Copernican loss: Learning a Discriminative Cosine Embedding

Dipan K. Pal and Marios Savvides
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
{dipanp,marioss}@cmu.edu

Abstract

Euclidean embeddings are ubiquitous in machine learning applications in classificaiton settings. However, typical testing schemes (such as in face recognition) utilize the cosine similarity for feature matching, which creates a disconnect between training and testing scenarios. In this paper, we present Copernican loss, which learns a *discriminative cosine* embedding that addresses this gap. Copernican loss discriminates between all samples from different classes within a batch of size m with just m gradient computations (as opposed to $m^2 - m$ in previous work). We demonstrate the efficacy of the proposed approach with extensive experiments on CIFAR 10 and 100. Further, while keeping away from performance boosting preprocessing steps such as face alignment in training and testing, we match high performance on large scale face recognition datasets namely LFW and IJB-A Janus.

Introduction

Recent developments in deep neural networks have addressed a wide array of components. There has been considerable effort in developing deeper structures (Simonyan and Zisserman 2014; Szegedy et al. 2015; He et al. 2016) and more effective non-linearities (Goodfellow et al. 2013; Nair and Hinton 2010; He et al. 2015). Apart from structural developments, there have been many efforts in combating over-fitting and obtaining better gradients (Ioffe and Szegedy 2015; Srivastava et al. 2014; Salimans and Kingma 2016). Although fewer in number, there have also been recent studies recognizing the importance of stronger loss functions such as (Hadsell, Chopra, and LeCun 2006; Schroff, Kalenichenko, and Philbin 2015; Wen et al. 2016; Tadmor et al. 2016). Indeed, a robust loss function which encourages highly discriminate feature learning is a direct way to provide the network with more informed gradients towards the ultimate supervised task. A fully connected layer coupled with the cross-entropy loss and the softmax layer, together which we call the Softmax loss in this paper, is arguably the most prevalent loss function in practice. The Softmax loss has proved to be very versatile in use, and is able to provide reasonably good gradients owing to the well-behaved cross-entropy loss. A few recent studies have attempted to modify the Softmax loss in order to increase discrimination in terms of larger angular margin (Liu et al. 2016), or normalize the features going into Softmax thereby solving a non-convex problem (Ranjan, Castillo, and Chellappa 2017).

A different thrust towards obtaining highly discriminative features involves minimizing alternate loss functions or augmenting the Softmax with supplementary losses. Constrastive loss (Hadsell, Chopra, and LeCun 2006) and the Triplet loss (Schroff, Kalenichenko, and Philbin 2015) replace the Softmax loss with losses which focus on learning a discriminative embedding while trying to minimize intra-class variation in the learnt features. This is done by carefully sampling training pairs or triplet sets which leads to expensive hard-sample mining in large-scale applications. Center loss (Wen et al. 2016) on the other hand, is an approach which ignores hard-sample mining while only trying to minimize intra-class variation along with the Softmax loss. Another approach which was proposed to work on random batches is the Multi-batch estimator (Tadmor et al. 2016). Multi-batch is an example of the metric learning approach which for a batch size of m, utilizes all $m^2 - m$ pairs for a better estimate of the gradient. All these works mentioned operate in the l_2 or Euclidean space. As we explain in a later section and demonstrate in our experiments, although l_2 embeddings perform well in many applications, the performance gain they provide is limited in situations when the number of samples per class is high. In such situations, forcing all samples from a class to be converge towards each other in the l_2 sense is too difficult a task since it requires the network having to converge not only in angle but also the *norm* of the features. Further, during testing in typical supervised classification such as face recognition, the most common metric is the cosine distance which ignores the norm. This creates a disconnect between training and testing since the network learns a behavior (to constrain the norm as well) that is ignored during test. Such a framework is inefficient. Recently[1], COCO (Liu, Li, and Wang 2017), a form of cosine loss was proposed for person recognition. COCO minimizes intra-class variation towards a class center and maximizes inter-class variation of samples with the centers of *other classes* as opposed to the global batch center which significantly raises computational complexity. The approach uses hard normalization and is similar

[1] (Liu, Li, and Wang 2017) was published at the time of writing this manuscript.

to recent other studies (Ranjan, Castillo, and Chellappa 2017; Wang et al. 2017), all of which formulate a non-convex constraint.

Contributions. Our proposed Copernican loss has two important properties. 1) It is designed to augment the standard Softmax loss while *explicitly* minimizing intra-class variation and *simultaneously* maximizing inter-class variation. 2) It operates using the *cosine distance* and thereby directly affects angles leading to a cosine embedding which removes the disconnect between training and testing. This improves efficiency since more of the model complexity is utilized to learn a more discriminative embedding rather than learning to constrain the norm. Copernican loss does not require hard sample mining or data augmentation of any kind, and can be effectively minimized using SGD on random mini-batches. It only needs to maintain a center for each class called the Planet center and computes the mean of the mini-batch called the Sun center (humoring the Copernican analogy of the solar system). In order to minimize *intra-class* variation, it minimizes the cosine distance of the samples to their corresponding Planet centers. In order to *discriminate* between the samples within a mini-batch, Copernican loss maximizes the cosine distance of the samples away from the mean of the mini-batch called (Sun center). This eliminates the need to compute a pair-wise gradients such as the Multi-batch (Tadmor et al. 2016) while providing the similarly discriminative gradients in a more efficient manner.

Copernican loss: Learning a Discriminative Cosine Embedding

Motivation and Intuition

The Need for Simultaneous Discrimination and Invariance. Learning robust features is a key problem in supervised learning. Robustness here refers to two specific properties of a useful feature. 1) *Invariance* to intra-class nuisance transformations while being 2) *discriminative* to inter-class transformations. Although there exist loss functions prevalent in practice *implicitly* optimize for this criteria, such as the Softmax loss [2] and negative log likelihood, the features learnt using pure forms of the loss functions are not robust enough for harder classification tasks. Thereby, *explicit* simultaneous maximization of intra-class similarity and inter-class discrimination is critical. In contrast to some loss functions in literature (Liu et al. 2016; Wen et al. 2016), Copernican loss *explicitly* optimizes for both objectives (invariance and discrimination) between *all* samples in a mini-batch. Although, constrastive embedding (Hadsell, Chopra, and LeCun 2006) and Triplet loss (Szegedy et al. 2015) both advocate simultaneous optimization of discrimination and invariance, they search for useful gradients through clever and expensive sample mining. One way to mitigate the need for mining is to discriminate all samples belonging to different classes away from each other. Copernican loss does this by moving the samples away from the global batch mean, which efficiently provides discriminative gradients without sample mining.

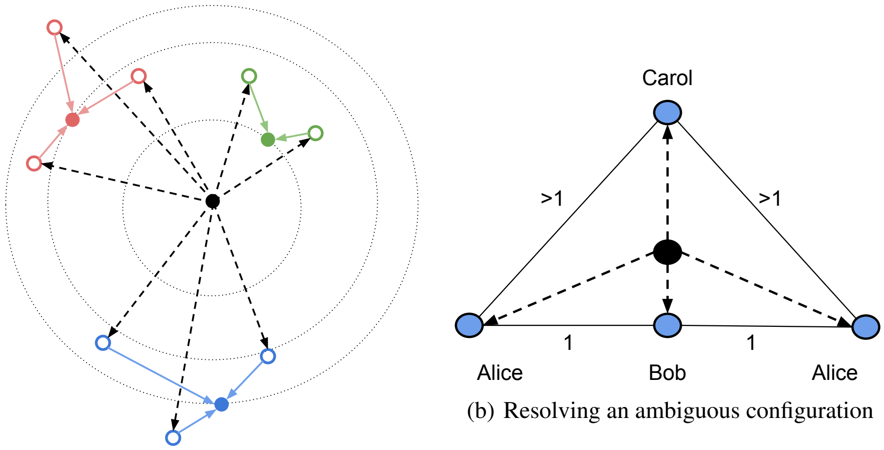
The Need for a Cosine Embedding. Classification in machine learning, at its fundamental level, is typically conducted using the inner-product. The inner-product between the weight vector w and the sample x is a product of their norms and the cosine of the angle between the two vectors. The ideal classification vector w would provide a high inner-product with a sample from the correct class and be low for ones from impostor classes. Thus we would want $w^T x_{cor} > w^T x_{imp} \Rightarrow ||x_{cor}||_2 \cos(\theta_{cor}) > ||x_{imp}||_2 \cos(\theta_{imp})$. The classification decision therefore, ultimately rests on the norm of the sample x and the cosine angle between the weight vector. In this light, there are two ways of increasing discrimination. 1) Increase the norm of the correct class, and 2) decrease the cosine angle between the weight vector and the sample. For binary class problems, increasing the norm of one class over another might be feasible, however for multi-class problems this approach would not be effective. Maximizing the norm of samples from one particular class over all others, would hinder correct classification of other classes. Thereby, one approach to increase discrimination that can be applied to multi-class problems is to maximize the angle (or equivalently the cosine distance) between classes. This reasoning also applies to the Softmax loss function which is perhaps the most commonly used loss in supervised deep learning and is also the baseline in our study. Indeed, minimizing the intra-class cosine distance while simultaneously maximizing the inter-class cosine distance seems to be a reasonable goal. This is also exactly for our proposed Copernican loss optimizes for.

Limitations on an l_2 embedding. There have been multiple loss functions proposed that learn an l_2 embedding such as the Center loss (Wen et al. 2016), Triplet loss (Schroff, Kalenichenko, and Philbin 2015) and Multi-batch (Tadmor et al. 2016). All of these losses explicitly minimize the l_2 distance between samples from the same class. Concretely, for sample features x_1 and x_2 from the same class, $||x_1 - x_2||_2^2 = ||x_1||_2^2 + ||x_2||_2^2 - 2||x_1||_2||x_2||_2 \cos(\theta)$, where θ is the angle between the two samples. Minimizing this quantity requires 1) minimizing the difference between norm of the features x_1 and x_2 and 2) minimizing the cosine distance between the two. There are two cons of this approach.

1. During testing and extraction of a similarity score (*e.g.* for open set face or object feature extraction), only the cosine distance is taken into account. This implies that explicitly constraining the norm is inefficient for the loss function layer from the perspective of matching since the model needs to *learn* sub-tasks (*i.e.* constraining the norm) that it does not require during testing. Successful approaches such as batch normalization (Ioffe and Szegedy 2015) on the other hand do not require the *model or the weights* themselves to perform normalization, they perform it explicitly through the normalization operation. This allows the model complexity to be used to focus on the angles between the samples instead.

2. More importantly, for tasks with a large number of samples per class (such as typical object recognition), trying to

[2] In this paper, we jointly refer to the last fully connected layer of a deep network, along with the cross-entropy loss followed by a softmax layer as the Softmax loss.



(a) 'Top' Copernican view of a mini-batch

(b) Resolving an ambiguous configuration

Figure 1: The figures illustrate a sample mini-batch. Red, blue and green denote different classes. Humoring our Copernican analogy, solid colored dots depict the class centers ('planets'), the hollow colored dots represent samples ('moons') and the black dot denotes the global mini-batch center ('sun'). Copernican loss minimizes the sum of planet loss and sun loss. Planet loss minimizes intra-class variation and pushes the samples (hollow colored dots) towards the class centers (solid colored dots) under the angular or cosine metric. Sun loss maximizes inter-class variation by moving all samples away from the global batch center (solid black dot). Dotted black lines denote the gradient direction for the sun loss and colored solid thin lines denote the gradient directions from planet loss. (a) 'Top' view of the mini-batch where no origin is observed. The dotted circles are *not* the unit norm hypersphere and are shown to simply motivate the Copernican analogy. (b) An example ambiguous configuration as presented in (Tadmor et al. 2016). Copernican loss can provide useful gradients without needing 1) hard sample mining as in (Schroff, Kalenichenko, and Philbin 2015) and using only 2) m gradient computations compared to $m^2 - m$ for (Tadmor et al. 2016).

have the large set of samples per class all converge to the same point in the euclidean sense proves to be a difficult challenge for a deep network. As we find in our experiments with Center loss (Wen et al. 2016)[3], it is much easier to simply have the angle of the sample features converge as opposed to convergence in the l_2 sense. With this approach, the network does not need to constrain norms, and model complexity is better utilized in creating a larger angular margin resulting in better performance during testing. Perhaps this is the reason that none of the previous works on learning l_2 embeddings (Wen et al. 2016; Schroff, Kalenichenko, and Philbin 2015; Tadmor et al. 2016; Hadsell, Chopra, and LeCun 2006) report results on object recognition datasets with a large number of samples per class. Examples include datasets such as CIFAR10 and CIFAR100 which have fewer classes (10 and 100 respectively) but many more samples per class (6000 and 600 images per class respectively). The focus of those studies is mainly face recognition which is characterized by a large number of classes (*e.g.* above 10,000 for CASIA-WebFace) with relatively few samples per class (average of about 50 samples per class for CASIA-WebFace).

Copernican loss

For a batch of size m, Copernican loss, denoted by L_C, is defined as the sum of three loses as follows.

$$L_C = L_{Soft} + \lambda \left(\underbrace{\frac{1}{m}\sum_i^m (1 - \cos(x_i, p_{y_i}))}_{L_P} \right) \quad (1)$$

$$+ \left(\underbrace{\frac{1}{m}\sum_i^m \max(0, \cos(x_i, s) - \beta)}_{L_S} \right) \quad (2)$$

Here, L_P is the Planet loss minimizing intra-class cosine variation and L_S is the Sun loss, maximizing inter-class cosine variation and L_{Soft} is the Softmax loss. β is the margin for Sun loss, s is the global center for the particular batch (or the 'sun') and p's are the class centers (or the 'planets') [4]. In the ideal case, s and p's would represent the class centers and the global center of the entire dataset. However, computing these quantities over the entire dataset would be very expensive, especially for large-scale applications. To get around this problem, in the case of the global center (sun s) of the entire dataset , we approximate it with the global center of each

[3] In our experiments with Center loss, which was shown to perform well in face recognition tasks (Wen et al. 2016), we found that it consistently performs worse than Copernican loss (and for $\lambda = 0.1$ worse than vanilla Softmax loss) on both CIFAR10 and CIFAR100 a fact which supports our hypothesis.

[4] To humor our Copernican analogy, the samples themselves could be considered to be the 'moons' of their corresponding planets (or class centers).

mini-batch. Therefore for every batch, $s = \frac{1}{m}\sum_i^m x_i$. Computation of the class centers (planets p) also face a similar issue due to scale. However, class centers cannot be effectively estimated using a single batch due to high variance, especially in the early stages of training. One way to obtain a better estimate is to maintain a class center and update it with every batch in the following sense,

$$p_{y_i}^{j+1} = p_{y_i}^{j} + \alpha \frac{1}{n} x_i$$

Here, we assume there are n instances of class i in the mini-batch. α is the update weight factor and is usually set to a small value (say 0.05). Compared to direct computation of class centers over the mini-batch set, this update provides more robustness to sample perturbation while simultaneously addressing the problem of scalability in estimating the centers. Copernican loss is simple to implement and unlike Triplet loss (Schroff, Kalenichenko, and Philbin 2015), it does not require hard sampling mining which would've increased the computation complexity of the overall loss. Further, computation of discriminative gradients only requries m gradient computations compared to $m(m-1)$ for Multi-batch (Tadmor et al. 2016).

Resolving an ambiguous configuration. We discuss the ambiguous configuration presented in (Tadmor et al. 2016) in Fig. 1(b). Here, four samples are mapped into the feature space during early training when classes are mixed. The algorithm needs to pick pairs of samples from different classes to take a gradient step towards better discrimination. Picking Alice-Carol pair would not help the configuration as they have sufficient separation (> 1) and neither will Bob-Alice pairs. As (Tadmor et al. 2016) points out, only Carol-Bob pairs will provide useful updates by moving Bob away. Center loss (Wen et al. 2016) would worsen things since it will push the two Alice's closer to Bob. Triplet loss (Schroff, Kalenichenko, and Philbin 2015) would need to sample Carol-Bob which is expensive to determine hard-sample mining and unlikely if no sample mining is performed. Multi-batch (Tadmor et al. 2016) solves this problem by considering *all* pairs in the mini-batch. Our proposed Copernican loss is guaranteed to provide discriminative gradients for all samples without the need for 1) sample mining nor 2) $m(m-2)$ gradient computations (we need only m), since it moves samples away from the Sun center (black dot in Fig. 1(b)). To bring Alice pair closer, Planet loss provides a gradient which will be effective since Bob has moved out of the way due to the Sun loss.

Optimization

The gradients of Copernican loss are straight forward. Since the sun center and planet centers are updated during forward pass, only gradients with respect to the input x_i is required to be derived which are are follows.

$$\frac{\partial L_P}{\partial x_i} = -\frac{1}{m}\sum_i^m \frac{1}{||x_i||_2}\left(\frac{p_{y_i}}{||p_{y_i}||_2} - \cos(x_i, p_{y_i})\frac{x_i}{||x_i||_2}\right) \tag{3}$$

$$\frac{\partial L_S}{\partial x_i} = \begin{cases} \frac{1}{m}\sum_i^m \frac{1}{||x_i||_2}\left(\frac{s}{||s||_2} - \cos(x_i, s)\frac{x_i}{||x_i||_2}\right) \\ 0 \end{cases} \tag{4}$$

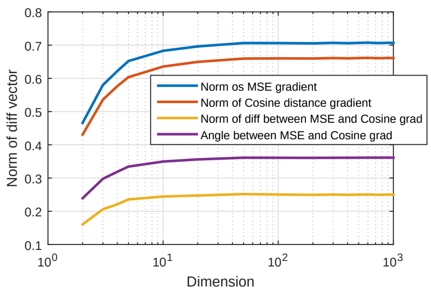


Figure 2: The norm of the difference vector between the Cosine distance gradient and the MSE distance gradient and the angle between them (in radians) remains fairly constant as dimensionality increases. Experiment was performed on random vectors for features and targets (planets).

where $\frac{\partial L_S}{\partial x_i}$ becomes 0 only if $\cos(x_i, s) \leq \beta$. It is interesting to note that the gradient direction of the cosine similarity is different from the MSE gradient by a scaled version of the sample feature. This scaling factor is the cosine angle between the sample feature and the target vector. Fig. 2 shows the norm of the difference between the MSE gradient and the cosine similarity gradient as dimensionality increases along with the angle between the two [5]. For all dimensions above 10, the angle between the cosine similarity gradient and MSE gradient is about 20°. It is interesting to note that a gradient direction differing by 20° can result in significantly improved performance as we find in our experiments. As the sample features converged to the planets or class centers, the MSE gradient direction converges to the cosine similarity gradient direction, since $\cos(x_i, p_{y_i}) \rightarrow 1$. This fact coupled with superior performance of the cosine similarity leads us to hypothesize that during the initial stages of training, when the features are not informative of the class structure, MSE or l_2 embedding gradients are more noisy. Cosine embeddings are able to obtain better gradients by ignoring this noise since it only looks at angular margins which is not affected by the norm of the vectors. We observe this in our experiments.

Experimental Validation

We present results on general object recognition benchmarks of CIFAR10 and CIFAR100 primarily to study and compare the behavior of the Copernican loss with vanilla Softmax and Center loss (Wen et al. 2016). We also showcase the efficacy of our method on large scale face recognition tasks with *no alignment* for a harder challenge. We implemented Copernican loss and Center loss following (Wen et al. 2016) in Torch. The class center update learning rate α was set to 0.05 for all experiments.

General model and implementation details. Our benchmarking model for all experiments for consistency is a Resnet following (He et al. 2016) based off the Torch implementa-

[5] Two random normalized vectors were used for the gradient computations. For each dimensionality, 1000 such random pairs were averaged over.

tion[6]. During development of our method, similar performance trends were observed for other architectures as well. For CIFAR experiments, the models followed the corresponding CIFAR architecture from ResNet Torch codebase.

For our face recognition experiments our network had 4 stages with 2 blocks each, with number of filters as $\{64, 128, 256, 512\}$. The network ended with a global pooling stage (8 by 8 for CIFAR and 7 by 7 for face recognition) followed by a fully connected layer. The feature dimension (input to fully connected layer) for CIFAR was 1024 and for face recognition was 512. The bottleneck layer type is a pure Resnet connection. The CIFAR models were trained on a single GPU whereas the face recognition models were trained on 2 GPUs, all with a mini-batch size of 128, a weight decay of 0.0005 and momentum of 0.9. Learning rate was set to 0.1 and reduced by a factor of 10 at the 50 % and at the 75 % mark.

Experiments with CIFAR 10 and CIFAR 100

Our experiments with CIFAR10 and 100 focus on exploring the effect of β and λ on performance. We also focus on comparisons with the baselines Softmax and Center loss. A comparison with a version of Copernican loss learning an l_2 embedding (instead of cosine) is also made. We train on default 50K training set and test on the 10K test set. The only data augmentation used was zero-padding the original image 32×32 image to 40×40 followed by a random crop of size 32×32 and flipping with a probability of 0.5. Fig. 3 and Table 1 present the results of these experiments. The final error rate reported is the average of the last 50 epochs (with very small variance).

Limitations of an l_2 embedding. To highlight the significance of a cosine embedding, we also benchmark against the l_2 version of Copernican loss (denoted as l_2 Copernican (Ours) in Table 1). The l_2 version minimizes the MSE instead of the cosine distance between the sample features x_i and the planet centers p_{y_i} and maximizes the same between the features x_i and the sun center s. From Table 1, we find that both Center loss and l_2 Copernican loss perform worse than the cosine embedding. Interestingly, Table 1 shows that Center loss performs worse than vanilla Softmax for $\lambda = 0.1$. This provides more evidence towards the limitations of an l_2 embedding. Indeed, trying to converge all 6000 samples per class for CIFAR 10 and 600 samples for CIFAR 100 to the same point in euclidean space (the class center) in the l_2 sense proves to be a difficult task which hampers overall network performance. A cosine embedding consistently provides significant improvement with the top performance being 5.04% on CIFAR 10 for $\lambda = 1, \beta = -1$ and and 23.68% on CIFAR 100 for $\lambda = 5, \beta = -0.5$, thus showcasing its advantages.

Effect of margin β. We explore the effect of margin β on Copernican loss. We fix $\lambda = 0.1$ for Copernican loss and Center loss. Fig. 3(a) and Table 1 showcase the results of these experiments. Note that a lower β tries to enforce more discrimination. We find that a margin $\beta = -1$ and $\beta = -0.5$ perform the best on CIFAR 10 and 100 respectively. As a general observation, we found that if the number of classes are high, a lower margin performs better.

Table 1: Top 1 Test Error Rate (T.E.R) % on CIFAR 10+ and CIFAR 100+ (+ denotes standard data augmentation as in (Huang et al. 2016; He et al. 2016)). Numbers in brackets denote either (λ) or (λ, β). $l2$-Copernican denotes the version which operates using the Euclidean distance (baseline) as opposed to Cosine similarity (proposed).

Method	CIFAR 10+	CIFAR 100+
MaxOut (Goodfellow et al. 2013)	9.38	38.57
Drop Connect (Wan et al. 2013)	9.32	
NiN (Lin, Chen, and Yan 2013)	8.81	35.68
FitNet (Romero et al. 2014)	8.39	35.04
DSN (Lee et al. 2015)	7.97	34.57
All-Conv(Springenberg et al. 2014)	7.25	33.71
Recurrent-Conv (Liang and Hu 2015)	7.09	31.75
ResNet (He et al. 2016)	6.43	
Generalized Pooling (Lee, Gallagher, and Tu 2016)	6.05	32.37
Large-Margin Softmax (Liu et al. 2016)	5.92	29.53
SoftMax	6.42	24.56
Center loss (0.1) (Wen et al. 2016)	6.67	24.89
Center loss (1) (Wen et al. 2016)	5.92	24.14
l_2 Copernican (Ours) (0.1)	5.67	24.75
l_2 Copernican (Ours) (1)	5.74	24.55
Copernican (Ours) $(1, 0.5)$	5.62	24.14
Copernican (Ours) $(1, -0.1)$	5.38	24.56
Copernican (Ours) $(1, -1)$	**5.04**	23.77
Copernican (Ours) $(5, -0.1)$	5.16	23.68
Copernican (Ours) $(5, -0.5)$	5.18	**22.81**

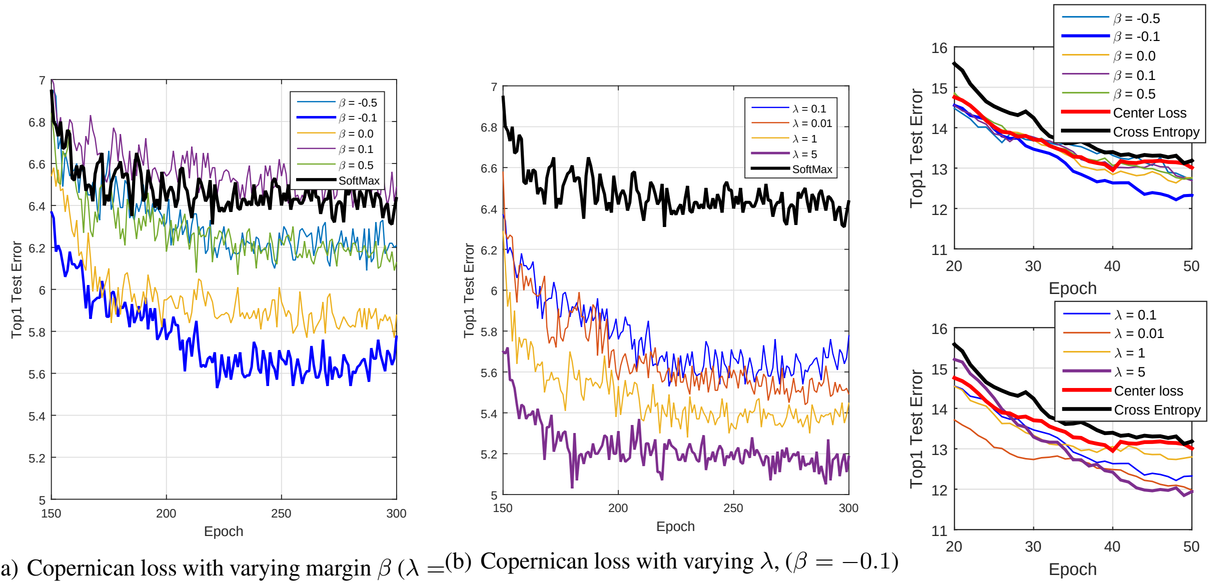
Effect of loss weight λ. We explore the effect of λ on Copernican loss with $\beta = -0.1$. Fig. 3(b) and Table 1 showcase the results of these experiments. For CIFAR, we find that a higher λ (≥ 1) improves performance for higher margin β. Setting λ to be high for very high β led to a decrease in performance. Nonetheless, in all experiments with λ, the entire range significantly outperformed Softmax loss. As a rule of thumb, keeping λ low (around 0.1) for large scale experiments (as we find in face recognition) with a low margin works well, for smaller scale experiments (such as CIFAR), a higher λ offers more performance.

Experiments in Large-Scale Face Recognition

We benchmark Copernican loss on two large scale face recognition datasets namely LFW (Huang et al. 2007) and IJB-A Janus (Klare et al. 2015). Face recognition is a challenging task in general. This is because most benchmarks have a large number of classes (above 10,000 classes), with each class having a few number of samples (average 50 samples per subject for CASIA-WebFace, compared to say 6000 for CIFAR10). Such a task requires losses which can focus on the interplay between the few samples from different classes and is a good benchmark for comparison.

Towards Alignment-free Face Recognition. Most systems in face recognition utilize a landmark scheme for alignment (Schroff, Kalenichenko, and Philbin 2015; Taigman et al. 2014; Tadmor et al. 2016; Wen et al. 2016). Indeed facial alignment boosts the performance of a face recognition

[6] https://github.com/facebook/fb.resnet.torch



(a) Copernican loss with varying margin β ($\lambda =$ 0.1)

(b) Copernican loss with varying λ, ($\beta = -0.1$)

(c) Early training

Figure 3: Top 1 Test Error on CIFAR 10 with varying (a) margin β within $\{-0.5, -0.1, 0.0, 0.1, 0.5\}$ with $\lambda = 0.1$ and (b) λ within $\{0.01, 0.1, 1, 5\}$ with margin $\beta = -0.1$. (c) Top 1 Test Error for early epochs during training. Test error was a running average over 10 epochs to reduce variance for clarity. Top and bottom show performance analogous to the settings of (a) and (b) respectively. We see that test error is low for most settings of Copernican loss even during early training suggesting that cosine embeddings provide more informative gradients early on as discussed in Sec. . This leads to faster convergence.

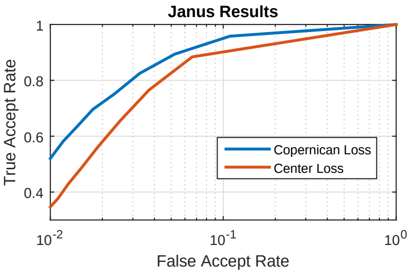


Figure 4: ROC curve on the IJB-A Janus protocol. Copernican loss outperforms Center loss. Performance reported while training and testing on unaligned data.

engine by a significant amount which has led to saturation of benchmarks such as LFW. Although alignment is useful in its own right and also for real-world application, moving towards alignment-free pipelines offers a huge room for improvement and development and demonstration of more powerful methods. With this thought in mind, we restrict ourselves to the extremely challenging protocol of being alignment-free and utilizing minimal pre-processing of training and testing data as we describe below.

Training and pre-processing. As mentioned, we utilize no alignment in either training or testing and therefore do not require any landmarking. All images we train and test on are 250 × 250 square crops centered on the face. For cropping testing data, we utilized the bounding box provided, computed the center of the box and cropped a 250 by 250 patch around it. We train on the CASIA-WebFace (Yi et al. 2014) database containing about 10,575 subjects and 494,414 images which were originally 250 × 250. Therefore the images were not processed any further and were only horizontally flipped for data augmentation. Due to the absence of alignment, there were many images with considerable in-plane rotation and/or slight scale variations which made the task much harder than the traditional aligned face recognition pipeline. This helps to showcase the efficacy of Copernican loss. Lastly, 127 was subtracted from each pixel value and the result was then divided by 127. We train models purely on CASIA-WebFace and test them on LFW and IJB-A Janus without any modification or finetuning.

Testing Benchmark: LFW. The LFW (Huang et al. 2007) database contains about 13,000 images collected from the web of 1680 distinct subjects. The database contains considerable degradations such as illumination, color, and age along with significant pose variation. We follow the unrestricted outside training data protocol for LFW exactly.

Testing Benchmark: IJB-A Janus. IJB-A (Klare et al. 2015) is a new difficult and publicly available challenge. IJB-A consists of 500 subjects under extreme conditions regarding pose, expression and illuminations with a total of 25,813 images. The IJB-A evaluation protocol mainly consists of face verification (1:1) and face identification (1:N). The interesting thing about this dataset is that each subject is described by a template containing a set of images or frames extracted from videos. We focus on the 1:1 tem-

Table 2: LFW results. Numbers in brackets denote either (λ) or (λ, β)

Method	Training Data	Alignment	Accuracy (%)
FaceNet (Schroff, Kalenichenko, and Philbin 2015)	200M private	Yes	99.65
Center loss (Wen et al. 2016)	WebFace+0.2M private	Yes	99.28
DeepFace (Taigman et al. 2014)	4.4M private	Yes	97.35
DeepID (Sun, Wang, and Tang 2014)	88K	Yes	97.45
Masi *et. al.* (Masi et al. 2016b)	WebFace+2.4M synth	Yes	98.06
Multi-batch(Tadmor et al. 2016)	2.6M	Yes	98.20
Wang *et. al.* (Wang, Otto, and Jain 2015)	WebFace	Yes	97.52
Yi *et. al.* (Yi et al. 2014)	WebFace	Yes	97.73
DCNN (Chen, Patel, and Chellappa 2016)	WebFace	Yes	97.45
Center loss (Wen et al. 2016) (0.1)	WebFace	No	98.13
Copernican loss $(0.1, 0.5)$ (Ours)	WebFace	**No**	**98.27**

Table 3: IJB-A Janus results. Numbers in brackets denote either (λ) or (λ, β)

Method	Training Data	Alignment	VR (%) @ 0.1 FAR)
Wang *et. al.* (Wang, Otto, and Jain 2015)	WebFace	Yes	89.5
DCNN (Chen, Patel, and Chellappa 2016)	WebFace	Yes	80.0
DCNN$_{ft}$ (Chen, Patel, and Chellappa 2016)	WebFace	Yes	88.3
DCNN$_{ft+m}$ (Chen, Patel, and Chellappa 2016)	WebFace	Yes	94.7
DCNN$_{ft+m+c}$ (Chen, Patel, and Chellappa 2016)	WebFace	Yes	96.1
DCNN$_{fusion}$ (Chen, Patel, and Chellappa 2016)	WebFace	Yes	96.7
PAM (Masi et al. 2016a)	WebFace	Yes	93.0
Chen *et. al.* (Chen et al. 2015)	WebFace	Yes	96.8
Center loss (Wen et al. 2016) (0.1)	WebFace	No	90.0
Copernican loss (0.1 0.5) (Ours)	WebFace	**No**	**95.0**

plate verification protocol. There are 10 splits with about 12,000 pair-wise template matches each resulting in a total of 117,420 template matches. To extract a score for the template pair T_i, T_j, we utilize the following formula. $S(T_i, T_j) = \sum_{\gamma=1}^{8} \frac{\sum_{t_a \in T_i, t_b \in T_j} s(t_a, t_b) \exp \gamma s(t_a, t_b)}{\sum_{t_a \in T_i, t_b \in T_j} \exp \gamma s(t_a, t_b)}$. Here $s(t_a.t_b)$ denotes the cosine similarity score between images t_a, t_b.

Results. The results of these experiments are shown in Table. 2 (LFW) and 3 (IJB-A). Fig. 4 shows the ROC curves for IJB-A Janus. From Table. 2, we find that despite not using alignment we achieve 98.27% on LFW which is higher accuracy than many studies that train on far more data *e.g.* (Tadmor et al. 2016; Wang, Otto, and Jain 2015; Taigman et al. 2014). We outperform Center loss trained without alignment on CASIA-WebFace . From Fig. 4 and Table. 3, we find that Copernican loss (95.0%) outperforms the baseline Center loss (90.0%) on IJB-A Janus by a significant amount. Further, our system trained with Copernican loss approaches the high performance achieved by other methods while training on the same dataset (CASIA-WebFace) while using no landmarks.

References

[Chen et al. 2015] Chen, J.-C.; Ranjan, R.; Kumar, A.; Chen, C.-H.; Patel, V. M.; and Chellappa, R. 2015. 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*, 118–126.

[Chen, Patel, and Chellappa 2016] Chen, J.-C.; Patel, V. M.; and Chellappa, R. 2016. Unconstrained face verification using deep cnn features. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, 1–9. IEEE.

[Goodfellow et al. 2013] Goodfellow, I. J.; Warde-Farley, D.; Mirza, M.; Courville, A.; and Bengio, Y. 2013. Maxout networks. *arXiv preprint arXiv:1302.4389*.

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

[He et al. 2015] He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

[He et al. 2016] He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

[Huang et al. 2007] Huang, G. B.; Ramesh, M.; Berg, T.; and Learned-Miller, E. 2007. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst.

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

[Klare et al. 2015] Klare, B. F.; Klein, B.; Taborsky, E.; Blanton, A.; Cheney, J.; Allen, K.; Grother, P.; Mah, A.; and Jain, A. K. 2015. 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*, 1931–1939.

[Lee et al. 2015] Lee, C.-Y.; Xie, S.; Gallagher, P.; Zhang, Z.; and Tu, Z. 2015. Deeply-supervised nets. In *Artificial Intelligence and Statistics*, 562–570.

[Lee, Gallagher, and Tu 2016] Lee, C.-Y.; Gallagher, P. W.; and Tu, Z. 2016. Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 464–472.

[Liang and Hu 2015] Liang, M., and Hu, X. 2015. Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3367–3375.

[Lin, Chen, and Yan 2013] Lin, M.; Chen, Q.; and Yan, S. 2013. Network in network. *arXiv preprint arXiv:1312.4400*.

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

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

[Masi et al. 2016a] Masi, I.; Rawls, S.; Medioni, G.; and Natarajan, P. 2016a. Pose-aware face recognition in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4838–4846.

[Masi et al. 2016b] Masi, I.; Tran, A. T.; Hassner, T.; Leksut, J. T.; and Medioni, G. 2016b. Do we really need to collect millions of faces for effective face recognition? In *European Conference on Computer Vision*, 579–596. Springer.

[Nair and Hinton 2010] Nair, V., and Hinton, G. E. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 807–814.

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

[Romero et al. 2014] Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.

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

[Schroff, Kalenichenko, and Philbin 2015] Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 815–823.

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

[Springenberg et al. 2014] Springenberg, J. T.; Dosovitskiy, A.; Brox, T.; and Riedmiller, M. 2014. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*.

[Srivastava et al. 2014] Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15(1):1929–1958.

[Sun, Wang, and Tang 2014] Sun, Y.; Wang, X.; and Tang, X. 2014. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1891–1898.

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

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

[Taigman et al. 2014] Taigman, Y.; Yang, M.; Ranzato, M.; and Wolf, L. 2014. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1701–1708.

[Wan et al. 2013] Wan, L.; Zeiler, M.; Zhang, S.; Cun, Y. L.; and Fergus, R. 2013. Regularization of neural networks using dropconnect. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 1058–1066.

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

[Wang, Otto, and Jain 2015] Wang, D.; Otto, C.; and Jain, A. K. 2015. Face search at scale: 80 million gallery. *CoRR* abs/1507.07242.

[Wen et al. 2016] Wen, Y.; Zhang, K.; Li, Z.; and Qiao, Y. 2016. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, 499–515. Springer.

[Yi et al. 2014] Yi, D.; Lei, Z.; Liao, S.; and Li, S. Z. 2014. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*.