

---

# Rethinking Feature Discrimination and Polymerization for Large-scale Recognition

---

**Yu Liu\***, **Hongyang Li\***, **Xiaogang Wang**

The Chinese University of Hong Kong  
 {yuliu, yangli, xgwang}@ee.cuhk.edu.hk

## Abstract

Feature matters. How to train a deep network to acquire discriminative features across categories and polymerized features within classes has always been at the core of many computer vision tasks, specially for large-scale recognition systems where test identities are unseen during training and the number of classes could be at million scale. In this paper, we address this problem based on the simple intuition that the cosine distance of features in high-dimensional space should be close enough within one class and far away across categories. To this end, we proposed the congenerous cosine (COCO) algorithm to simultaneously optimize the cosine similarity among data. It inherits the softmax property to make inter-class features discriminative as well as shares the idea of class centroid in metric learning. Unlike previous work where the center is a temporal, statistical variable within one mini-batch during training, the formulated centroid is responsible for clustering inner-class features to enforce them polymerized around the network truncus. COCO is bundled with discriminative training and learned end-to-end with stable convergence. Experiments on five benchmarks have been extensively conducted to verify the effectiveness of our approach on both small-scale classification task and large-scale human recognition problem.

## 1 Introduction

Deep neural networks [14, 8, 9] have dramatically advanced the computer vision community in recent years, with high performance boost in tremendous tasks, such as image classification [8, 34, 21], object detection [4, 32, 20, 30, 27], object tracking [2], saliency detection [19], etc. The essence behind the success of deep learning resides in both the superior expression power of non-linear complexity in high-dimension space [9] and the large-scale datasets [3, 7] where the deep networks could, in full extent, learn complicated patterns and representative features. The basic question still halts the core of many vision tasks: *how can we obtain discriminative features across categories and polymerized features within one class?*

The most fundamental task in computer vision is image classification [34, 14, 8]: given  $N$  categories in the training set, we train a deep model that could best discriminate features among categories in high-dimensional space. The test set also consists of the same  $N$  classes (identities). However, there are two challenges when we apply the classification task in real-world applications. The first is the number of categories could be exponentially increased (*e.g.*, MegaFace [12] contains one million test identities). This requires the features should be *discriminative* to have large margin across classes. The second is the category mismatch between training and test set. Often we train on a set of  $N$  known classes and evaluate the algorithm on  $M$  new, different identities. This hastens the features should be *polymerized* to share small distances within classes.

---

\*The first two authors contribute equally.

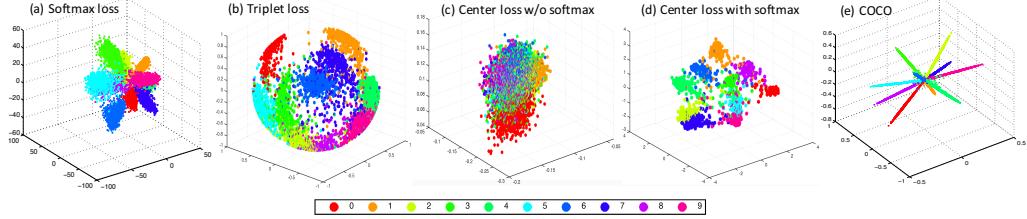


Figure 1: Feature visualization under different loss strategies, trained on MNIST [15] with 10 classes. (a) Softmax loss; (b) triplet loss; (c) center loss without softmax; (d) center loss; (e) our proposed COCO. The feature dimension is three for better illustration. The triplet loss [33] has a constraint on feature embedding ( $\|f\|_2 = 1$ ) and thus the learned features are scattered on a hypersphere.

In this paper, we optimize the deeply learned features to be both discriminative and polymerized, given the large-scale constraint: ultra category number and open test set. Faced with the aforementioned challenges, the community extends the classification problem into two sub tasks: data verification and identification. Since our main focus lies in the field of human-related tasks, we use the term face with data exchangeably thereafter. Face verification [38, 31] aims at differentiating whether a pair of instances belong to the same identity; whilst face identification [25, 35] strives for predicting the class of an identity given the test gallery [12]. Person recognition [28, 44] also belongs to the verification case, only with more non-frontal and unstrained face (head) settings<sup>2</sup>.

For the feature discrimination concern, the softmax loss and its variants [26, 31, 25] are proposed to provide a hyperplane with large margin in high-dimensional space and thus effectively distinguish features across classes. However, it ignores the intra-class similarity, making the sample distance within one class distant also and thus amply filling in the feature space (Fig. 1(a)); for the feature polymerization concern, one can resort to distance metric learning [43, 41]. Previous work [36] usually introduced neural networks, using contrastive loss [35] or triplet loss [39, 24, 33], to learn the features, followed by a simple metric such as Euclidean or cosine distance to verify the identity (Fig. 1(b)). Such methods are effective to decrease the intra-class distance in some extent and yet bears an internal drawback due to the formulation: the selection of positive and negative samples are relative within each batch and thus could occur training instability, especially the number of classes and data scale increase (see Fig. 3(b)).

An alternative is to jointly combine the discriminative feature learning in a softmax spirit and the polymerized feature learning in a metric learning manner. The center loss [42] and its variant [45] are two typical approaches. They linearly combine two terms as the total loss: the first being softmax and the second a class center supervision  $\|\mathbf{x}_i - \mathbf{c}_{l_i}\|$ . The hybrid solution can effectively enlarge the inter-class distance as well as the inner-class similarity; nevertheless, the concept of class center is updated in a statistical manner without consulting to the network parameter update during training. Moreover, the center loss term has to be bundled with an ad-hoc softmax (thus requires more model parameters); if not, the center barely changes according to its formulation and thus the loss could be zero (*c.f.*, Fig. 1(c)-(d)).

To this end, we propose the congenerous cosine (COCO) algorithm to provide a new perspective on jointly considering the feature discrimination and polymerization. The intuition is that we directly optimize and compare the cosine distance (similarity) between features: the intra-class distance should be as minimal as possible whereas the inter-class variation should be magnified to the most extent (see Fig. 1(e)). COCO inherits the softmax property to make features discriminative in the high-dimensional space as well as keeps the idea of class centroid. Note that unlike the statistical role of  $\mathbf{c}_{l_i}$  in [42], the update of our centroid is performed simultaneously during the network training; and there is no additional loss term in the formulation. By virtue of class centroid bundled with discriminative training, COCO is learned end-to-end with stable convergence. Experiments on two small-scale image classification datasets first verify the feasibility of our proposed approach and three

<sup>2</sup> The term recognition has a broad sense: detection, verification, classification, *etc.* Person recognition is face verification with more unstrained samples. For not making any confusion, we follow this terminology as did in the community. In this paper, we refer to face verification, identification and person recognition, in a general sense, as *human recognition*.

large-scale human recognition benchmarks further demonstrate the effectiveness of applying COCO in an ultra class number and open test set environment. The source code is publicly available<sup>3</sup>.

## 2 Related Work

The related work of different loss methods to optimize features is compared and discussed throughout the paper. In this section, we mainly focus on the face and person recognition applications.

**Face verification and identification** have been extensively investigated [12, 31, 25, 36, 35, 42] recently due to the dramatic demand in real-life applications. Schroff *et al.* [33] directly learned a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity. DeepID2 [35] employed a contrastive loss with both the identification and verification supervision. It increased the dimension of hidden representations and added supervision to early convolutional layers. Tadmor *et al.* [36] introduced a multibatch method to generate invariant face signatures through training pairs. The training tended to be faster to convergence by virtue of smaller variance of the estimator. SphereFace [25] formulated an angular softmax loss that enabled CNN to learn discriminative features by imposing constraints on a hypersphere manifold.

**Person recognition** in photo albums [1, 29, 28, 22, 44, 18] aims at recognizing the identity of people in daily life photos, where such scenarios can be complex with cluttered background. [1] first address the problem by proposing a Markov random field framework to combine all contextual cues to recognize the identity of persons. Recently, Zhang *et al.* [44] introduce a large-scale dataset called PIPA for this task. The test set on PIPA is split into two subsets, namely `test_0` and `test_1` with roughly the same number of instances. In [28], a detailed analysis of different cues is explicitly investigated and three additional test splits are proposed for evaluation. [22] embed scene and relation contexts in LSTM and formulate person recognition as a sequence prediction task. Note that previous work [44, 28, 18, 22] use the training set only for extracting features and a follow-up classifier (SVM or neural network) is *trained* on `test_0`. The recognition system is evaluated on the `test_1` set. We argue that such a practice is infeasible and ad hoc in realistic application since the second training on `test_0` is auxiliary and needs re-training if new samples are added.

## 3 Optimizing Congenerous Cosine Distance

### 3.1 COCO Formulation and Optimization

Let  $\mathbf{f}^{(i)} \in \mathbb{R}^D$  denote the feature vector of the  $i$ -th sample, where  $D$  is the feature dimension. We first introduce the cosine similarity of two features as:

$$\mathcal{C}(\mathbf{f}^{(i)}, \mathbf{f}^{(j)}) = \mathbf{f}^{(i)T} \cdot \mathbf{f}^{(j)} / \|\mathbf{f}^{(i)}\| \|\mathbf{f}^{(j)}\|. \quad (1)$$

The cosine similarity metric quantifies how close two samples are in the feature space. A natural intuition to a desirable loss is to increase the similarity of samples within a category and enlarge the centroid distance of samples across classes. Let  $l_i, l_j \in \{1, \dots, K\}$  be the labels of sample  $i, j$ , where  $K$  is the total number of categories, we have the following loss to *maximize*:

$$\mathcal{L}^{naive} = \sum_{i,j \in \mathcal{B}} \frac{\delta(l_i, l_j) \mathcal{C}(\mathbf{f}^{(i)}, \mathbf{f}^{(j)})}{[1 - \delta(l_i, l_j)] \mathcal{C}(\mathbf{f}^{(i)}, \mathbf{f}^{(j)}) + \epsilon}, \quad (2)$$

where  $\mathcal{B}$  denotes the mini-batch,  $\delta(\cdot, \cdot)$  is an indicator function and  $\epsilon$  is a trivial number for computation stability. The naive design in (2) is reasonable in theory and yet suffers from computational inefficiency. Since the complexity of the loss above is  $\mathcal{O}(C_M^2) = \mathcal{O}(M^2)$ , it increases quadratically as batch size  $M$  goes bigger. Also the network suffers from unstable parameter update and is hard to converge if we directly compute loss from two arbitrary samples from a mini-batch: a similar drawback as does in the triplet loss [39].

As discussed previously, an effective solution to polymerize the inner-class feature distance is to provide a series of truncus in the network: serving as the centroid for each class and thus enforcing features to be learned around these hubs. To this end, we define the *centroid* of class  $k$  as the average

---

<sup>3</sup> [https://github.com/sciencefans/coco\\_loss](https://github.com/sciencefans/coco_loss)

of features over a mini-batch:  $\mathbf{c}_k = \frac{1}{N_k} \sum_{i \in \mathcal{B}} \delta(l_i, k) \mathbf{f}^{(i)} \in \mathbb{R}^D$ , where  $N_k$  is the number of samples that belong to class  $k$  within the batch. Incorporating the spirit of class centroid into Eqn. (2), one can derive the following revised loss to maximize:

$$\mathcal{L}^{revise} = \sum_{i \in \mathcal{B}} \frac{\exp \mathcal{C}(\mathbf{f}^{(i)}, \mathbf{c}_{l_i})}{\sum_{m \neq l_i} \exp \mathcal{C}(\mathbf{f}^{(i)}, \mathbf{c}_m)}, \quad (3)$$

where  $m$  indexes along the category dimension. The direct intuition behind Eqn. (3) is to measure the distance of one sample against other samples by way of a class centroid, instead of a direct pairwise comparison as in Eqn. (2). The numerator ensures sample  $i$  is close enough to its class center  $l_i$  and the denominator enforces a minimal distance against samples in other classes. The exponential operator  $\exp$  is to transfer the cosine similarity to a normalized probability output.

To optimize the aforementioned loss in practice, we first normalize the feature and centroid by  $l_2$  norm and then scale the feature before feeding them into the loss layer:

$$\hat{\mathbf{c}}_k = \frac{\mathbf{c}_k}{\|\mathbf{c}_k\|}, \hat{\mathbf{f}}^{(i)} = \frac{\alpha \mathbf{f}^{(i)}}{\|\mathbf{f}^{(i)}\|}, p_k^{(i)} = \frac{\exp(\hat{\mathbf{c}}_k^T \cdot \hat{\mathbf{f}}^{(i)})}{\sum_m \exp(\hat{\mathbf{c}}_m^T \cdot \hat{\mathbf{f}}^{(i)})}, \quad (4)$$

where  $\alpha$  is the scale factor and  $p_k^{(i)}$  is an alternative expression<sup>4</sup> to the inner term of summation in Eqn. (3). Therefore, the proposed congenerous cosine (COCO) loss are formulated, in a cross-entropy manner to minimize, as follows:

$$\mathcal{L}^{COCO}(\mathbf{f}^{(i)}, \mathbf{c}_k) = - \sum_{i \in \mathcal{B}, k} t_k^{(i)} \log p_k^{(i)} = - \sum_{i \in \mathcal{B}} \log p_{l_i}^{(i)}, \quad (5)$$

where  $k$  indexes along the class dimension in  $\mathbb{R}^K$  and  $t_k^{(i)} \in \{0, 1\}$  is the binary mapping of sample  $i$  based on its label  $l_i$ . The proposed COCO inherits the properties from (2) to (3), that is to increase the discrimination across categories as well as to polymerize the compactness within one class in a cooperative way. Moreover, it reduces the computation complexity compared with the naive version and can be implemented in a cheat way.

Note that both the features and cluster centroids are trained end-to-end. We now derive the gradients of loss  $\mathcal{L}^{COCO}$  w.r.t. the unnormalized input feature  $\mathbf{f}^{(i)}$  and the centroid  $\mathbf{c}_k$ . For brevity, we drop the sample index  $i$  and superscript notation in the loss. Denote  $\nabla_x \mathcal{L}, x \in \{d, j\}$  as the element-wise top gradient of loss w.r.t. the normalized feature, i.e.,  $\partial \mathcal{L} / \partial \hat{f}_x$ , we have the following gradient by applying the chain rule:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial f_j} &= \sum_d \nabla_d \mathcal{L} \cdot \frac{\partial \hat{f}_d}{\partial f_j} = \nabla_j \mathcal{L} \cdot \frac{\partial \hat{f}_j}{\partial f_j} + \sum_{d \neq j} \nabla_d \mathcal{L} \cdot \frac{\partial \hat{f}_d}{\partial f_j}, \\ &= \alpha \left( \frac{\nabla_j \mathcal{L}}{\|\mathbf{f}\|} - \sum_d \nabla_d \mathcal{L} \cdot \frac{f_i f_d}{\|\mathbf{f}\|^3} \right) = \frac{\alpha (\nabla_j \mathcal{L} - (\nabla \mathcal{L}^T \cdot \hat{\mathbf{f}}) \hat{f}_j)}{\|\mathbf{f}\|}, \end{aligned} \quad (6)$$

where  $\nabla \mathcal{L} \in \mathbb{R}^D$  is the vector form of  $\nabla_x \mathcal{L}$ . Considering the specific loss defined in Eqn. (5), the top gradient  $\nabla_x \mathcal{L}$  can be obtained:

$$\nabla_x \mathcal{L} = \sum_k \frac{\partial \mathcal{L}}{\partial z_k} \cdot \frac{\partial z_k}{\partial \hat{f}_x} = \sum_k (p_k - t_k) \cdot \hat{c}_{kx}. \quad (7)$$

The derivation of gradient w.r.t. centroid  $\partial \mathcal{L} / \partial c_{kj}$  can be derived in a similar manner. The features are initialized from pretrain models and the initial value of  $\mathbf{c}_k$  is thereby obtained.

### 3.2 Towards an Optimal Scale Factor $\alpha$

As stated in previous subsection, we first normalize and scale the feature; now we have the following theorem to prove that there exists an optimal value for the factor. The derivations are provided in the supplementary.

---

<sup>4</sup> In practice, it is more feasibly convenient to include  $m = l_i$  in the denominator and the result barely differs.

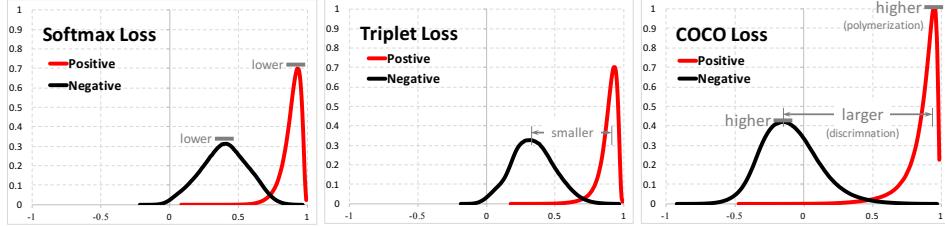


Figure 2: Histogram statistics of cosine distance for positive and negative pairs on MNIST [15]. Feature discrimination and polymerization are more evident using COCO than other losses.

**Theorem 1.** Given the optimization loss  $\mathcal{L}$  has an upper bound  $\epsilon$ , i.e.,  $\mathcal{L} < \epsilon$ ; and the neural network has a class number of  $K$ , the scale factor  $\alpha$  enforced on the input feature has a lower boundary:

$$\alpha > \frac{1}{2} \log \frac{K-1}{\exp \epsilon - 1}. \quad (8)$$

### 3.3 COCO Verification on Image Classification

In this subsection, we provide results and discussions to verify the effectiveness of COCO on small-scale image classification datasets, namely, MNIST [15] and CIFAR-10 [13]. The dataset, network structure and implementation details are provided in the supplementary.

**An optimal scale factor matters.** Take CIFAR-10 as example. When  $\alpha$  is small ( $\alpha = 1$ , error rate 12.4), the lower bound of loss is high, leading to a high loss in easy samples; however, the loss for hard examples still remains at low value. This will make the gradients from easy and hard samples stay at the same level. This is hard for the network to optimize towards hard samples - leading to a higher error rate. When  $\alpha = 10$ , the error rate is 7.22; if we set the factor to the optimal value defined in Eqn. (8), the error rate is 6.25.

**Feature discrimination and polymerization.** Figure 2 shows the histogram of the cosine distance among positive pairs (i.e., two arbitrary samples that belong to the same class) and negative pairs. We can see that the distance in softmax and triplet cases resemble each other; in the COCO case, the discrepancy between the positive and negative is larger (center distance on  $x$ -axis between two clusters); the intra-class similarity is also more polymerized (area under two clusters).

**Training loss.** Figure 3 describes the training loss curves under different loss strategies on CIFAR and LFW datasets. We can observe that our proposed method undergoes a stable convergence as the epoch goes; whereas other losses do not witness an obvious drop when the learning rate decreases. The triplet loss is effective the number of classes is small on CIFAR; however, when the task extends to a larger scale and includes over 1,000 categories, the triplet term suffers from severe disturbance during training since the optimized feature distance is altered frequently within one batch.

**Quantitative results.** We report the image classification error rate on MNIST and CIFAR-10 in Table 1. Note that MNIST is quite a small-scale benchmark and the performance tends to saturate. Moreover, we conduct the ablative study on different losses: softmax, center loss, triplet loss and their combination. For fair comparison, they all share the same CNN structure as COCO's. We can observe that the center loss alone bears a limited improvement (6.66 vs 6.70) over softmax; the triplet loss has the training instability concern and thus the error rate is much higher (12.69). Without resorting to softmax, our method is neat in formulation and achieves the lowest error (6.25).

## 4 COCO for Large-scale Recognition

### 4.1 Face Verification and Identification

We follow the general convention as in [42, 33, 25] to conduct face verification and identification: train a network using COCO loss to obtain robust features, extract features in the test set based on the trained model, and compare or identify the unknown classes given a large scale test setting. The main modifications to the traditional practice are two folds: first we employ the detection pipeline

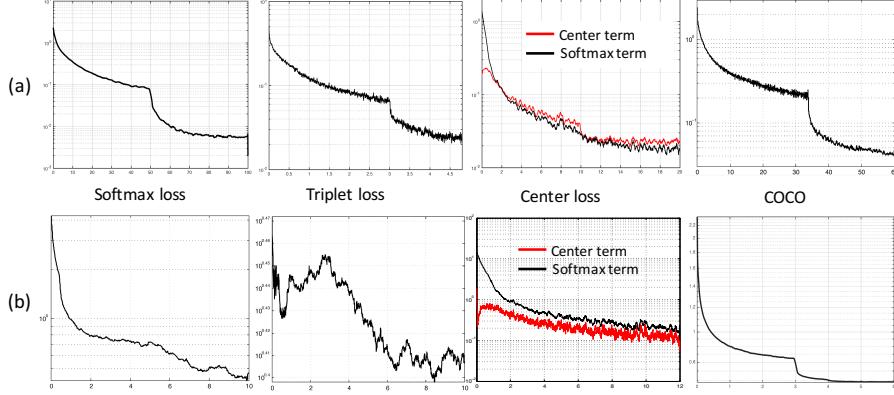


Figure 3: Training loss vs epoch of different losses. (a): Image classification on CIFAR-10 [13]; (b): Face verification on trained on MS-1M celebrity set. Note that for the center loss there are two curves, corresponding to the center and softmax term, respectively.

to leverage the feature selection and conduct landmark regression, in a multi-task RPN manner [32]. One task is for classifying the foreground or background landmarks while the other focus on regressing them. There are five landmarks: left or right eyes, nose, and left or right corners of mouth. The second revision is the landmark alignment via affine transformation, in order to switch the position to a base one.

## 4.2 Person Recognition

Method	MNIST	CIFAR-10
DropConnect [37]	0.57	9.41
NIN [23]	0.47	10.47
Maxout [6]	0.45	11.68
DSN [17]	0.39	9.69
R-CNN [5]	0.31	8.69
GenPool [16]	0.31	7.62
Softmax	0.36	6.70
Center loss + softmax	0.32	6.66
Triplet loss	1.45	12.69
Triplet loss + softmax	0.38	6.73
<b>COCO</b>	<b>0.30</b>	<b>6.25</b>

Table 1: Classification error rate (%) on MNIST and CIFAR-10 without data augmentation.

patch alignment are provided in the supplementary. Given the cropped and aligned patches, we train (finetune) models for different regions on PIPA training set using COCO loss. At testing stage, we measure the cosine similarity between two test splits to recognize the identity in `test_1` based on the labels in `test_0`. The similarity between two patches  $i$  and  $j$  in `test_1` and `test_0` is denoted as  $s(r)_{ij}$ , where  $r$  indicates a region. We normalize the preliminary  $s(r)_{ij}$  in order to have scores across different regions comparable:

$$s(r)_{ij} = \mathcal{C}(\mathbf{f}^{(i,r)}, \mathbf{f}^{(j,r)}), \quad \hat{s}(r)_{ij} = \left( 1 + \exp [ - (\beta_0 + \beta_1 s(r)_{ij}) ] \right)^{-1}, \quad (9)$$

where  $\beta_0, \beta_1$  are parameters of the logistic regression. The final score  $S_{ij}$  is a weighted mean of the normalized scores  $\hat{s}(r)_{ij}$  of each region:  $S_{ij} = \sum_{r=1}^R \gamma^r \cdot \hat{s}(r)_{ij}$ , where  $R$  is the total number of regions and  $\gamma^r$  being the weight of each region's score. The identity of patch  $i$  in `test_1` is decided

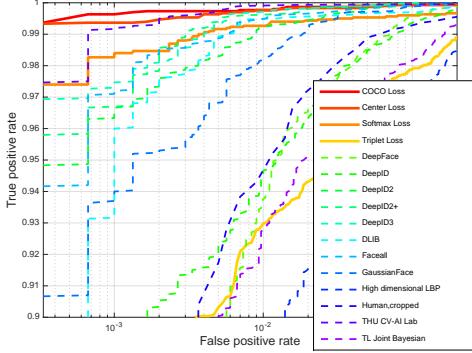


Figure 4: The ROC curves of different methods for face verification on LFW. Our model using COCO to train features achieves a higher AUC score.

Method	Train Data	mAcc
FaceNet [33]	200M	99.65
DeepID2 [35] *	300K	99.47
CenterFace [42]	700K	99.28
L-Softmax [26]	webface	98.71
SphereFace [25]	webface	99.42
Softmax	half MS-1M	99.75
Center loss + softmax	half MS-1M	99.78
Triplet loss	half MS-1M	98.85
Triplet loss + softmax	half MS-1M	99.68
COCO	half MS-1M	<b>99.86</b>

Table 2: Face verification accuracy (%) on LFW. The first three models use private outside training data. \* Result comes from an ensemble method that has 25 models.

by the label corresponding to the maximum score in the reference set:  $l_i = \arg \max_{j*} S_{ij}$ . The test parameters of  $\beta$  and  $\gamma$  are determined by a validation set on PIPA.

## 5 Experiments

**Dataset and evaluation metric.** *Face recognition.* The Labeled Face in the Wild (LFW) dataset [10] contains 13,233 web-collected images from 5749 different identities, with large variations in pose, expression and illuminations. Following the standard protocol of unrestricted with labeled outside data, we test on 6,000 face pairs. The MegaFace [12] is a very challenging dataset and aims to evaluate the performance of face recognition algorithms at the million scale of distractors: people who are not in the test set. It includes gallery set and probe set. The gallery set consists of more than one million images from 690K different individuals, as a subset of Flickr photos from Yahoo. The probe set descends from two existing databases: Facescrub and FGNet. *Person recognition.* The People In Photo Albums (PIPA) dataset [44] is divided into train, validation, test and leftover sets, where the head of each instance is annotated in all sets and it consists of over 60,000 instances of around 2,000 individuals. In this work, thanks to the discriminative and polymerized features learned by COCO, we take full advantage of the training set and remove the second training on `test_0`. Moreover, [28] introduced three more challenging splits besides the original test split, namely `album`, `time` and `day`. Each new split emphasizes different temporal distance (various albums, events, days, etc.) between the two subsets of the test data. The evaluation metric is the averaged classification accuracy over all instances on `test_1`.

### 5.1 Face Verification and Identification

We now apply the COCO feature learning to large-scale human recognition system. For fair comparison of our method to other losses (softmax, center loss, triplet loss and their combination), we use the same CNN network structure in the following experiments. Features on LFW is available on the github repository.

Figure 4 shows the ROC curves of different methods and Table 2 reports the accuracy (%) for face verification on LFW [10]. It is observed that COCO model is competitively superior (99.86) than its counterparts. Compared with FaceNet [33] which intakes around 200M outside training data, or DeepID2 [35] which ensembles 25 models, our approach demonstrates the effectiveness of feature learning. Table 3 illustrates the identification accuracy on MegaFace challenge [12], where the test identities share no interaction with those in training and the data scale is quite large. Our model using COCO is competent at different number of distractors. The combination of center loss with softmax ranks second (76.57 vs 75.79 @ 1M), which verifies that applying center loss alone cannot guarantee a good feature learning for large-scale task. The triplet loss (69.13) is inferior and our model competes favorably against those using large protocol of training data.

Method	Protocol	@1M	Method	Protocol	@1M
NTechLab	small	58.22	NTechLab	large	73.30
DeepSense	small	70.98	DeepSense	large	74.80
SphereFace [25]	small	75.77	ShanghaiTech	large	74.05
L-Softmax [26]	small	67.13	Google FaceNet	large	70.50
Method	Protocol	@1M	@100k	@10k	@1k
Softmax	small	71.17	85.78	93.22	96.14
Center loss + softmax	small	75.79	90.74	96.45	97.91
Triplet loss	small	69.13	85.64	93.38	96.66
Triplet loss + softmax	small	70.22	86.77	94.71	97.02
COCO	small	<b>76.57</b>	<b>91.77</b>	<b>96.72</b>	<b>98.03</b>

Table 3: Face identification accuracy (%) on MegaFace challenge at various number of distractors.

Method	Face	Head	Upper body	Whole body	original	album	time	day
RNN [22]	✓	✓			84.17	80.78	74.00	53.75
	✓		✓		89.24	81.46	<u>76.84</u>	<u>61.48</u>
	✓			✓	88.40	82.15	70.90	57.87
		✓	✓		88.76	79.15	68.64	42.91
		✓	✓		84.93	78.25	66.43	43.73
		✓		✓	87.43	77.54	67.40	42.30
			✓	✓	81.93	73.84	62.46	34.77
-	✓	✓	✓		87.86	80.85	71.65	59.03
-	✓	✓		✓	88.13	<u>82.87</u>	73.01	55.52
-		✓	✓	✓	89.71	78.29	66.60	52.21
-	✓		✓	✓	<u>91.43</u>	80.67	70.46	55.56
Softmax	✓	✓	✓	✓	88.73	80.26	71.56	50.36
-	✓	✓	✓	✓	<b>92.78</b>	<b>83.53</b>	<b>77.68</b>	<b>61.73</b>

Table 4: Person recognition on PIPA [44]. Investigation on the merging similarity score from different body regions during inference. The top two results in each split set are marked in **bold** and *italic*, respectively. COCO loss is applied in all cases if not specified.

## 5.2 Person Recognition

**Comparison with state-of-the-arts and visual results.** The comparison figure and visual results are provided in the supplementary due to page limit. We can see that our recognition system outperforms against previous state-of-the-arts in all four test splits. Note that the test identities share no interaction with those in the training set and thus this task is also a challenging open set verification problem. The visual results show some examples of the predicted instances by our model, where complex scenes with non-frontal faces and body occlusion can be handled properly in most scenarios. The last two columns show failure cases where our method could not differentiate the identities, this is probably due to the very similar appearance configuration in these scenarios.

**Ablation study on different regions.** Table 4 depicts the investigation on merging the similarity score from different body regions during inference. [22] also employs a multi-region processing step and we include it in Table 4. For some region combination, on certain splits our model is superior (*e.g.*, head plus upper body region, 88 vs 84 on original, 79 vs 78 on album, 68 vs 66 on time); whereas on other splits ours is inferior (*e.g.*, 42 vs 43 on day). This is probably due to distribution imbalance among splits: the upper body region differs greatly in appearance with some instances absent of this region, making COCO hard to learn the corresponding features. However, under the score integration scheme, the final prediction (last line) can complement features learned among different regions and achieves better performance against [22].

## 6 Conclusion

In this work, we re-investigate the feature learning problem based on the motivation that deeply learned features should be both discriminative and polymerized. We address the problem by proposing a congenerous cosine (COCO) loss, which optimizes the cosine distance among data features to simultaneously enlarge inter-class variation and intra-class similarity. COCO can be learned in a neat way with stable end-to-end training. We have extensively conducted experiments on five benchmarks, from image classification to face identification, to demonstrate the effectiveness of our approach, especially applying it to the large-scale human recognition tasks.

## References

- [1] D. Anguelov, K. chih Lee, S. B. Gokturk, and B. Sumengen. Contextual identity recognition in personal photo albums. In *CVPR*, 2007.
- [2] Z. Chi, H. Li, H. Lu, and M.-H. Yang. Dual deep network for visual tracking. *IEEE Trans. on Image Processing*, 2017.
- [3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.
- [4] R. Girshick. Fast R-CNN. In *ICCV*, 2015.
- [5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.
- [6] I. J. Goodfellow, D. Warde-farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. In *ICML*. 2013.
- [7] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. MS-Celeb-1M: a dataset and benchmark for large-scale face recognition. *arXiv preprint:1607.08221*, 2016.
- [8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [9] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313, 2006.
- [10] 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.
- [11] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*. 2015.
- [12] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *CVPR*, 2016.
- [13] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. In *Technical Report*, 2009.
- [14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012.
- [15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324, 1998.
- [16] C.-Y. Lee, P. W. Gallagher, and Z. Tu. Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. *arXiv preprint:1509.08985*, 2015.
- [17] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. *arXiv preprint:1409.5185*, 2014.
- [18] H. Li, J. Brandt, Z. Lin, X. Shen, and G. Hua. A multi-level contextual model for person recognition in photo albums. In *CVPR*, 2016.
- [19] H. Li, J. Chen, H. Lu, and Z. Chi. CNN for saliency detection with low-level feature integration. *Neurocomputing*, 226:212–220, 2017.
- [20] H. Li, Y. Liu, W. Ouyang, and X. Wang. Zoom out-and-in network with map attention decision for region proposal and object detection. In *arXiv preprint: 1709.04347*, 2017.
- [21] H. Li, W. Ouyang, and X. Wang. Multi-bias non-linear activation in deep neural networks. In *ICML*, 2016.
- [22] Y. Li, G. Lin, B. Zhuang, L. Liu, C. Shen, and A. van den Hengel. Sequential person recognition in photo albums with a recurrent network. *arXiv preprint:1611.09967*, 2016.
- [23] M. Lin, Q. Chen, and S. Yan. Network in network. In *ICLR*, 2014.
- [24] H. Liu, Y. Tian, Y. Wang, L. Pang, and T. Huang. Deep relative distance learning: Tell the difference between similar vehicles. In *CVPR*, 2016.
- [25] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017.

- [26] W. Liu, Y. Wen, Z. Yu, and M. Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, 2016.
- [27] Y. Liu, H. Li, J. Yan, F. Wei, X. Wang, and X. Tang. Recurrent scale approximation for object detection in CNN. In *IEEE International Conference on Computer Vision*, 2017.
- [28] S. J. Oh, R. Benenson, M. Fritz, and B. Schiele. Person recognition in personal photo collections. In *ICCV*, 2015.
- [29] S. J. Oh, R. Benenson, M. Fritz, and B. Schiele. Faceless person recognition; privacy implications in social media. In *ECCV*, 2016.
- [30] W. Ouyang, H. Li, X. Zeng, and X. Wang. Learning deep representation with large-scale attributes. In *ICCV*, 2015.
- [31] R. Ranjan, C. D. Castillo, and R. Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint:1703.09507*, 2017.
- [32] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NIPS*, 2015.
- [33] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.
- [34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.
- [35] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *CVPR*, 2015.
- [36] O. Tadmor, Y. Wexler, T. Rosenwein, S. Shalev-Shwartz, and A. Shashua. Learning a metric embedding for face recognition using the multibatch method. In *NIPS*, 2016.
- [37] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus. Regularization of neural networks using dropconnect. In *ICML*, 2013.
- [38] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille. Normface: L2 hypersphere embedding for face verification. *arXiv preprint:1704.06369*, 2017.
- [39] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015.
- [40] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016.
- [41] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 2009.
- [42] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016.
- [43] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *NIPS*, 2003.
- [44] N. Zhang, M. Paluri, Y. Taigman, R. Fergus, and L. Bourdev. Beyond frontal faces: Improving person recognition using multiple cues. In *CVPR*, 2015.
- [45] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao. Range loss for deep face recognition with long-tail. *arXiv preprint:1611.08976*, 2016.

## Supplementary

### 6.1 Proof on the optimal scale factor, Theorem 1

*Proof.* It is obvious that the infimum value of COCO loss is of the form:

$$\inf \mathcal{L}^{COCO} = -\log \frac{\exp s^+}{\exp s^+ + (K-1) \exp s^-}, \quad (10)$$

$$s^+ = \alpha \exp(\cos \theta) \leq \alpha, \quad (11)$$

$$s^- = \alpha \exp(\cos \theta) \geq -\alpha, \quad (12)$$

where  $s$  is the probability output in the final classification layer. For brevity, we denote  $\inf s = s^-$  and  $\sup s = s^+$ ;  $\theta$  is the angle between two features. Often we have  $\mathbb{E}[\cos \theta] \geq 0$  in practice and yet the minimal value of  $\cos \theta$  can still be  $-1$ . Therefore,

$$\sup \mathcal{L}^{COCO} = -\log \frac{\exp(-\alpha)}{\exp(-\alpha) + (K-1) \exp \alpha}. \quad (13)$$

The loss to minimize in an optimization problem usually meets some upper bound criteria, and we denote this target value as  $\epsilon$ , i.e.,  $\inf \mathcal{L}^{COCO} < \epsilon$ . Incorporating this prior knowledge into previous derivations, we have:

$$\log(\exp \alpha + (K-1) \exp(-\alpha)) < \alpha + \epsilon, \quad (14)$$

$$\alpha > \frac{1}{2} \log \frac{K-1}{\exp \epsilon - 1}. \quad (15)$$

□

It is empirically found that a typical value for  $\epsilon$  is  $10^{-4}$  and thus the scale factor can be expressed in a deterministic closed form:

$$\alpha = \frac{1}{2} \log(K-1) + 3. \quad (16)$$

### 6.2 Details on Image Classification and Face Recognition

For image classification on CIFAR, we use a simple ResNet-20 structure and train the network from scratch. We set the momentum as 0.9 and the weight decay to be 0.005. The base learning rate is set to be 0.1, 0.1, 0.05, respectively. We drop the learning rate by 10way and stop to decrease the learning rate until it reaches a minimum value (0.0001). All the convolutional layers are initialized with Gaussian distribution with mean of zero and standard variation of 0.05 or 0.1.

For face recognition, we use the Inception ResNet model and add a new fully-connected layer after the final global pooling layer to generate a 128-dim feature. The model is trained from scratch. The training data is a subset of the Microsoft-1M celebrity dataset, which consists of 80k identities with 3M images in total, **removing all overlapping IDs**. For COCO, softmax and center loss, we train the network with SGD solver with a base learning rate 0.01 and drop it to 10% every five epoch. The total training time is 12 epoch. The momentum is set to 0.9. Batch normalization [11] with moving average is employed. For the triplet loss, we have a base learning rate to 0.1 with a bigger batch size 128\*3 (it is hard to converge if the size is less than 64\*3); we drop the rate to 10% every 90k iteration. The total training time is around 300k iterations.

**Evaluation metric on face recognition.** For verification, the test set is LFW [10] and the metric is under standard ‘Unrestricted, Labeled Outside Data’ setting of LFW. It verifies 6,000 pairs to determine if they belong to the same person. We report ROC curve and the mean accuracy. For identification on MegaFace [12], we have the test set into two subsets: the first is probe set that consists of FaceScrub (3530 images of 80 persons); the second is gallery set that consists of distractors (1M images with no overlap with FaceScrub). The test metric is under standard MegaFace Challenge-1 setting. We report the CMC curve in terms of Top-1 accuracy with 10, 100, 1000, 10k, 100k and 1M distractors.

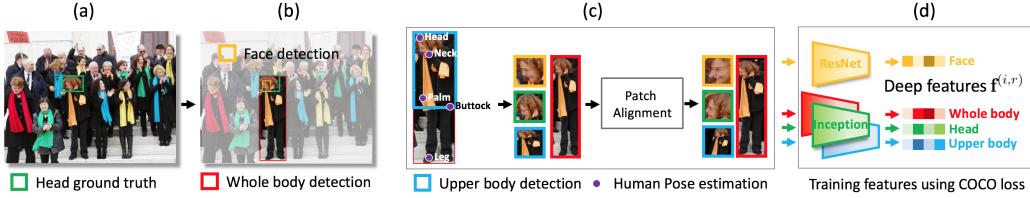


Figure 5: Training workflow of person recognition task. (a) Each person is labelled with a ground-truth box of the head. (b): Face, whole body and upper body detection. (c): Each region (patch) is aligned to a base position to alleviate the inner-class variance. (d) Each aligned patch is further fed into a deep model to obtain representative and robust features using COCO loss.

Test split	Face		Head		Upper body			Whole body		
	n-a	ali	n-a	ali	[22]	n-a	ali	[22]	n-a	ali
original	95.47	97.45	74.23	82.69	81.75	76.67	80.75	79.92	75.04	79.06
album	94.66	96.57	65.47	73.77	74.21	66.23	69.58	70.78	64.21	67.27
time	91.03	93.36	55.88	64.31	63.73	55.24	57.40	58.80	55.53	54.62
day	90.36	91.32	35.27	44.24	42.75	26.49	32.09	34.61	32.85	29.59

Table 5: Ablation study on body regions and feature alignment using softmax as loss. We report the classification accuracy (%) where n-a denotes the non-alignment case and ali indicates the alignment case. Note that for face region, we only evaluate instances with faces.

### 6.3 More Algorithm Details for Person Recognition

The COCO features are trained on four regions (thus four models), namely, *face*, *head*, *upper body* and *whole body*. We take the face region as an illustration for the alignment scheme. A face detector is first pretrained in a RPN spirit [32]. The face detector identifies some keypoints (eye, brow, mouth, etc.) and we align the detected face patch to a base location via an affine transformation. Let  $p, q \in \mathbb{R}^{z \times 2}$  denote  $m$  keypoints and the aligned locations, respectively. We define  $P, Q$  as two affine spaces, then the transformation  $\mathcal{A} : P \mapsto Q$  is defined as:  $p \mapsto q = \mathbf{A}p + b$ , where  $\mathbf{A} \in \mathbb{R}^{z \times z}$  is a linear transformation matrix in  $P$  and  $b \in \mathbb{R}^{z \times 2}$  being the bias in  $Q$ . Such an alignment scheme ensures samples both within and across categories do not have large variance: if the model is learned without alignment, it has to distinguish more patterns, e.g., different rotations among persons, making it more prone to overfitting; with alignment, it can better classify features of various identities despite of rotation, viewpoint, translation, etc. Given the cropped and aligned patches, we finetune the face model on PIPA training set using COCO loss.

The head region is given as the ground truth for each person. To detect a whole body, we also pre-train a detector in the RPN framework. The model is trained on the large-scale human celebrity dataset [7], where we use the first 87021 identities in 4638717 images. The network structure is an inception model [11] with the final pooling layer replaced by a fully connected layer. To determine the upper body region, we conduct human pose estimation [40] to identify keypoints of the body and the upper part is thereby located by these points. The head, whole body and upper body models, which are used for COCO loss training, are finetuned on PIPA training set using the pretrained inception model, following similar procedure of patch alignment stated previously for the face region. The aligned patches of four regions are shown in Figure 5(c).

### 6.4 Experiments on Person Recognition

**Ablation study on feature alignment for different regions.** Table 5 reports the performance of using feature alignment and different body regions, where several remarks could be observed. First, the alignment case in each region performs better by a large margin than the non-alignment case, which verifies the motivation of patch alignment to alleviate inner-class variance stated in the main paper. Second, for the alignment case, the most representative features to identify a person reside in the region of face, followed by head, upper body and whole body at last. Such a clue is not that

Methods	original	album	time	day
PIPER [44]	83.05	-	-	-
RNN [22]	84.93	78.25	66.43	43.73
Naeil [28]	86.78	78.72	69.29	46.61
Multi-context [18]	88.75	83.33	77.00	59.35
Ours	<b>92.78</b>	<b>83.53</b>	<b>77.68</b>	<b>61.73</b>

Table 6: Verification accuracy (%) comparison to state-of-the-arts on PIPA [44] for person recognition.

obvious for the non-alignment case. Third, we notice that for the whole body region, accuracy in the non-alignment case is higher than that of the alignment case in time and day. This is probably due to the improper definition of base points on these two sets.

**Visual results.** Figure 6 shows some visualization results for person recognition task.



Figure 6: Visualization of our method on the PIPA `test_1` set. Given the image as input to be predicted in (a), its nearest neighbour in the feature space from `text_0` is shown in (b). The identity of the input is determined by the label of its neighbour. Our model can handle complex scenes with non-frontal faces and body occlusion. The last two columns show failure cases.