Hyperparameter-Free Out-of-Distribution Detection Using Cosine Similarity

Engkarat Techapanurak[1][0000−0002−4908−6293],
Masanori Suganuma[1,2][0000−0002−1469−9663], and
Takayuki Okatani[1,2][0000−0001−9222−763X]

[1] Tohoku University, Sendai, 980-8579, Japan
[2] RIKEN Center for AIP, Japan
{engkarat,suganuma,okatani}@vision.is.tohoku.ac.jp

Abstract. The ability to detect out-of-distribution (OOD) samples is vital to secure the reliability of deep neural networks in real-world applications. Considering the nature of OOD samples, detection methods should not have hyperparameters that need to be tuned depending on incoming OOD samples. However, most recently proposed methods do not meet this requirement, leading to a compromised performance in real-world applications. In this paper, we propose a simple and computationally efficient, hyperparameter-free method that uses cosine similarity. Although recent studies show its effectiveness for metric learning, it remains uncertain if cosine similarity works well also for OOD detection. There are several differences in the design of output layers from the metric learning methods; they are essential to achieve the best performance. We show through experiments that our method outperforms the existing methods on the evaluation test recently proposed by Shafaei et al., which takes the above issue of hyperparameter dependency into account; it achieves at least comparable performance to the state-of-the-art on the conventional test, where other methods but ours are allowed to use explicit OOD samples for determining hyperparameters. Lastly, we provide a brief discussion of why cosine similarity works so well, referring to an explanation by Hsu et al.

1 Introduction

It is widely recognized that deep neural networks tend to show unpredictable behaviors for *out-of-distribution* (OOD) samples, i.e., samples coming from a different distribution from that of the training samples. They often give high confidence (i.e., high softmax value) to OOD samples, not only to *in-distribution* (ID) samples (i.e., test samples from the same distribution as the training samples). Therefore, it has been a major research topic to detect OOD samples in classification performed by deep neural networks; many methods have been proposed so far [1–8].

A problem with the existing methods, especially those currently recognized as the state-of-the-art in the community, is that they have hyperparameters specific to OOD detection. They determine these hyperparameters using a certain

amount of OOD samples as 'validation' data; that is, these studies assume the availability of (at least a small amount of) OOD samples. This assumption, however, is unlikely to hold true in practice; considering the definition of OOD, it is more natural to assume its distribution to be unknown. Even when the assumption is indeed wrong, it will be fine if OOD detection performance is insensitive to the choice of the hyperparamters, more rigorously, *if the hyperparameters tuned on the assumed OOD samples generalize well to incoming OOD samples we encounter in practice.* However, a recent study [9] indicates that this is not the case, concluding that none of the existing methods is ready to use, especially for the tasks with high-dimensional data space, e.g., image classification.

In this paper, we propose a novel method that uses cosine similarity for OOD detection, in which class probabilities are modeled using softmax of scaled cosine similarity. It is free of any hyperparameters associated with OOD detection, and thus there is no need to access OOD samples to determine hyperparameters, making the proposed method free from the above issue. We show through experiments that it outperforms the existing methods by a large margin on the recently proposed test [9], which takes the above issue of hyperparameter dependency into account; it also attains at least comparable performance to the state-of-the-art methods on the conventional test, in which the other methods but ours tune hyperparameters using explicit OOD samples.

It should be noted that a concurrent work [10] also shows the effectiveness of softmax of the scaled cosine similarity for OOD detection. Our method is technically mostly the same, but the present paper shows several different results/conclusions from their paper. The paper [10] shows a conjecture that the scaling factor of the cosine similarity approximates the probability of an input being in-distribution, contributing to improved detection performance. In this paper, however, we show empirical evidence that this is not the case. It is also noted that, although recent methods for metric learning [11–16] similarly employ scaled cosine similarity as well, they do not guarantee its effectiveness on OOD detection. There are several differences from them in the output layer's design, which contributes to detection accuracy. Concerning this, we provide a detailed ablation study to clarify the method's differences from common metric learning approaches.

2 Related Works

2.1 Uncertainty of Prediction

It is known that when applied to classification tasks, deep neural networks often exhibit overconfidence for unseen inputs. Many studies have been conducted to find a solution to this issue. A popular approach is to evaluate uncertainty of a prediction and use it as its reliability measure. There are many studies on this approach, most of which are based on the framework of Bayesian neural networks or its approximation [17–20]. It is reported that predicted uncertainty is useful for real-world applications [21–25]. However, it is still an open problem to accurately evaluate uncertainty. There are also studies on calibration of confidence scores

[26–28]. Some studies propose to build a meta system overseeing the classifier that can estimate the reliability of its prediction [29, 30].

2.2 Out-of-distribution (OOD) Detection

Detection Methods. A more direct approach to the above issue is OOD detection. A baseline method that thresholds confidence score, i.e., the maximum softmax output, is evaluated in [1]. This study presents a design of experiments for evaluation of OOD detection methods, which has been employed in the subsequent studies. Since then, many studies have been conducted. It should be noted that these methods have hyperparameters for OOD detection, which need to be determined in some way. Some studies assume a portion of OOD samples to be given and regard them as a 'validation' set, by which the hyperparemters are determined.

ODIN [2] applies perturbation with a constant magnitude ϵ to an input $\boldsymbol{x}$ in the direction of increasing the confidence score (i.e., the maximum softmax) and then uses the increased score in the same way as the baseline. An observation behind this procedure is that such perturbation tends to increase confidence score more for ID samples than for OOD samples. Rigorously, $\boldsymbol{x}$ is perturbed to increase a temperature-scaled softmax value. Thus, ODIN has two hyperparameters ϵ and the softmax temperature. In the experiments reported in [2], ϵ as well as the temperature are determined by using a portion of samples from a target OOD dataset; this is done for each pair of ID and OOD datasets.

The current state-of-the-art of OOD detection is achieved by the methods [4, 3] employing input perturbation similar to ODIN. It should be noted that there are many studies with different motivations, such as generative models [31, 32], a prior distribution [6], robustification by training networks to predict word embedding of class labels [5], pretraining of networks [33, 34], and batch-wise fine-tuning [7].

In [4], a method that employs an ensemble of networks and similar input perturbation is proposed, achieving the state-of-the-art performance. In the training step of this method, ID classes are split into two sets, one of which is virtually treated as ID classes and the other as OOD classes. A network is then trained so that the entropy for the former samples is minimized while that for the latter samples is maximized. Repeating this for different K splits of classes yields K leave-out classifiers (i.e., networks). At test time, an input $\boldsymbol{x}$ is given to these K networks, whose outputs are summed to calculate ID class scores and an OOD score, where $\boldsymbol{x}$ is perturbed with magnitude ϵ in the direction of minimizing the entropy. In the experiments, ϵ, the temperature, and additional hyperparameters are determined by selecting a particular dataset (i.e., iSUN [35]) as the OOD dataset, and OOD detection performance on different OOD datasets is evaluated.

In [3], another method is proposed, which models layer activation over ID samples with class-wise Gaussian distributions. It uses the induced Mahalanobis distances to class centroids for conducting the classification as well as OOD detection. It employs logistic regression integrating information from multiple

layers and input perturbation similar to ODIN, which possesses several hyperparameters. For their determination, it is suggested to use explicit OOD samples, as in ODIN [2]. Another method is additionally suggested to avoid this potentially unrealistic assumption, which is to create adversarial examples for ID samples [36] and use them as OOD samples, determining the hyperparameters. However, even this method is not free of hyperparameters; the creation of adversarial examples needs at least one (i.e., perturbation magnitude). It is not discussed how to choose it in their paper.

Evaluation Methods. Most of the recent studies employ the following evaluation method [1]. Specifying a pair of ID and OOD datasets (e.g., CIFAR-10 for ID and SVHN for OOD), it measures accuracy of distinguishing the OOD samples and ID samples. As the task is detection, appropriate metrics are used, such as accuracy at true positive rate (TPR) = 95%, area under the ROC curve (AUROC), and under the precision-recall curve (AUPR). As is noted in Sec. 1, most of the existing methods assume the availability of OOD samples and use them to determine their hyperparameters. Note that these OOD samples are selected from the *true* OOD dataset specified in this evaluation method. We will refer to this *one-vs-one* evaluation.

Recently, Shafaei et al. have raised a concern about the dependency of the existing methods on the explicit knowledge of the true OOD dataset, and proposed a novel evaluation method that aims at measuring the practical performance of OOD detection [9]. It assumes an ID dataset and multiple OOD datasets $\mathcal{D} = \{D_1, \ldots\}$ for evaluation. Then, the evaluation starts with choosing one dataset $D_i \in \mathcal{D}$ and use the samples from it to determine the hyperparameters of the method under evaluation; it then evaluates its detection accuracy when regarding each of the other datasets in $\mathcal{D}$ (i.e., $\mathcal{D} \backslash D_i$) as the OOD dataset, reporting the average accuracy over $\mathcal{D} \backslash D_i$. Note that this test returns the accuracy for each dataset in $\mathcal{D}$ (used for the assumed OOD dataset). We will refer to this *less-biased* evaluation.

2.3 Cosine Similarity

The proposed method employs softmax of scaled cosine similarity instead of ordinary softmax of logits. A similar approach has already been employed in recent studies of metric learning, such as L_2-constrained softmax [11], SphereFace [12], NormFace [13], CosFace [14], ArcFace [15], AdaCos [16], etc. Although it may seem straightforward to apply these methods to OOD detection, to the authors' knowledge, there is no study that has tried this before.

These metric learning methods are identical in that they use cosine similarity. They differ in i) if and how the weight $\boldsymbol{w}$ or the feature $\boldsymbol{f}$ of the last layer are normalized; ii) if and how margins are used with the cosine similarity to encourage maximization of inter-class variance and minimization of intra-class variance; and iii) how the scale parameter (i.e., s in (3)) is treated, i.e., as either a hyperparameter, a learnable parameter [13], or other [16]. According to this

categorization, our method is the most similar to NormFace [13] and AdaCos [16], in which both $\boldsymbol{w}$ and $\boldsymbol{f}$ are normalized and no margin is utilized. However, our method still differs from these metric learning methods in that it predicts s along with class probabilities at inference time. Ours also differs in that it uses a single fully-connected layer to compute the cosine similarity, whereas these metric learning methods use two fully-connected layers.

3 Proposed Method

3.1 Softmax of Scaled Cosine Similarity

The standard formulation of multi-class classification is to make the network predict class probabilities for an input, and use cross-entropy loss to evaluate the correctness of the prediction. The predicted class probabilities are obtained by applying softmax to the linear transform $\boldsymbol{W}\boldsymbol{f}+\boldsymbol{b}$ of the activation or feature $\boldsymbol{f}$ of the last layer, and then the loss is calculated assuming 1-of-K coding of the true class c as

$$\mathcal{L} = -\log \frac{e^{\boldsymbol{w}_c^\top \boldsymbol{f}+b_c}}{\sum_{i=1}^{C} e^{\boldsymbol{w}_i^\top \boldsymbol{f}+b_i}}, \tag{1}$$

where $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_C]^\top$ and $\boldsymbol{b} = [b_1, \ldots, b_C]^\top$.

Metric learning attempts to learn feature space suitable for the purpose of open-set classification, e.g., face verification. Unlike earlier methods employing triplet loss [37, 38] and contrastive loss [39, 40], recent methods [13–15] modify the loss (1) and minimize the cross entropy loss as with the standard multi-class classification. The main idea is to use the cosine of the angle between the weight $\boldsymbol{w}_i$ and the feature $\boldsymbol{f}$ as a class score. Specifically, $\cos\theta_i \equiv \boldsymbol{w}_i^\top \boldsymbol{f}/(\|\boldsymbol{w}_i\|\|\boldsymbol{f}\|)$ is used instead of the logit $\boldsymbol{w}_i^\top \boldsymbol{f} + b_i$ in (1); then a new loss is given as

$$\mathcal{L} = -\log \frac{e^{\cos\theta_c}}{\sum_{i=1}^{C} e^{\cos\theta_i}}. \tag{2}$$

The behavior of softmax, i.e., how soft its maximum operation will be, depends on the distribution of its inputs, which can be controlled by a scaling parameter of the inputs, called temperature T. This parameter is used for several purposes [41, 26]. In metric learning methods, it is employed to widen the range $[-1, 1]$ of $\cos\theta_i$'s inputted to softmax; specifically, all the input cosine $\cos\theta_i$'s are scaled by a parameter $s(= 1/T)$, revising the above loss as

$$\mathcal{L} = -\log \frac{e^{s\cos\theta_c}}{\sum_{i=1}^{C} e^{s\cos\theta_i}}. \tag{3}$$

3.2 Predicting the Scaling Parameter

In most of the metric learning methods employing similar loss functions, the scaling parameter s in (3) is treated as either a hyperparameter chosen in a

validation step or a parameter automatically determined in the training step [13, 16] There is yet another method for determining s, which is to predict it from $\boldsymbol{f}$ together with class probabilities. This makes the method hyperparameter-free. Moreover, we empirically found that this performs the best. Among several ways of computing s from $\boldsymbol{f}$, the following works the best:

$$s = \exp\{\mathrm{BN}(\boldsymbol{w}_s^\top \boldsymbol{f} + b_s)\}, \tag{4}$$

where BN is batch normalization [42], and $\boldsymbol{w}_s$ and b_s are the weight and bias of the added branch to predict s.

3.3 Design of the Output Layer

In the aforementioned studies of metric learning, ResNets are employed as a base network and are modified to implement the softmax of cosine similarity. Modern CNNs like ResNets are usually designed to have a single fully-connected (FC) layer between the final pooling layer (i.e., global average pooling) and the network output. As ReLU activation function is applied to the inputs of the pooling layer, if we use the last FC layer for computing cosine similarity (i.e., treating its input as $\boldsymbol{f}$ and its weights as $\boldsymbol{w}_i$'s), then the elements of $\boldsymbol{f}$ take only non-negative values. Thus, the metric learning methods add an extra single FC layer on top of the FC layer and use the output of the first FC layer as $\boldsymbol{f}$, making $\boldsymbol{f}$ (after normalization) distribute on the whole hypersphere. In short, the metric learning methods employ two FC layers at the final section of the network.

However, we found that for the purpose of OOD detection, having two fully-connected layers does not perform better than simply using the output of the final pooling layer as $\boldsymbol{f}$. Details will be given in our experimental results. Note that in the case of a single FC layer, as $\boldsymbol{f}$ takes only non-negative values, $\boldsymbol{f}$ resides in the first quadrant of the space, which is very narrow subspace comparative to the entire space.

To train the modified network, we use a standard method. In our experiments, we employ SGD with weight decay as the optimizer, as in the previous studies of OOD detection [2, 3, 5, 4]. In several studies of metric learning [14, 15, 43], weight decay is also employed on all the layers of networks. However, it may have different effects on the last layer of the network employing cosine similarity, where weights are normalized and thus its length does not affect the loss. In our experiments, we found that it works better when we do not apply weight decay to the last layer.

3.4 Detecting OOD samples

Detecting OOD samples is performed in the following way. Given an input $\boldsymbol{x}$, our network computes $\cos\theta_i$ $(i = 1, \ldots, C)$. Let i_{max} be the index of the maximum of these cosine values. We use $\cos\theta_{i_{max}}$ for distinguishing ID and OOD samples. To be specific, setting a threshold, we declare $\boldsymbol{x}$ is an OOD sample if $\cos\theta_{i_{max}}$ is lower than it. Otherwise, we classify $\boldsymbol{x}$ into the class i_{max} with the predicted probability $e^{s\cos\theta_{i_{max}}} / \sum e^{s\cos\theta_i}$.

4 Experimental Results

4.1 Experimental Settings

We conducted experiments to evaluate the proposed method and compare it with existing methods.

Evaluation Methods. We employ the one-vs-one and less-biased evaluation methods explained in Sec. 2.2. The major difference between the two is in the assumption of prior knowledge about OOD datasets, which affects the determination of the hyperparameters of the OOD detection methods under evaluation. Note therefore that *the difference does not matter for our method*, as it does not need any hyperparameter; it only affects the other compared methods.

One-vs-one evaluation This evaluation assumes one ID and one OOD datasets. A network is trained on the ID dataset and each method attempts to distinguish ID and OOD samples using the network. Each method may use a fixed number of samples from the specified OOD datasets for its hyperparameter determination. We followed the experimental configurations commonly employed in the previous studies [2–4].

Less-biased evaluation This evaluation uses one ID and many OOD datasets. Each method may access one of the OOD datasets to determine its hyperparameters but its evaluation is conducted on the task of distinguishing the ID samples and samples from each of the other OOD datasets. We followed the study of Shafaei et al. [9] with slight modifications. First, we use AUROC instead of detection accuracy for evaluation metrics (additionally, accuracy at TPR= 95% and AUPR-IN in the supplementary material), as we believe that they are better metrics for detection tasks, and they are employed in the one-vs-one evaluation. Second, we add more OOD datasets to those used in their study to further increase the effectiveness and practicality of the evaluation.

Tasks and Datasets. We use CIFAR-10/100 for the target classification tasks in all the experiments. Using them as ID datasets, we use the following OOD datasets in one-vs-one evaluation: TinyImageNet (cropped and resized) [44], LSUN (cropped and resized) [45], iSUN [35],[3] SVHN [46] and Food-101 [47] For the less-biased evaluation, we additionally use STL-10 [48], MNIST [49], NotMNIST, and Fashion MNIST [50]. As for STL-10 and Food-101, we resize their images to 32×32 pixels.

Remark We found that the cropped images of TinyImageNet and LSUN that are provided by the GitHub repository of [2], which are employed in many recent studies, have a black frame of two-pixel width around them; see the supplementary material for details. Although we are not sure if this is intentional,

[3] Datasets are available at `https://github.com/facebookresearch/odin`.

considering that the frame will make OOD detection easier, we use two versions with/without the black frame in our experiments; the frame-free version is indicated by '$*$' in what follows. In the main paper, we show mainly results on the frame-free versions. Those on the original versions are shown in the supplementary material, although it does not affect our conclusion.

Networks and Their Training. For networks, we employ the two CNNs commonly used in the previous studies, i.e., *Wide ResNet* [51] and *DenseNet* [52] as the base networks. Following [2], we use WRN-28-10 and DenseNet-100-12 having 100 layers with growth rate 12. The former is trained with batch size $= 128$ for 200 epochs with weight decay $= 0.0005$, and the latter is trained with batch size $= 64$ for 300 epochs with weight decay $= 0.0001$. Dropout is not used in the both networks. We employ a learning rate schedule, where the learning rate starts with 0.1 and decreases by 1/10 at 50% and 75% of the training steps.

The proposed method modifies the final layer and the loss of the base networks. Table 1 shows comparisons between the base networks and their modified version. The numbers are an average over five runs and their standard deviations are shown in parenthesis. It is seen that the modification tends to lower classification accuracy by a small amount. If this difference does matter, one may use the proposed network only for OOD detection and the standard network for ID classification.

Table 1. Performance of the base networks and their modified versions for the proposed method for the task of classification of ID (in-distribution) samples.

Network	In-Dist	Testing Accuracy	
		Standard	Cosine
Dense-100-12	CIFAR-10	95.11(0.10)	94.92(0.04)
	CIFAR-100	76.97(0.24)	75.65(0.12)
WRN-28-10	CIFAR-10	95.99(0.09)	95.72(0.05)
	CIFAR-100	81.04(0.37)	78.53(0.28)

Compared Methods. The methods we compare are as follows: the baseline method [1], ODIN [2], Mahalanobis detector [3][4], and leave-out ensemble [4]. The last two methods are reported to achieve the highest performance in the case of a single network and multiple networks, respectively. We conduct experiments separately with the first three and the last one due to the difference in settings. We report those with the leave-out ensemble in the supplementary material.

All these methods (but the baseline) have hyperparameters for OOD detection. For ODIN and the Mahalanobis detector, we follow the authors' methods

[4] We used the publicly available code: `https://github.com/pokaxpoka/deep_Mahalanobis_detector`

[2, 3] to determine them using a portion of the true OOD dataset. For the leave-out ensemble (comparisons in the supplementary material), we use the values of detection accuracy from its paper [4], in which the authors use a specific OOD dataset (iSUN) for hyperparameter determination.

4.2 Comparison by Less-biased Evaluation

We first show the performance of the four methods, i.e., the baseline, ODIN, the Mahalanobis detector, and ours, measured by the less-biased evaluation method. Figure 1 shows the results[5]. The details of the experimental settings are as follows. We use either CIFAR-10 or CIFAR-100 for the ID dataset. For the actual OOD dataset, we choose one of the eleven datasets described above and evaluate the OOD detection performance on each of the eleven ID-OOD pairs. For each ID-OOD pair, we use one of the rest (i.e., ten datasets) as a hypothesized OOD dataset, using which the hyperparameters are chosen for ODIN and Mahalanobis detector. We iterate this for the ten datasets. For each method/setting, we evaluate five models trained from different initial values. Finally, we calculate, for each method on each ID-OOD pair, the mean and standard deviation of AUROC (a bar and its error bar in Fig. 1) over the five models and the ten hypothesized datasets (for ODIN and Mahalanobis).

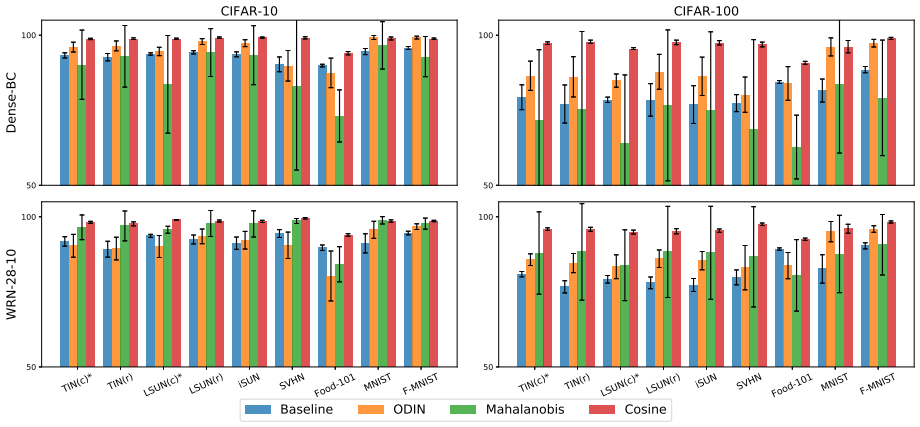

Fig. 1. OOD detection performance (AUROC) measured by the less-biased evaluation [9] for the baseline method [1], ODIN, [2] and the Mahalanobis detector [3], and the proposed one (denoted as 'Cosine'). Other metrics, i.e., accuracy at TPR=95% and AUPR-IN, are reported in the supplementary material.

It is seen from Fig. 1 that the proposed method consistently achieves better performance than others. It is noted that the Mahalanobis detector, which shows

[5] A complete table including other metrics, i.e., accuracy at TPR= 95% and AUPR-IN, is shown in the supplementary material.

the state-of-the-art performance in the conventional (i.e., one-vs-one) evaluation, shows unstable behaviors; the mean of AUROC tends to vary significantly and the standard deviation is very large depending on the dataset used for hyperparameter determination. The same observation applies to ODIN.

This clearly demonstrates the issue with these methods, that is, their performance is dependent on the choice of the hyperparameters. On the other hand, the proposed method performs consistently for all the cases. This is also confirmed from Fig. 2, which shows a different plot of the same experimental result; it shows AUROC for a *single* OOD dataset instead of the *mean* over multiple OOD datasets shown in Fig. 1. It is seen that the performance of the Mahalanobis detector varies a lot depending on the assumed OOD dataset. Additionally, it can be seen that the dataset yielding the highest performance differs for different true OOD datasets; iSUN, TIN(r), or Gaussian etc. is the best for detecting LSUN(r) as OOD, whereas F-MNIST or NotMNIST is the best for detecting MNIST as OOD.

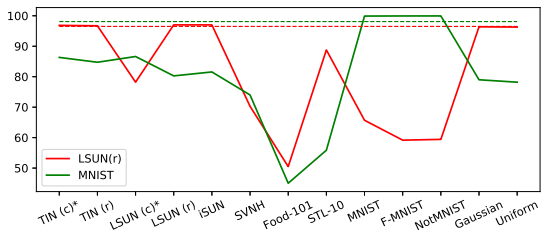

Fig. 2. Dependency of detection performance (AUROC) on the assumed OOD datasets (whose names are given in the horizontal axis) used for determining hyperparameters. Mahalanobis detector (solid lines) [3] and our method (broken lines). CIFAR-100 is used as ID and either LSUN(r) (in red color) or MNIST (in green color) is used as true OOD. DenseNet-100-12 is used for the network. Our method does not have hyperparameters and thus is independent of the assumed OOD dataset.

4.3 Comparison by One-vs-one Evaluation

We then show the comparison of the same four methods in the one-vs-one evaluation, which is employed in the majority of the previous studies. We ran each method five times from the training step, where the network weights are initialized randomly, and report the mean and standard deviation here. Table 2 shows the results. It is observed that the proposed method achieves better or at least competitive performance to the others. When using DenseNet-100-12, the proposed method consistently achieves higher performance than the Mahalanobis detector on almost all the datasets.

Table 2. OOD detection performance of the four methods measured by conventional one-vs-one evaluation.

ID	OOD	AUROC			
		Base [1]	ODIN [2]	Maha [3]	Cosine
Dense-100-12 CIFAR-10	TIN (c)	94.90(0.43)	98.79(0.32)	94.48(1.19)	**98.89(0.24)**
	TIN (c)*	93.26(0.85)	96.67(0.97)	97.36(0.39)	**98.74(0.23)**
	TIN (r)	92.67(1.23)	97.20(1.17)	**98.91(0.23)**	98.82(0.29)
	LSUN (c)	95.57(0.20)	98.48(0.14)	89.06(3.21)	**99.09(0.12)**
	LSUN (c)*	93.72(0.39)	96.41(0.52)	93.63(0.69)	**98.83(0.18)**
	LSUN (r)	94.28(0.52)	98.43(0.49)	99.00(0.23)	**99.19(0.22)**
	iSUN	93.62(0.83)	97.92(0.71)	98.95(0.21)	**99.20(0.19)**
	SVNH	90.28(2.47)	95.11(0.48)	98.89(0.37)	**99.11(0.36)**
	Food-101	89.87(0.44)	92.06(0.71)	80.38(3.83)	**93.98(0.54)**
Dense-100-12 CIFAR-100	TIN (c)	83.70(4.00)	94.48(3.21)	92.97(1.63)	**97.90(0.29)**
	TIN (c)*	79.32(4.14)	88.54(4.27)	93.18(0.39)	**97.31(0.45)**
	TIN (r)	77.07(6.35)	88.14(6.92)	96.81(0.27)	**97.82(0.53)**
	LSUN (c)	82.92(0.59)	94.72(0.59)	91.65(2.96)	**96.73(0.31)**
	LSUN (c)*	78.46(0.91)	87.89(1.13)	85.44(1.85)	**95.52(0.32)**
	LSUN (r)	78.44(5.41)	90.38(4.76)	97.00(0.15)	**97.59(0.75)**
	iSUN	76.89(6.28)	88.27(6.49)	97.04(0.10)	**97.45(0.73)**
	SVNH	77.36(2.83)	91.60(0.73)	96.48(0.68)	**96.90(0.79)**
	Food-101	84.38(0.48)	**90.82(0.60)**	67.14(1.39)	90.79(0.49)

ID	OOD	AUROC			
		Base [1]	ODIN [2]	Maha [3]	Cosine
WRN-28-10 CIFAR-10	TIN (c)	93.86(0.90)	95.88(1.01)	95.99(1.04)	**98.35(0.32)**
	TIN (c)*	91.79(1.57)	92.17(2.19)	**98.50(0.11)**	98.17(0.33)
	TIN (r)	89.21(2.65)	90.60(3.21)	**99.15(0.18)**	97.65(0.66)
	LSUN (c)	95.41(0.26)	97.20(0.15)	92.65(1.33)	**99.19(0.07)**
	LSUN (c)*	93.67(0.50)	95.08(0.42)	96.90(0.35)	**98.98(0.07)**
	LSUN (r)	92.45(1.48)	94.48(1.70)	**99.37(0.13)**	98.59(0.34)
	iSUN	91.22(2.05)	93.25(2.43)	**99.29(0.10)**	98.48(0.36)
	SVNH	94.43(1.30)	93.34(3.60)	99.28(0.09)	**99.52(0.24)**
	Food-101	89.71(0.90)	89.18(2.37)	90.43(1.54)	**93.95(0.41)**
WRN-28-10 CIFAR-100	TIN (c)	84.47(1.24)	91.72(1.10)	92.58(2.60)	**96.76(0.34)**
	TIN (c)*	80.90(0.90)	87.08(1.29)	**96.45(0.30)**	95.91(0.42)
	TIN (r)	76.67(2.03)	86.28(2.43)	**97.82(0.13)**	95.84(0.67)
	LSUN (c)	81.91(1.31)	91.75(0.44)	80.48(1.14)	**96.09(0.62)**
	LSUN (c)*	79.17(1.25)	88.06(0.46)	91.13(0.52)	**94.92(0.65)**
	LSUN (r)	78.00(1.95)	87.90(1.83)	**97.80(0.15)**	95.18(0.86)
	iSUN	77.29(2.15)	87.07(2.00)	**97.66(0.14)**	95.39(0.55)
	SVNH	79.82(2.49)	93.46(1.05)	**97.96(0.49)**	97.52(0.41)
	Food-101	89.25(0.40)	90.76(0.35)	91.15(0.66)	**92.53(0.38)**

4.4 Ablation Study

Although the proposed method employs softmax of cosine similarity equivalent to metric learning methods, there are differences in detailed designs, even compared with the most similar NormFace [13]. To be specific, they are the scale prediction (referred to as *Scale* in Table 3), the use of a single FC layer instead of two FC layers (*Single FC*), and non-application of weight decay to the last FC layer (*w/o WD*). To see their impacts on performance, we conducted an ablation study, in which WRN-28-10 is used for the base network and TinyImageNet (resized) is chosen for an OOD dataset.

Table 3 shows the results. Row 1 shows the results of the baseline method [1], which are obtained in our experiments. Row 2 shows the results obtained by incorporating the scale prediction in the standard networks; to be specific, s predicted from $\boldsymbol{f}$ according to (4) is multipled with logits as $s \cdot (\boldsymbol{w}_i \boldsymbol{f} + b_i)$ $(i = 1, \ldots, C)$, which are then normalized by softmax to yield the cross-entropy loss. As is shown in Row 2, this simple modification to the baseline boosts the performance, which is surprising.

Row 3 and below show results when cosine similarity is used for OOD detection. Rows 3 to 6 show the results obtained when a fixed value (i.e., 16, 32, 64, 128) is chosen for s. It is observed from this that the application of scaling affects a lot detection performance, and it tends to be sensitive to their choice. This means that, if s is treated as a fixed parameter, it will become a hyperparameter that needs to be tuned for each dataset. Row 7 shows the result when the scale is predicted from $\boldsymbol{f}$ as in Row 1 but with cosine similarity. It is seen that this provides results comparable to the best case of manually chosen scales.

Row 8 shows the results obtained by further stopping application of weight decay to the last layer, which is the proposed method. It is seen that this achieves the best performance for both CIFAR-10 and CIFAR-100. Rows 9 and 10 show

Table 3. Ablation tests for evaluating the contribution of different components (i.e., 'Cosine', 'Single FC', 'Scale', and 'w/o WD'; see details from the main text) of the proposed method. AUROCs for detection of OOD samples (TinyImageNet (resized)) are shown.

	Cosine	Single FC	Scale	w/o WD	C-10	C-100
(1)			Baseline [1]		89.22	76.59
(2)	✗	✓	Pred	✗	95.74	88.70
(3)	✓	✓	16	✗	94.09	82.76
(4)	✓	✓	32	✗	96.53	89.02
(5)	✓	✓	64	✗	87.06	95.66
(6)	✓	✓	128	✗	62.02	94.82
(7)	✓	✓	Pred	✗	95.16	91.30
(8)	✓	✓	**Pred**	✓	**97.66**	**95.84**
(9)	✓	✗	Pred	✗	94.71	87.55
(10)	✓	✗	Pred	✓	89.90	86.96

the results obtained by the network having two FC layers in its final part, as in the recent metric learning methods. Following the studies of metric learning, we use 512 units in the intermediate layer. In this architecture, it is better to employ weight decay in the last layer as with the metric learning methods (i.e., Rows 9 vs 10). In conclusion, these results confirm that the use of cosine similarity as well as all the three components are indispensable to achieve the best performance.

5 Effectiveness of the Scaling Factor

5.1 Explanation by Hsu et al.

Our method and that of Hsu et al. [10] share the key component, the scaled cosine similarity, $s \cos \theta_i$, in which the angle θ_i with the i-th class centroid as well as the scale s are both *predicted* from the input x. In [10], not s but its inverse (i.e., $1/s$), denoted by $g(x)$, is predicted in a different way. The authors argue that $g(x)$ approximates $p(d_{in}|x)$, the probability of the input x being in-distribution. They then argue that this contributes to better OOD detection performance. However, this explanation contradicts with empirical observation, and therefore it must be wrong. Figure 3 (the upper row) shows the histograms of $g(x)$, which is computed according to the method of [10], for ID and OOD samples. Here, we use CIFAR-100 for the ID dataset and several others for OOD datasets; test samples are used for both. As is clearly seen, $g(x)$ is statistically not larger for ID samples than OOD samples, although its value should be consistently larger for ID than OOD samples if their argument is true. The lower row of Fig. 3 shows the histograms of the cosine similarity that is used for detecting OOD samples, showing its ability to distinguish ID and OOD samples. In short, $g(x)$ cannot be seen as an approximation of $p(d_{in}|x)$ and it alone cannot detect OOD samples

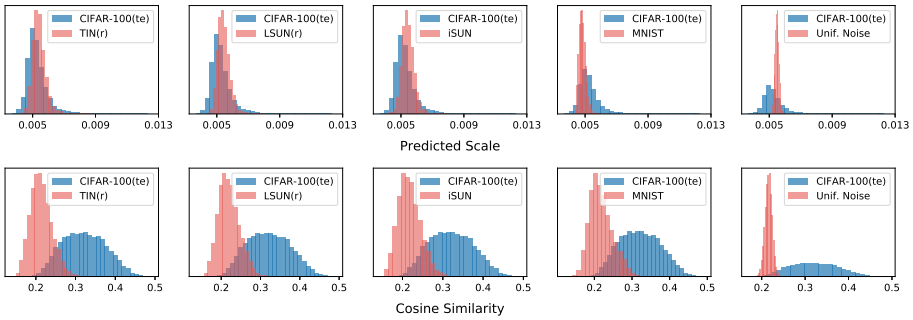

Fig. 3. Upper: Histograms of $g(x)$ of [10] for samples from ID (CIFAR-100) and different OOD datasets, respectively. Lower: Histograms of the cosine similarity for the same ID and OOD samples. The network is WRN-28-10 and dropout is not employed.

with good accuracy. The authors show that using dropout regularization induces different behavior of $g(x)$ between ID and OOD samples, but it is not employed in the main experiments evaluating OOD detection performance. Although it is not clear why the use of dropout makes $g(x)$ behave (slightly) differently, it should be concluded that the aforementioned claim on $g(x)$ is not the reason for the good performance of OOD detection.

5.2 Why Is Predicting s Essential?

Then, why is it essential to make the network predict the scale s. We remind the readers that we use $\cos\theta_i$ without s to detect OOD samples, which is the case as well with [10]. Thus, it is obviously associated not with prediction but with learning; that is, it contributes to better learning of feature space for OOD detection. An observation from our experiments is that s tends to be small at the initial training stage and becomes larger as the training goes, as shown in Fig. 4. This is reasonable since small s induces high entropy (i.e., softmax scores being more uniform and flattened) and large s induces low entropy; at the initial stage, there are a lot of misclassifications due to random weight initialization, leading to large cross-entropy loss, which will be compensated by making s small. More importantly, once the network has learned to correctly classify ID samples, or specifically, once it has learned to be able to consistently output max-logits for the correct classes, then s will start to become large; the minimization of the loss will be achieved not by reorganizing the feature space but by making s larger. We conjecture that this mechanism serves as a regularization to avoid overfitting the learned feature space to ID samples, while such overfitting occurs in the training of the standard networks. We believe that this leads to the difference in the OOD detection performance between the proposed cosine networks and standard networks.

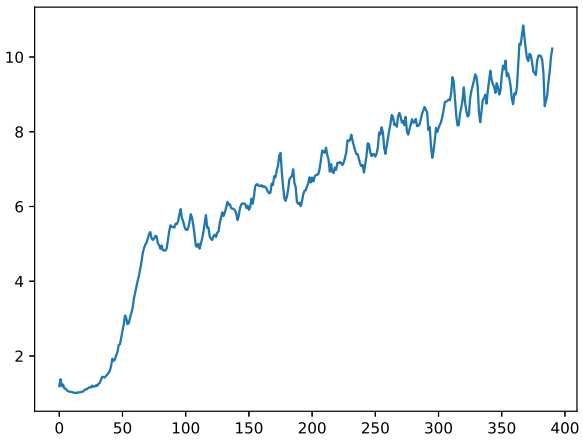

Fig. 4. Evolution of the scale s in first training epoch. The x-axis shows the training step.

6 Summary and Conclusions

In this paper, we have presented a novel method for OOD detection, and experimentally confirmed its superiority to existing approaches. We started our discussion with the observation that existing methods have hyperparameters specific to OOD detection, and their performance can be sensitive to their determination. The proposed method does not have such hyperparameters. It is based on the softmax of scaled cosine similarity and can be used with any networks by replacing their output layer. Training is performed by the standard method, i.e., minimizing a cross-entropy loss on the target classification task. Although a similar approach has already been employed in metric learning methods, the proposed method has several technical differences, which are essential to achieve high OOD detection performance, as was demonstrated in our ablation test. We have shown experimental comparisons between the proposed method and the existing methods using two different evaluation methods, i.e., the less-biased evaluation recently proposed in [9] and the conventional one-vs-one evaluation. In the former evaluation, which takes the above issue with hyperparameter determination into account, the proposed method shows clear superiority to others. Our method also shows at least comparable performance to them in the conventional evaluation. These results support the practicality of the proposed method in real-world applications. Lastly, we have briefly discussed why cosine similarity is effective for OOD detection.

Acknowledgements

This work was partly supported by JSPS KAKENHI Grant Number JP19H01110.

References

1. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. In: Proceedings of the International Conference on Learning Representations. (2017)
2. Liang, S., Li, Y., Srikant, R.: Enhancing the reliability of out-of-distribution image detection in neural networks. In: Proceedings of the International Conference on Learning Representations. (2017)
3. Lee, K., Lee, K., Lee, H., Shin, J.: A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In: Advances in Neural Information Processing Systems. (2018)
4. Vyas, A., Jammalamadaka, N., Zhu, X., Das, D., Kaul, B., Willke, T.L.: Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In: Proceedings of the European Conference on Computer Vision. (2018)
5. Shalev, G., Adi, Y., Keshet, J.: Out-of-distribution detection using multiple semantic label representations. In: Advances in Neural Information Processing Systems. (2018)
6. Malinin, A., Gales, M.: Predictive uncertainty estimation via prior networks. In: Advances in Neural Information Processing Systems. (2018)
7. Yu, Q., Aizawa, K.: Unsupervised out-of-distribution detection by maximum classifier discrepancy. In: Proceedings of the International Conference on Computer Vision. (2019)
8. Macêdo, D., Ludermir, T.: Neural networks out-of-distribution detection: Hyperparameter-free isotropic maximization loss, the principle of maximum entropy, cold training, and branched inferences. arXiv:2006.04005 (2020)
9. Shafaei, A., Schmidt, M., Little, J.J.: A less biased evaluation of out-of-distribution sample detectors. In: Proceedings of the British Machine Vision Conference. (2019)
10. Hsu, Y.C., Shen, Y., Jin, H., Kira, Z.: Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In: Proceedings of the Conference on Computer Vision and Pattern Recognition. (2020)
11. Ranjan, R., Castillo, C.D., Chellappa, R.: L2-constrained softmax loss for discriminative face verification. arXiv:1703.09507 (2017)
12. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: Deep hypersphere embedding for face recognition. In: Proceedings of the Conference on Computer Vision and Pattern Recognition. (2017)
13. Wang, F., Xiang, X., Cheng, J., Yuille, A.L.: Normface: L2 hypersphere embedding for face verification. In: Proceedings of the ACM International Conference on Multimedia. (2017)
14. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: Cosface: Large margin cosine loss for deep face recognition. In: Proceedings of the Conference on Computer Vision and Pattern Recognition. (2018)
15. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the Conference on Computer Vision and Pattern Recognition. (2019)
16. Zhang, X., Zhao, R., Qiao, Y., Wang, X., Li, H.: Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In: Proceedings of the Conference on Computer Vision and Pattern Recognition. (2019)
17. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: Proceedings of the International Conference on Machine Learning. (2016)

18. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Advances in Neural Information Processing Systems. (2017)
19. Azizpour, H., Teye, M., Smith, K.: Bayesian uncertainty estimation for batch normalized deep networks. In: Proceedings of the International Conference on Machine Learning. (2018)
20. Gast, J., Roth, S.: Lightweight probabilistic deep networks. In: Proceedings of the Conference on Computer Vision and Pattern Recognition. (2018)
21. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: Advances in Neural Information Processing Systems. (2017)
22. Leibig, C., Allken, V., Ayhan, M.S., Berens, P., Wahl, S.: Leveraging uncertainty information from deep neural networks for disease detection. Scientific reports **7** (2017) 17816
23. DeVries, T., Taylor, G.W.: Leveraging uncertainty estimates for predicting segmentation quality. arXiv:1807.00502 (2018)
24. Blum, H., Sarlin, P.E., Nieto, J., Siegwart, R., Cadena, C.: Fishyscapes: A benchmark for safe semantic segmentation in autonomous driving. In: Proceedings of the International Conference on Computer Vision Workshops. (2019)
25. Bevandić, P., Krešo, I., Oršić, M., Šegvić, S.: Simultaneous semantic segmentation and outlier detection in presence of domain shift. In: Proceedings of the German Conference on Pattern Recognition. (2019)
26. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Proceedings of the International Conference on Machine Learning. (2017)
27. Kuleshov, V., Fenner, N., Ermon, S.: Accurate uncertainties for deep learning using calibrated regression. In: Proceedings of the International Conference on Machine Learning. (2018)
28. Subramanya, A., Srinivas, S., Babu, R.V.: Confidence estimation in deep neural networks via density modelling. In: Proceedings of International Conference on Multimedia and Expo. (2017)
29. Scheirer, W.J., Rocha, A., Micheals, R.J., Boult, T.E.: Meta-recognition: The theory and practice of recognition score analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **33** (2011) 1689–1695
30. Chen, T., Navrátil, J., Iyengar, V., Shanmugam, K.: Confidence scoring using whitebox meta-models with linear classifier probes. In: Proceedings of the International Conference on Artificial Intelligence and Statistics. (2018)
31. Lee, K., Lee, H., Lee, K., Shin, J.: Training confidence-calibrated classifiers for detecting out-of-distribution samples. In: Proceedings of the International Conference on Learning Representations. (2018)
32. Ren, J., Liu, P.J., Fertig, E., Snoek, J., Poplin, R., DePristo, M.A., Dillon, J.V., Lakshminarayanan, B.: Likelihood ratios for out-of-distribution detection. In: Advances in Neural Information Processing Systems. (2019)
33. Hendrycks, D., Mazeika, M., Dietterich, T.G.: Deep anomaly detection with outlier exposure. In: Proceedings of the International Conference on Learning Representations. (2019)
34. Hendrycks, D., Lee, K., Mazeika, M.: Using pre-training can improve model robustness and uncertainty. In: Proceedings of the International Conference on Machine Learning. (2019)
35. Xu, P., Ehinger, K.A., Zhang, Y., Finkelstein, A., Kulkarni, S.R., Xiao, J.: Turkergaze: Crowdsourcing saliency with webcam based eye tracking. arXiv:1504.06755 (2015)

36. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: Proceedings of the International Conference on Learning Representations. (2015)
37. Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., Wu, Y.: Learning fine-grained image similarity with deep ranking. In: Proceedings of the Conference on Computer Vision and Pattern Recognition. (2014)
38. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the Conference on Computer Vision and Pattern Recognition. (2015)
39. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: Proceedings of the Conference on Computer Vision and Pattern Recognition. (2006)
40. Sun, Y., Wang, X., Tang, X.: Sparsifying neural network connections for face recognition. In: Proceedings of the Conference on Computer Vision and Pattern Recognition. (2016)
41. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: Advances in Neural Information Processing Systems. (2014)
42. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the International Conference on Machine Learning. (2015)
43. Liu, W., Wen, Y., Yu, Z., Yang, M.: Large-margin softmax loss for convolutional neural networks. In: Proceedings of the International Conference on Machine Learning. (2016)
44. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proceedings of the Conference on Computer Vision and Pattern Recognition. (2009)
45. Yu, F., Zhang, Y., Song, S., Seff, A., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv:1506.03365 (2015)
46. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: Advances in Neural Information Processing Systems Workshop on Deep Learning and Unsupervised Feature Learning. (2011)
47. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101 – mining discriminative components with random forests. In: Proceedings of The European Conference on Computer Vision. (2014)
48. Coates, A., Lee, H., Ng, A.Y.: An analysis of single layer networks in unsupervised feature learning. In: Proceedings of the International Conference on Artificial Intelligence and Statistics. (2011)
49. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE. (1998)
50. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747 (2017)
51. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: Proceedings of the British Machine Vision Conference. (2016)
52. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the Conference on Computer Vision and Pattern Recognition. (2017)