and 2). (b) Angular classification margin for θ_1 for different $\delta = \cos\theta_2$.

tion. We provide an analytical justification illustrating the benefits of feature normalization and thereby cosine feature embeddings. Feature matching during testing in face recognition is typically done through cosine distance creating a gap between testing and training protocols which do not utilize normalization. The incorporation of Ring loss during training eliminates this gap. Ring loss is differentiable that allows for seamless and simple integration into deep architectures trained using gradient based methods. We find that Ring loss provides consistent improvements over a large range of its hyperparameter when compared to other baselines in normalization and indeed other losses proposed for face recognition in general. Interestingly, we also find that Ring loss helps in being robust to lower resolutions through the norm constraint.

2. Ring loss: Convex Feature Normalization

2.1. Intuition and Motivation.

There have been recent studies on the use of norm constraints right before the Softmax loss [25, 17]. However, the formulations investigated are non-convex in the feature representations leading to difficulties in optimization. Further, there is a need for better understanding of the benefits of normalization itself. Wang *et.al.* [25] argue that the 'radial' nature of the Softmax features is not a useful property, thereby cosine similarity should be preferred leading to normalized features. A concrete reason was, however, not provided. Ranjan *et.al.* [17] show that the Softmax loss encodes the quality of the data (images) into the norm thereby deviating from the ultimate objective of learning a good representation purely for classification.[1] Therefore for better classification, normalization forces the network to be invariant to such details. This is certainly not the entire story and in fact overlooks some key properties of feature normalization. We now motivate Ring loss with three arguments. 1)

We show that the norm constraint is beneficial to maintain a balance between the angular classification margins of multiple classes. 2) It removes the disconnect between training and testing metrics. 3) It minimizes test errors due to angular variation due to low norm features.

The Angular Classification Margin Imbalance. Consider a binary classification task with two feature vectors x_1 and x_2 from class 1 and 2 respectively, extracted using some model (possibly a deep network). Let the classification weight vector for class 1 and 2 be w_1, w_2 respectively (potentially Softmax). An example arrangement is shown in Fig. 2(a). Then in general, in order for the class 1 vector w_1 to pick x_1 *and not* x_2 for correct classification, we require $w_1^T x_1 > w_1^T x_2 \Rightarrow \|x_1\|_2 \cos\theta_1 > \|x_2\|_2 \cos\theta_2$[2]. Here, θ_1 and θ_2 are the angles between the weight vector w_1 (class 1 vector only) and x_1, x_2 respectively[3]. We call the feasible set (range for θ_1) for this inequality to hold as the *angular classification margin*. Note that it is also a function of θ_2. Setting $\frac{\|x_2\|_2}{\|x_1\|_2} = r$, we observe $r > 0$ and that for correct classification, we need $\cos\theta_1 > r\cos\theta_2 \Rightarrow \theta_1 < \cos^{-1}(r\cos\theta_2)$ since $\cos\theta$ is a decreasing function between $[-1, 1]$ for $\theta \in [0, \pi]$. This inequality needs to hold true for any θ_2. Fixing $\cos\theta_2 = \delta$, we have $\theta_1 < \cos^{-1}(r\delta)$. From the domain constraints of $\cos^{-1}$, we have $-1 \leq r\delta \leq 1 \Rightarrow \frac{-1}{\delta} \leq r \leq \frac{1}{\delta}$. Combining this inequality with $r > 0$, we have $0 < r \leq \frac{1}{|\delta|} \Rightarrow \|x_2\|_2 \leq \frac{1}{\delta}\|x_1\|_2 \ \forall \delta \in (0\ 1]$. For our purposes it suffices to only look at the case $\delta > 0$ since the $\delta < 0$ doesn't change the inequality $-1 \leq r\delta \leq 1$ and is more interesting.

Discussion on the angular classification margin. We plot the upper bound on θ_1 (*i.e.* $\cos^{-1}(r\cos\theta_2)$) for a range of δ ([0.1, 1]) and the corresponding range of r. Fig. 2(b) showcases the plot. Consider $\delta = 0.1$ which implies that

---

[1] We in fact, find in our pilot study that the Softmax features also encode the 'difficulty' of the class.

[2] Although, it is more common to in turn investigate competition between two weight vectors to classify a single sample, we find that this alternate perspective provide some novel and interesting insights.

[3] Note that this reasoning is applicable to any loss function trying to enforce this inequality in some form.
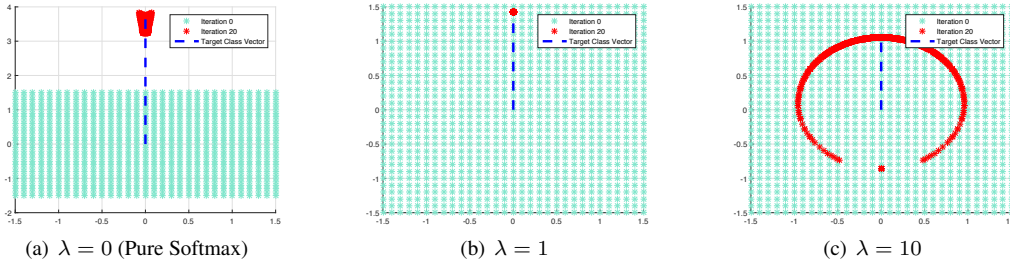
| (a) $\lambda = 0$ (Pure Softmax) | (b) $\lambda = 1$ | (c) $\lambda = 10$ |

Figure 3: Ring loss Visualizations: (a), (b) and (c) show the final convergence of the samples (for varying λ). The blue-green dots are the samples before the gradient update and the red dots are the same samples after the update. The dotted blue vector is the target class direction. $\lambda = 0$ fails to converge and does not constrain the norm whereas $\lambda = 10$ takes very small steps towards Softmax gradients. A good balance is achieved at $\lambda = 1$. In our large scale experiments, a large range of λ achieves this balance.

the sample x_2 has a large angular distance from w_1 (about $85°$). This case is favorable in general since one would expect a lower probability of x_2 being classified to class 1. However, we see that as r increases (difference in norm of x_1, x_2), the classification margin for x_1 decreases from $90°$ to eventually $0°$. In other terms, as the norm of x_2 increases w.r.t x_1, the angular margin for x_1 to be classified correctly while rejecting x_2 by w_1, decreases. The difference in norm ($r > 1$) therefore will have an adverse effect during training by effectively enforcing smaller angular classification margins for classes with smaller norm samples. This also leads to lop-sided classification margins for multiple classes due to the difference in class norms as can be seen in Fig. 1(a). This effect is only magnified as δ increases (or the sample x_2 comes closer to w_1). Fig. 2(b) shows that the angular classification margin decreases much more rapidly as δ increases. However, $r < 1$ leads to a larger margin and seems to be beneficial for classifying class 1 (as compared to $r > 1$). One might argue that this suggests that the $r < 1$ should be enforced for better performance. However, note that the same reasoning applies correspondingly to class 2, where we want to classify x_2 to w_2 while rejecting x_1. This creates a trade off between performance on class 1 versus class 2 based on r which also directly scales to multi-class problems. In typical recognition applications such as face recognition, this is not desirable. Ideally, we would want to represent all classes equally well. Setting $r = 1$ or constraining the norms of the samples from both classes to be the same ensures this.

Effects of Softmax on the norm of MNIST features. We qualitatively observe the effects of vanilla Softmax on the norm of the features (and thereby classification margin) on MNIST in Fig. 1(a). We see that digits 3, 6 and 8 have large norm features which are typically the classes that are harder to distinguish between. Therefore, we observe $r < 1$ for these three 'difficult' classes (w.r.t to the other 'easier' classes) thereby providing a larger angular classification margin to the three classes. On the other hand, digits 1, 9 and 7 have lower norm corresponding to $r > 1$ w.r.t to the other classes, since the model can afford to decrease the margin for these 'easy' classes as a trade off. We also observe that arguably most easily distinguishable class, digit

1, has the lowest norm thereby the highest r. On the other hand, Fig. 1(b) showcases the features learned using Softmax augmented with our proposed Ring loss, which forces the network to learn feature normalization through a convex formulation thereby mitigating this imbalance in angular classification margins.

Regularizing Softmax loss with the norm constraint. The ideal training scenario for a system testing under the cosine metric would be where all features pointing in the same *direction* have the same loss. However, this is not true for the most commonly used loss function, Softmax and its variants (FC layer combined with the softmax function and the cross-entropy loss). Assuming that the weights are normalized, i.e. $\|w_k\| = 1$, the Softmax loss for feature vector $\mathcal{F}(\mathbf{x}_i)$ can be expressed as (for the correct class y_i):

$$L_{SM} = -\log \frac{\exp w_k \mathcal{F}(\mathbf{x}_i)}{\sum_{k'=1}^{K} \exp w_{k'} \mathcal{F}(\mathbf{x}_i)} F \qquad (1)$$

$$= -\log \frac{\exp \|\mathcal{F}(\mathbf{x}_i)\| \cos \theta_{ki}}{\sum_{k'=1}^{K} \exp \|\mathcal{F}(\mathbf{x}_i)\| \cos \theta_{k'i}} \qquad (2)$$

Clearly, despite having the same direction, two features with different norms have different losses. From this perspective, the straightforward solution to regularize the loss and remove the influence of the norm is to normalize the features before Softmax as explored in l_2-constrained Softmax [17]. However, this approach is effectively a projection method, i.e. it calculates the loss as if the features are normalized to the same scale, while the actual *network* does not *learn* to normalize features.

The need for features normalization in feature space. As an illustration, consider the training and testing set features trained by vanilla Softmax, of the digit 8 from MNIST in Fig. 4. Fig. 4(a) shows that at the end of training, the features are well behaved with a large variation in the norm of the features with a few samples with low norm. However, Fig. 4(b) shows that that the features for the test samples are much more erratic. There is a similar variation in norm but now most of the low norm features have huge variation in *angle*. Indeed, variation in samples for lower norm features translates to a larger variation in angle than the same
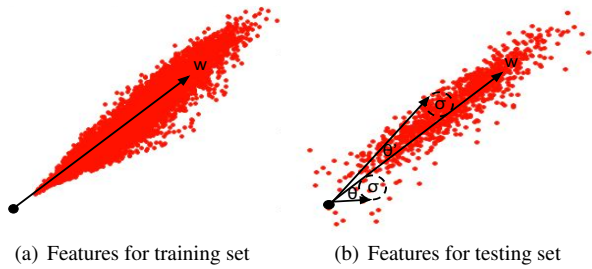
(a) Features for training set   (b) Features for testing set

Figure 4: MNIST features for digit 8 trained using vanilla Softmax loss.

Figure 5: Ring loss improves MNIST testing accuracy across all classes by reducing inter-class norm variance. Norm Ratio is the ratio between average class norm and average norm of all features.

2.2. Ring loss

Ring loss Definition. Ring loss L_R is defined as

$$L_R = \frac{\lambda}{2m} \sum_{i=1}^{m} (\|\mathcal{F}(\mathbf{x}_i)\|_2 - R)^2 \tag{4}$$

where $\mathcal{F}(\mathbf{x}_i)$ is the deep network feature for the sample $\mathbf{x}_i$. Here, R is the target norm value which is also learned and λ is the loss weight enforcing a trade-off between the primary loss function. m is the batch-size. The square on the norm difference helps the network to take larger steps when the norm of a sample is too far off from R leading to faster convergence. The corresponding gradients are as follows.

$$\frac{\partial L_R}{\partial R} = -\frac{\lambda}{m} \sum_{i=1}^{m} (\|\mathcal{F}(\mathbf{x}_i)\|_2 - R) \tag{5}$$

$$\frac{\partial L_R}{\partial \mathcal{F}(\mathbf{x}_i)} = \frac{\lambda}{m} \left(1 - \frac{R}{\|\mathcal{F}(\mathbf{x}_i)\|_2}\right) \mathcal{F}(\mathbf{x}_i) \tag{6}$$

Ring loss (L_R) can be used along with any other loss function such as Softmax or large-margin Softmax [14]. The loss encourages norm of samples being value R (a learned parameter) rather than explicit enforcing through a hard normalization operation. This approach provides informed gradients towards a better minimum which helps the network to satisfy the normalization constraint. The network therefore, learns to normalize the features using *model weights themselves* (rather than needing an explicit non-convex normalization operation as in [17], or batch normalization [9]). In contrast and in connection, batch normalization [9] enforces the scaled normal distribution for each element in the feature independently. This does not constrain the overall norm of the feature to be equal across all samples and neither addresses the class imbalance problem. As shown in Fig. 5, Ring loss stabilizes the feature norm across all classes, and, in turn, rectifies the classification imbalance for Softmax to perform better overall.

Ring loss Convergence Visualizations. To illustrate the effect of the Softmax loss augmented with the enforced soft-normalization, we conduct some analytical simulations. We generate a 2D mesh of points from $(-1.5, 1.5)$ in x,y-axis. We then compute the gradients of Ring loss ($R = 1$) assuming the dottef blue vertical line (see Fig. 3)

for higher norm samples features. This translates to higher errors in classification under the cosine metric (as is common in face recognition). This is yet another motivation to normalize features during training. Forcing the network to *learn* to normalize the features helps to mitigate this problem during test wherein the network learns to work in the normalized feature space. A related motivation to feature normalization was proposed by Ranjan *et.al.* [17] wherein it was argued that low resolution of an image results in a low norm feature leading to test errors. Their solution to project (not implicitly learn) the feature to the scaled unit hypersphere was also aimed at handling low resolution. We find in our large scale experiment with low resolution images (see Exp. 6 Fig. 8) that soft normalization by Ring loss achieves better results. In fact hard projection method by l_2-constrained Softmax [17] performs worse than Softmax for a downsampling factor of 64.

Incorporating the norm constraint as a convex problem. Identifying the need to normalize the sample features from the network, we now formulate the problem. We define L_S as the primary loss function (for instance Softmax loss). Assuming that $\mathcal{F}$ provides deep features for a sample x as $\mathcal{F}(x)$, we would like to minimize the loss subject to the normalization constraint as follows,

$$\min L_S(\mathcal{F}(x)) \;\; s.t. \;\; \|\mathcal{F}(x)\|_2 = R \tag{3}$$

Here, R is the scale constant that we would like the features to be normalized to. This is the exact formulation recently studied and implemented by [17, 25]. Note that this problem is non-convex in $\mathcal{F}(x)$ since the set of feasible solutions is itself non-convex due to the norm equality constraint. Approaches which use standard SGD while ignoring this critical point would not be providing feasible solutions to this problem thereby, the network $\mathcal{F}$ would not learn to output normalized features. Indeed, the features obtained using this straightforward approach are not normalized as was found in Fig. 3b in [17] compared to our approach (Fig. 1(b)). One naive approach to get around this problem would be to relax the norm equality constraint to an inequality. This objective will now be convex, however it does not necessarily enforce equal norm features. In order to incorporate the formulation as a convex constraint, the following form is directly useful as we find below.
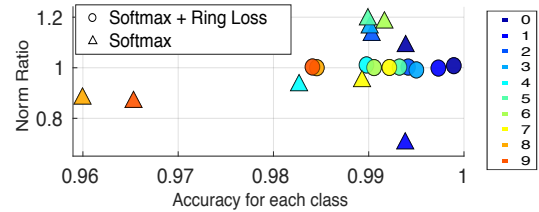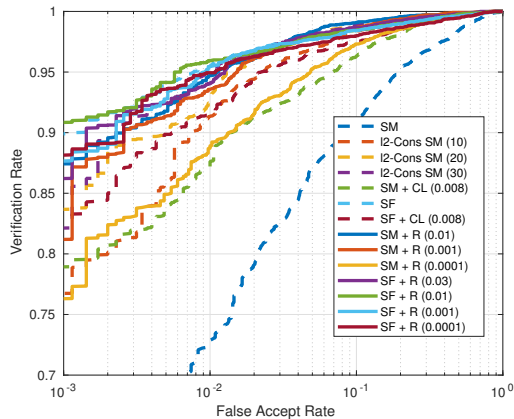
Figure 6: ROC curves on the CFP Frontal vs. Profile verification protocol. For all Figures and Tables, SM denotes Softmax, SF denotes SphereFace [14],l2-Cons SM denotes [17], + CL denotes Center Loss augmentation [26] and finally + R denotes Ring loss augmentation. Numbers in bracket denote value of hyperparameter (loss weight), *i.e.* α for [17], λ for Center loss and Ring loss.

as the target class and update each point with a fixed step size for 20 steps. We run the simulation for $\lambda = \{0, 1, 10\}$. Note that $\lambda = 0$ represents pure Softmax. Fig. 3 depicts the results of these simulations. Sub-figures (a), (b) and (c) in Fig. 3 show the initial points on the mesh grid (light green) and the final updated points (red). For pure Softmax ($\lambda = 0$), we see that the updates increases norm of the samples and moreover they do not converge. For a reasonable loss weight of $\lambda = 1$, Ring loss gradients can help the updated points converge much faster in the same number of iterations. For heavily weighted Ring loss with $\lambda = 10$, we see that the gradients force the samples to a unit norm since R was set to 1 while overpowering Softmax gradients. These figures suggest that there exists a trade off enforced by λ between the Softmax loss L_S and the normalization loss. We observe similar trade-offs in our experiments.

3. Experimental Validation

We benchmark Ring loss on large scale face recognition tasks while augmenting two loss functions. The first one is the ubiquitous Softmax, and the second being a successful variant of Large-margin Softmax [15] called SphereFace [14]. We present results on five large-scale benchmarks of LFW [8], IARPA Janus Benchmark IJB-A [11], Janus Challenge Set 3 (CS3) dataset (which is a super set of the IJB-A Janus dataset), Celebrities Frontal-Profile (CFP) [20] and finally the MegaFace dataset [10]. We also present results of Ring loss augmented Softmax features on low resolution images from Janus CS3 to showcase resolution robust face matching.

Implementation Details. For all the experiments in this paper, we usethe ResNet 64 (Res64) layer architecture from

Liu *et. al.* [14]. For Center loss, we utilized the code repository online and used the best hyperparameter setting reported[4]. The l_2-constrained Softmax loss was implemented follwing [17] by integrating a normalization and scaling layer[5] before the last fully-connected layer. For experiments with L-softmax [15] and SphereFace [14], we used the publicly available Caffe implementation. The Resnet 64 layer (Res64) architecture results in a feature dimension of 512 (at the fc5 layer), which is used for matching using the cosine distance. Ring loss and Center loss are both applied on this feature *i.e.* to the output of the fc5 layer. All models were trained on the MS-Celeb 1M dataset [5]. The dataset was cleaned to remove potential outliers within each class and also noisy classes before training. To clean the dataset we used a pretrained model to extract features from the MS-Celeb 1M dataset. Then, classes that had variance in the MSE, between the sample features and the mean feature of that class, above a certain threshold were discarded. Following this, from the filtered classes, images that have their MSE error between their feature vector and the class mean feature vector higher than a threshold are discarded. After this procedure, we are left with about 31,000 identities and about 3.5 million images. The learning rate was initialized to 0.1 and then decreased by a factor of 10 at 80K and 150K iterations for a total of 165K iterations. All models evaluated were the 165K iteration model[6].

Preprocessing. All faces were detected and aligned using [27] which provided landmarks for the two eye centers, nose and mouth corners (5 points). Since MS-Celeb1M, IJB-A Janus and Janus CS3 have harder faces we use a robust detector *i.e.* CMS-RCNN [29] to detect faces and a fast landmarker that is robust to pose [2]. The faces were then aligned using a similarity transformation and were cropped to 112×96 in the RGB format. The pixel level activations were normalized by subtracting 127.5 and then dividing by 128. For failed detections, the training set images are ignored. In the case of testing, ground truth landmarks were used from the corresponding dataset.

Exp 1. Testing Benchmark: LFW. The LFW [8] database contains about 13,000 images for about 1680 subjects with a total of 6,000 defined matches. The primary nuisance transformations are illumination, pose, color jittering and age. As the field has progressed, LFW has been considered to be saturated and prone to spurious minor variances in performance (in the last % of accuracy) owing to the small size of the protocol. Small differences in accuracy on this protocol do not accurately reflect the generalizing

---

[4]see `https://github.com/ydwen/caffe-face.git`

[5]see `https://github.com/craftGBD/caffe-GBD`. In our experiments, for $\alpha = 50$ the gradients exploded due the relatively deep Res64 architecture and learning α initialized at 30 did not converge.

[6]For all Tables, results reported after double horizontal lines are from models trained during our study. The results above the lines reported directly from the paper as cited.
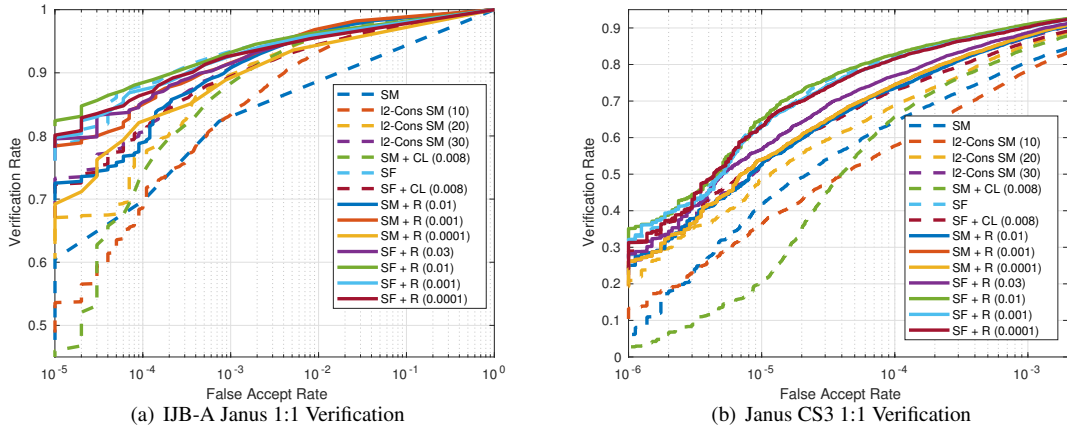
Figure 7: ROC curves on the (a) IJB-A Janus 1:1 verification protocol and the (b) Janus CS3 1:1 verification protocol. For all Figures and Tables, SM denotes Softmax, SF denotes SphereFace [14],l2-Cons SM denotes [17], + CL denotes Center Loss augmentation [26] and finally + R denotes Ring loss augmentation. Numbers in bracket denote value of hyperparameter (loss weight), *i.e.* α for [17], λ for Center loss and Ring loss.

capabilities of a high performing model. Nonetheless, as a benchmark, we report performance on this dataset.

Results: LFW. Table 1 showcases the results on the LFW protocol. We find that for Softmax (SM + R), Ring loss normalization seems to significantly improve performance (up from 98.47% to **99.52%** using Ring loss with $\lambda = 0.01$). We find similar trends while using Ring loss with SphereFace. The LFW accuracy of SphereFace improves from 99.47% to **99.50%**. We note that since even our baselines are high performing, there is a lot of variance in the results owing to the small size of the LFW protocol (just 6000 matches compared to about 8 million matches in the Janus CS3 protocol which shows clearer trends). Indeed we find clearer trends with MegaFace, IJB-A and CS3 all of which are orders of magnitude larger protocols.

Exp 2. Testing Benchmark: IJB-A Janus. IJB-A [11] is a challenging dataset which consists of 500 subjects with extreme pose, expression and illumination with a total of 25,813 images. Each subject is described by a template instead of a single image. This allows for score fusion techniques to be developed. The setting is suited for applications which have multiple sources of images/video frames. We report results on the 1:1 template matching protocol containing 10 splits with about 12,000 pair-wise template matches each resulting in a total of 117,420 template matches. The template matching score for two templates T_i, T_j is determined by using the following formula, $S(T_i, T_j) = \sum_{\gamma=1}^{K} \frac{\sum_{t_a \in T_i, t_b \in T_j} s(t_a, t_b) \exp \gamma s(t_a, t_b)}{\sum_{t_a \in T_i, t_b \in T_j} \exp \gamma s(t_a, t_b)}$ where $s(t_a.t_b)$ is the cosine similarity score between images t_a, t_b and $K = 8$.

Results: IJB-A Janus. Table. 3 and Fig. 7(a) present these results. We see that Softmax + Ring loss (0.001) outperforms Softmax by a large margin, particularly 60.52% verification rate compared to **78.41%** verification at 10^{-5} FAR.

Further, it outperforms Center loss [26] (46.01%) and l_2-constrained Softmax (73.29%) [17]. Although SphereFace performs better than Softmax + Ring loss, an augmentation by Ring loss boosts SphereFace's performance from 78.52% to **82.41%** verification rate for $\lambda = 0.01$. This matches the state-of-the-art reported in [17] which uses a 101-layer ResNext architecture despite our system using a much shallower 64-layer ResNet architecture. The effect of high λ akin to the effects simulated in Fig. 3 show in this setting for $\lambda = 0.03$ for SphereFace augmentation. We observe this trade-off in Janus CS3, CFP and MegaFace results as well. Nonetheless, we notice that Ring loss augmentation provides consistent improvements over a large range of λ for both Softmax and Sphereface. This is in sharp contrast with l_2-constrained Softmax whose performance varies significantly with α rendering it difficult to optimize. In fact for $\alpha = 10$, it performs worse than Softmax.

Exp 3. Testing Benchmark: Janus CS3. The Janus CS3 dataset is a super set of the IARPA IJB-A dataset. It contains about 11,876 still images and 55,372 video frames from 7,094 videos. For the CS3 1:1 template verification protocol there are a total of 1,871 subjects and 12,590 templates. The CS3 template verification protocol has over 8 million template matches which amounts to an extremely large number of template verifications. There are about 1,870 templates in the gallery and about 10,700 templates in the probe. The CS3 protocol being a super set of IJB-A, has a large number of extremely challenging images and faces. The challenging conditions range from extreme illumination, extreme pose to significant occlusion. For sample images, please refer to Fig. 4 and Fig. 10 in [12], Fig. 1 and Fig. 6 in [3]. Since the protocol is template matching, we utilize the same template score fusion technique we utilize in the IJB-A results with $K = 2$.

Results: Janus CS3. Table. 4 and Fig. 7(b) showcases

(a) Downsampling 4x   (b) Downsampling 16x   (c) Downsampling 25x   (d) Downsampling 36x   (e) Downsampling 64x
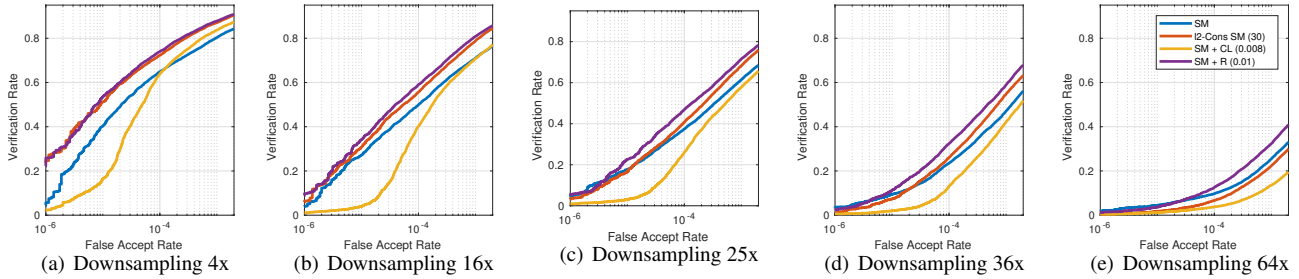
Figure 8: ROC curves for the downsampling experiment on Janus CS3. Ring loss (SM + R $\lambda = 0.01$) learns the most robust features, whereas l_2-constrained Softmax (l2-Cons SM $\alpha = 30$) [17] performs poorly (worse than the baseline Softmax) at very high downsampling factor of 64x.

our results on the CS3 dataset. We report verification rates (VR) at 10^{-3} through 10^{-6} FAR. We find that our Ring loss augmented Softmax model outperforms the previous best reported results on the CS3 dataset. Recall that the Softmax + Ring loss model (SM + R) was trained only on a subset of the MS-Celeb dataset and achieves a VR of 74.56% at 10^{-4} FAR. This is in contrast to Lin *et. al.* who train on MS-Celeb plus CASIA-WebFace (an additional 0.5 million images) and achieve 72.52 %. Further, we find that even though our baseline Sphereface Res64 model outperforms the previous state-of-the-art, our Ring loss augmented Sphereface model outperforms all other models to achieve high a VR of **82.74 %** at 10^{-4} FAR. At very low FAR of 10^{-6} our SF + R model achieves VR **35.18 %** which to the best of our knowledge is the state-of-the-art on the challenging Janus CS3. In accordance with the results on IJB-A Janus, Ring loss provides consistent improvements over large ranges of λ whereas l_2-constrained Softmax exhibits significant variation w.r.t. to its hyperparameter.

Exp 4. Testing Benchmark: MegaFace. The recently released MegaFace benchmark is extremely challenging which defines matching with a gallery of about 1 million distractors [10]. The aspect of face recognition that this database test is discrimination in the presence of very large number of distractors. The testing database contains two sets of face images. The distractor set contains 1 million distractor subjects (images). The target set contain 100K images from 530 celebrities.

Result: MegaFace. Table. 2 showcases our results. Even at this extremely large scale evaluation (evaluating Face-Scrub against 1 million), the addition of Ring loss provides significant improvement to the baseline approaches. The identification rate (%) for Softmax upon the addition of Ring loss ($\lambda = 0.001$) improves from 56.36% to a high **71.67%** and for SphereFace it improves from 74.95% to **75.22%** for a *single* patch model. This is higher than the single patch model reported in the orginal Sphereface paper (72.72% [14]). We outperform Center loss [26] augmenting both Softmax (67.24%) and Sphereface (71.15%). We find that though for MegaFace, l_2-constrained Softmax [17] for $\alpha = 30$ achieves 72.22%, there is yet again significant varia-

Table 1: Accuracy (%) on LFW.

Method	Training Data	Accuracy (%)
FaceNet [19]	200M private	99.65
Deep-ID2+ [22]	CelebFace+	99.15
Range loss [28]	WebFace +Celeb1M(1.5M)	99.52
Baidu [13]	1.3M	99.77
Norm Face [25]	WebFace	99.19
SM	MS-Celeb	98.47
$l2$-Cons SM (30) [17]	MS-Celeb	99.55
$l2$-Cons SM (20) [17]	MS-Celeb	99.47
$l2$-Cons SM (10) [17]	MS-Celeb	99.45
SM + CL [26]	MS-Celeb	99.17
SF [14]	MS-Celeb	99.47
SF + CL [26, 14]	MS-Celeb	99.52
SM + R (0.01)	MS-Celeb	99.52
SM + R (0.001)	MS-Celeb	99.50
SM + R (0.0001)	MS-Celeb	99.28
SF + R (0.03)	MS-Celeb	99.48
SF + R (0.01)	MS-Celeb	99.43
SF + R (0.001)	MS-Celeb	99.42
SF + R (0.0001)	MS-Celeb	99.50

tion in performance that occurs due to a change in the hyper parameter α (66.20% for $\alpha = 10$ to 72.22% for $\alpha = 30$). Ring loss hyper parameter (λ), as we find again, is more easily tunable and manageable. This results in a smaller variance in performance for both Softmax and SphereFace augmentations.

Exp 5. Testing Benchmark: CFP Frontal vs. Profile. Recently the CFP (Celebrities Frontal-Profile) dataset was released to evaluate algorithms exclusively on frontal versus profile matches [20]. This small dataset has about 7,000 pairs of matches defined with 3,500 same pairs and 3,500 not-same pairs for about 500 different subjects. For sample images please refer to Fig. 1 of [20]. The dataset presents a challenge since each of the probe images is almost entirely profile thereby presenting extreme pose along with illumination and expression challenges.

Result: CFP Frontal vs. Profile. Fig. 6 showcases the ROC curves for this experiment whereas Table. 2 shows the

Method	Acc % (MegaFace)	10^{-3} (CFP)
SM	56.36	55.86
$l2$-Cons SM (30) [17]	72.22	82.14
$l2$-Cons SM (20) [17]	70.29	83.69
$l2$-Cons SM (10) [17]	66.20	76.77
SM + CL [26]	67.24	78.94
SF [14]	74.95	89.94
SF + CL [26, 14]	71.15	82.97
SM + R (0.01)	71.10	87.43
SM + R (0.001)	71.67	81.29
SM + R (0.0001)	69.41	76.30
SF + R (0.03)	73.05	86.23
SF + R (0.01)	74.93	**90.94**
SF + R (0.001)	**75.22**	87.69
SF + R (0.0001)	74.45	88.17

Table 2: Identification rates on MegaFace with 1 million distractors (Accuracy %) and Verification rates at 10^{-3} FAR for the CFP Frontal vs. Profile protocol.

Method	10^{-5}	10^{-4}	10^{-3}
$l2$-Cons SM* (101) [17]	-	87.9	93.7
$l2$-Cons SM* (101x) [17]	-	88.3	93.8
SM	60.52	69.69	83.10
$l2$-Cons SM (30) [17]	73.29	80.65	90.72
$l2$-Cons SM (20) [17]	67.63	76.88	89.89
$l2$-Cons SM (10) [17]	53.74	68.58	83.42
SM + CL [26]	46.01	74.10	88.32
SF [14]	78.52	88.0	**93.24**
SF + CL [26, 14]	72.35	81.11	89.26
SM + R (0.01)	72.53	79.1	90.8
SM + R (0.001)	78.41	85.0	91.5
SM + R (0.0001)	69.23	82.30	89.20
SF + R (0.03)	79.54	85.37	91.64
SF + R (0.01)	**82.41**	**88.5**	93.22
SF + R (0.001)	79.74	87.71	92.62
SF + R (0.0001)	80.13	86.34	92.57

Table 3: Verification % on the IJB-A Janus 1:1 verification protocol. $l2$-Cons SM* indicates the result reported in [17] which uses a 101 layer ResNet/ResNext architecture.

verification rates at 10^{-3} FAR. Ring loss (**87.43%**) provides consistent and significant boost in performance over Softmax (55.86%). We find however, SphereFace required more careful tuning of λ with $\lambda = 0.01$ (**90.94%**) outperforming the baseline. Further, Softmax and Ring loss with $\lambda = 0.01$ significantly outperforms all runs for l_2-constrained Softmax [17] (83.69). Thus, Ring loss helps in providing higher verification rates while dealing with frontal to highly off-angle matches thereby explicitly demonstrating robustness to pose variation.

Exp 6. Low Resolution Experiments on Janus CS3.

Method	10^{-6}	10^{-5}	10^{-4}	10^{-3}
Bodla *et. al.* Final1 [3]	-	-	69.81	82.89
Bodla *et. al.* Final2 [3]	-	-	68.45	82.97
Lin *et. al.* [12]	-	-	72.52	83.55
SM	6.16	42.03	64.52	80.86
$l2$-Cons SM (30) [17]	24.47	52.32	73.36	87.46
$l2$-Cons SM (20) [17]	21.14	48.82	68.84	85.34
$l2$-Cons SM (10) [17]	13.28	36.08	57.80	78.36
SM + CL [26]	2.88	20.87	65.71	84.55
SF [14]	28.51	63.92	82.29	90.58
SF + CL [26, 14]	28.99	53.36	72.91	86.14
SM + R (0.01)	25.17	52.60	73.56	87.50
SM + R (0.001)	26.62	54.13	74.56	87.93
SM + R (0.0001)	17.35	50.65	71.06	85.48
SF + R (0.03)	27.27	56.84	76.97	88.75
SF + R (0.01)	**35.18**	**65.02**	**82.74**	**90.99**
SF + R (0.001)	32.19	63.13	81.62	90.17
SF + R (0.0001)	32.01	63.12	81.57	90.24

Table 4: Verification % on the Janus CS3 1:1 verification protocol.

One of the main motivations for l_2-constrained Softmax was to handle images with varying resolution. Low resolution images were found to result in low norm features and vice versa. Ranjan *et.al.* [17] argued normalization (through l_2-constrained Softmax) would help deal with this issue. In order to test the efficacy of our alternate convex normalization formulation towards handling low resolution faces, we synthetically downsample Janus CS3 from an original size of (112×96) by a factor of 4x, 16x, 25x, 36x and 64x respectively (images were downsampled and resized back up using bicubic interpolation in order to fit the model). We run the Janus CS3 protocol and plot the ROC curves in Fig. 8. We find that the Ring loss helps Softmax features be more robust to resolution. Though l_2-constrained Softmax provides improvement over Softmax, it's performance is lower than Ring loss. Further, at extremely high downsampling of 64x, l_2-constrained Softmax in fact performs worse than Softmax, whereas Ring loss provides a clear improvement. Center loss fails early on at 16x. We therefore find that our simple convex soft normalization approach is more effective at arresting performance drop due to resolution in accordance with the motivation for as normalization presented in [17].

Conclusion. We motivate feature normalization in a principled manner and develop an elegant, simple and straight forward to implement convex approach towards that goal. We find that Ring loss consistently provides significant improvements over a large range of the hyperparameter λ. Further, it helps the network itself to learn normalization thereby being robust to a large range of degradations.

References

[1] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] C. Bhagavatula, C. Zhu, K. Luu, and M. Savvides. Faster than real-time facial alignment: A 3d spatial transformer network approach in unconstrained poses. *arXiv preprint arXiv:1707.05653*, 2017.

[3] N. Bodla, J. Zheng, H. Xu, J.-C. Chen, C. Castillo, and R. Chellappa. Deep heterogeneous feature fusion for template-based face recognition. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 586–595. IEEE, 2017.

[4] L. Chunjie, Z. Jianfeng, W. Lei, and Y. Qiang. Cosine normalization: Using cosine similarity instead of dot product in neural networks. *arXiv preprint arXiv:1702.05870*, 2017.

[5] 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.

[6] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 1735–1742. IEEE, 2006.

[7] 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.

[8] 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 07-49, University of Massachusetts, Amherst, October 2007.

[9] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[10] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4873–4882, 2016.

[11] B. F. Klare, B. Klein, E. Taborsky, 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.

[12] W.-A. Lin, J.-C. Chen, and R. Chellappa. A proximity-aware hierarchical clustering of faces. *arXiv preprint arXiv:1703.04835*, 2017.

[13] 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.

[14] 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.

[15] W. Liu, Y. Wen, Z. Yu, and M. Yang. Large-margin softmax loss for convolutional neural networks. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 507–516, 2016.

[16] Y. Liu, H. Li, and X. Wang. Learning deep features via congenerous cosine loss for person recognition. *arXiv preprint arXiv:1702.06890*, 2017.

[17] R. Ranjan, C. D. Castillo, and R. Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017.

[18] T. Salimans and D. P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 901–901, 2016.

[19] 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.

[20] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs. Frontal to profile face verification in the wild. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016.

[21] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[22] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, pages 1988–1996, 2014.

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[24] O. Tadmor, T. Rosenwein, S. Shalev-Shwartz, Y. Wexler, and A. Shashua. Learning a metric embedding for face recognition using the multibatch method. In *Advances In Neural Information Processing Systems*, pages 1388–1389, 2016.

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

[26] 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.

[27] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.

[28] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao. Range loss for deep face recognition with long-tail. *CoRR*, abs/1611.08976, 2016.

[29] C. Zhu, Y. Zheng, K. Luu, and M. Savvides. Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection. In *Deep Learning for Biometrics*, pages 57–79. Springer, 2017.