A Cosine Similarity-based Method for Out-of-Distribution Detection

Nguyen Ngoc-Hieu [1] Nguyen Hung-Quang [2] The-Anh Ta [1] Thanh Nguyen-Tang [3] Khoa D. Doan [2]
Hoang Thanh-Tung [1]

Abstract

The ability to detect OOD data is a crucial aspect of practical machine learning applications. In this work, we show that cosine similarity between the test feature and the typical ID feature is a good indicator of OOD data. We propose Class Typical Matching (CTM), a post hoc OOD detection algorithm that uses a cosine similarity scoring function. Extensive experiments on multiple benchmarks show that CTM outperforms existing post hoc OOD detection methods.

1. Introduction

In machine learning, distribution shift is the problem where the test distribution is not identical to the training distribution. Deep learning (DL) models, including those with good i.i.d. generalization performance, often perform poorly when distribution shifts occur (Nguyen et al., 2014). In real-world applications, distribution shifts are unavoidable because the environment changes in time. An emerging requirement for DL systems is that they must be able to handle distribution shifts (Amodei et al., 2016).

A common approach to the distribution shift problem is to detect out-of-distribution (OOD) samples, samples from the shifted distribution, and remove them from the test data. There is a wide array of OOD detection methods, ranging from classification-based, and density-based to distance-based methods (Yang et al., 2021). Classification-based methods, which classify incoming data as OOD or in-distribution (ID) based on the confidence or feature embedding assigned by a classifier, are some of the most commonly used methods. Several improvements to classification methods have been proposed, including modifying the loss function (Wei et al., 2022; Ming et al., 2023), changing the classifier architecture (Malinin & Gales, 2018), and using post hoc processing techniques (Hendrycks & Gimpel, 2017; Liang et al., 2017; Liu et al., 2020; Lee et al., 2018b). Among these techniques, post hoc methods are often preferred in practice due to their simplicity and ease of integration with pre-trained models without the need for additional training.

In this paper, we introduce Class Typical Matching (CTM), a post hoc algorithm for OOD detection. CTM is based on our observation that the cosine similarity between the test input's feature and the in-distribution features is very useful for OOD detection (Section 2.2). Different from other post hoc methods such as `Mahalanobis` (Lee et al., 2018a) and `KNN` (Sun et al., 2022) which leverage Euclidean distance in the feature space, our method uses cosine similarity for OOD score computation. In section 4, we theoretically show that cosine similarity is a good indicator for OOD samples. Our contributions are as follows:

1. We empirically and theoretically show that cosine similarity between the feature representation of a test input and a typical ID feature is an effective scoring function for OOD detection.

2. We propose CTM, a post hoc method that uses angular information for improved OOD detection.

3. We perform extensive experiments and ablation studies to evaluate the proposed method across 3 ID datasets and 10 OOD datasets.

2. Method

2.1. Prelimaries

Problem statement. We denote $\mathcal{X} \subseteq \mathbb{R}^d$ the input space and $\mathcal{Y} = \{y_1, \dots, y_C\}$ the label space. A classifier $f : \mathcal{X} \mapsto \mathbb{R}^C$ learns to map a given input $\mathbf{x} \in \mathcal{X}$ to the output space. Let $p_{\text{train}}(\mathbf{x}, y)$ denote a probability distribution defined on $\mathcal{X} \times \mathcal{Y}$. Further more, let $p_{\text{train}}(\mathbf{x})$ and $p_{\text{train}}(y)$ denote the marginal probability distribution on $\mathcal{X}$ and $\mathcal{Y}$, respectively. The goal is to design a binary function estimator $g : \mathcal{X} \to \{\text{in}, \text{out}\}$ that classifies whether a test example $\mathbf{x}$ is generated from $p_{\text{train}}(\mathbf{x})$ or not.

The concept of distribution shift is very diverse. It presents a

[1] FPT Software AI Center [2] VinUniversity [3] Johns Hopkins University. Correspondence to: Nguyen Ngoc-Hieu <ngochieutb13@gmail.com>.

Published at the ICML 2023 Workshop on Spurious Correlations, Invariance, and Stability. Baltimore, Maryland, USA. Copyright 2023 by the author(s).

challenge because being robust against one type of shift does not mean it will be effective against another shift. Therefore, it is important to characterize real-world shifts in order to develop effective methods for mitigating their impact. In this paper, we work with the image data and adopt an experimental setting presented in previous works where OOD samples are drawn from unknown classes (Fang et al., 2022; Yang et al., 2021).

One common approach to out-of-distribution (OOD) detection is to construct a scoring function $S : \mathcal{X} \mapsto \mathbb{R}$ that assigns lower scores to points drawn from an out-distribution $q(\mathbf{x})$. The detector, denoted as g, is then constructed based on the level set obtained from the score function

$$g(\mathbf{x}) = \begin{cases} \text{ID}, & \text{if } S(\mathbf{x}) \geq \lambda \\ \text{OOD}, & \text{if } S(\mathbf{x}) < \lambda \end{cases},$$

where $S(\mathbf{x})$ denotes a scoring function and λ is commonly set so that g correctly classifies a high proportion (e.g., 95%) of in-distribution (ID) data.

Notation. Posthoc methods often use a trained neural network to derive the score function. A trained deep NN classifier generally consists of two components: (1) a deep feature extractor that maps the input to a feature embedding and (2) a head that maps the embedding to an output. The most common choice for feature embedding is the output of the penultimate layer just before the classification layer. We denote the feature embedding map by $h : \mathcal{X} \mapsto \mathbb{R}^m$, where m is the dimension of the embedding. Given $\mathbf{x} \in \mathcal{X}$, denote $\mathbf{z} \in \mathbb{R}^m$ the feature of $\mathbf{x}$ i.e. $\mathbf{z} = h(\mathbf{x})$. The last FC layer in a neural network is given by:

$$f(\mathbf{x}) = Wh(\mathbf{x}) + \mathbf{b} = W\mathbf{z} + \mathbf{b} \tag{1}$$

where $W \in \mathbb{R}^{C\times m}$ is the weight matrix, and $\mathbf{b} \in \mathbb{R}^C$ is the bias vector. We also denote $\mathbf{w}_k$ the k-th row of W. For operators, we denote $\langle \cdot, \cdot \rangle$ the inner product between two vectors and $\| \cdot \|$ is the Euclidean norm.

2.2. Method

Hendrycks et al. (2022) show that the Maximum logit score is a strong baseline for large-scale OOD detection. This method scores each input by the largest values of their logits vector. Concretely, given the input's penultimate feature $\mathbf{z}$ the score is computed by the following equation:

$$\max_k \langle \mathbf{w}_k, \mathbf{z} \rangle + b_k$$

where $\mathbf{w}_k$ and b_k are weights and bias of the last layer w.r.t class k. This score function can also be formulated as

$$\max_k \|\mathbf{w}_k\| \|\mathbf{z}\| \cos(\mathbf{w}_k, \mathbf{z}) + b_k$$

where $\cos(\mathbf{w}_k, \mathbf{z})$ is the cosine similarity between $\mathbf{w}_k$ and $\mathbf{z}$. This formulation separates norm terms ($\|\mathbf{w}_k\|$, $\|\mathbf{z}\|$) and angular term $\cos(\mathbf{w}_k, \mathbf{z})$. Note that for a particular input, the model's prediction is $\arg\max_k \|\mathbf{w}_k\| \|\mathbf{z}\| \cos(\mathbf{w}_k, \mathbf{z}) + b_k$. As $\|\mathbf{z}\|$ is fixed for different k and the terms $\|\mathbf{w}_k\|$ and b_k are independent of the input, the cosine similarity term $\cos(\mathbf{w}_k, \mathbf{z})$ carries the most information for the model's prediction. When the $\cos(\mathbf{w}_k, \mathbf{z})$ values are similar for different k values, the score is influenced by the norms of $\mathbf{w}_k$ and b_k. This can make the OOD problem more challenging if the OOD sample is assigned to a class with larger weight norm $\|\mathbf{w}_k\|$ and bias b_k. We also observe that the norm of OOD feature embeddings can be large and increase the score. In fact, Sun et al. (2022) found that using the normalized penultimate feature greatly improves the `KNN` method, while Wei et al. (2022) suggests that the norm of *logit* is the source of the over-confident behavior of neural network trained with cross-entropy loss. `CIDER` (Ming et al., 2023) uses hypersphere representation to benefit OOD detection tasks by designing an end-to-end loss function. We argue that using only the term $\cos(\mathbf{w}_k, \mathbf{z})$ can retain the performance on the OOD detection task. Furthermore, our empirical finding suggests that replacing $\mathbf{w}_k$ with $\boldsymbol{\mu}_k$ - the mean of feature vectors in class k, and using $\cos(\boldsymbol{\mu}_k, \mathbf{z})$ improve OOD detection performance. Intuitively, this score can be thought of as computing the similarity between the input's feature and the typical feature of class k.

From the above motivation, we propose using cosine similarity with within-class feature mean $\boldsymbol{\mu}_k$ for OOD detection

$$g(\mathbf{x}) = \begin{cases} \text{ID}, & \text{if } \max\limits_k \cos(\boldsymbol{\mu}_k, \mathbf{z}) \geq \lambda \\ \text{OOD}, & \text{otherwise} \end{cases},$$

where λ is the threshold. The score function $S(\mathbf{x}) = \max_k \cos(\boldsymbol{\mu}_k, \mathbf{z})$ measures the similarity between the test input's feature and within-class mean features. In the next section, we show that this simple idea is, in fact, very effective for detecting OOD inputs.

3. Experiments

In this section, we present the experimental results of CTM on several benchmarks and an ablation study of the method.

3.1. Experimental Setup

Datasets and models. We conducted experiments on moderate and large-scale benchmarks. The moderate benchmarks include CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009) as in-distribution datasets and six out-of-distribution datasets: SVHN (Netzer et al., 2011), LSUN-Crop (Yu et al., 2015), LSUN-Resize (Yu et al., 2015), iSUN (Xu et al., 2015), Textures (Cimpoi et al., 2014), and Places365 (Zhou et al., 2017). The model used in the CIFAR benchmarks is

Table 1. **OOD detection results on ImageNet.** Proposed and baseline methods are based on a ResNet-50 (He et al., 2016) model trained on ImageNet-1k (Deng et al., 2009) only. ↓ indicates smaller values are better and ↑ indicates larger values are better.

	OOD Datasets									
Methods	**iNaturalist**		**SUN**		**Places**		**Textures**		**Average**	
	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
Softmax score (Hendrycks & Gimpel, 2017)	54.99	87.74	70.83	80.86	73.99	79.76	68.00	79.61	66.95	81.99
MaxLogit (Hendrycks et al., 2022)	50.78	91.15	60.42	86.44	66.07	84.03	54.93	86.39	58.05	87.00
ODIN (Liang et al., 2017)	47.66	89.66	60.15	84.59	67.89	81.78	50.23	85.62	56.48	85.41
Mahalanobis (Lee et al., 2018b)	97.00	52.65	98.50	42.41	98.40	41.79	55.80	85.01	87.43	55.47
Energy score (Liu et al., 2020)	55.72	89.95	59.26	**85.89**	64.92	**82.86**	53.72	85.99	58.41	86.17
KNN (Sun et al., 2022)	59.08	86.20	69.53	80.10	77.09	74.87	**11.56**	**97.18**	54.32	84.59
CTM (Our)	**22.58**	**95.51**	**55.02**	85.55	**63.07**	81.73	15.25	96.70	**38.98**	**89.87**

a pre-trained DenseNet-101 (Huang et al., 2017). The proposed method was also evaluated on a large-scale dataset, using ImageNet-1k (Deng et al., 2009) as the ID dataset and four OOD datasets: iNaturalist (Van Horn et al., 2018), SUN (Xiao et al., 2010), Place365 (Zhou et al., 2017), and Textures (Cimpoi et al., 2014). We used ResNet-50 (He et al., 2016) as the backbone for the ImangeNet benchmark. All networks were pre-trained using ID datasets without regularizing on auxiliary outlier data. The model parameters remained unchanged during the OOD detection phase, providing a fair comparison among the different methods.

Evaluation metrics. In our study, we evaluated the performance of OOD detection by measuring the following metrics: (1) the False Positive Rate (FPR95), which is the percentage of OOD images that were wrongly classified as ID images when the true positive rate of ID examples is 95%; (2) the Area Under the Receiver Operating Characteristic curve (AUROC), which assesses the overall performance of the OOD detection method; and (3) the Area Under the Precision-Recall curve (AUPR).

3.2. Results on both CIFAR and ImageNet benchmarks

We compared our method with other established post-hoc methods that do not require modifying the training process and have similar computational complexity and time requirements. Specifically, we selected `MSP` (Hendrycks & Gimpel, 2017), `MaxLogit` (Hendrycks et al., 2022), `Energy` (Liu et al., 2020), `ODIN` (Liang et al., 2017), `Mahalanobis` (Lee et al., 2018b), and `KNN` (Sun et al., 2022). The results of the CIFAR evaluations are presented in Table 2. We report the average performance over the six OOD datasets for 2 evaluation metrics: FPR95 and AUROC. On average CTM has 96.40% AUROC on CIFAR-10 and 89.11% AUROC on CIFAR-100, which is competitive with `KNN` method on CIFAR-10 and surpasses it on CIFAR-100 benchmark while algorithmically simpler. Detailed performance on individual datasets is reported in Appendix B.

Table 2. **OOD detection results on CIFAR benchmarks.** The results were averaged from 6 OOD datasets and measured in terms of FPR95 and AUROC. All values are percentages. All methods are based on a DenseNet-101 (Huang et al., 2017) model trained on ID data only.

	CIFAR-10		**CIFAR-100**	
Method	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑
Softmax score	48.73	92.46	80.13	74.36
MaxLogit	26.44	94.47	69.98	80.31
Energy score	26.55	94.57	68.45	81.19
ODIN	24.57	93.71	58.14	84.49
Mahalanobis	31.42	89.15	55.37	82.73
KNN	**16.61**	**96.71**	42.34	87.56
CTM (Ours)	18.23	96.40	**41.76**	**89.11**

Table 1 presents the performance of OOD detection methods on ImageNet. As we can see, CTM establishes favorable performance across OOD datasets and evaluation metrics. It reduces the FPR95 metrics by 15.43% compare to `KNN`.

3.3. Cosine similarity is informative.

We test the effectiveness of cosine similarity for OOD detection by making two modifications to the prediction process of an already trained network, (1) remove the bias b_k and normalize $\mathbf{w}_k$ and (2) normalize the penultimate features before feeding them to the linear layer. The prediction function after these modifications is given by the following equation:

$$\arg\max_k \frac{\exp\langle \hat{\mathbf{w}}_k, \hat{\mathbf{z}}\rangle}{\sum_c \exp\langle \hat{\mathbf{w}}_c, \hat{\mathbf{z}}\rangle} = \arg\max_k \frac{\exp\cos(\mathbf{w}_k, \mathbf{z})}{\sum_c \exp\cos\theta_c(\mathbf{z})}$$

where $\hat{\mathbf{w}}_k = \frac{\mathbf{w}_k}{\|\mathbf{w}_k\|}$ and $\hat{\mathbf{z}} = \frac{\mathbf{z}}{\|\mathbf{z}\|}$. We call this modification **CW** stands for cosine with weight. CW discards the magnitude component in $\mathbf{z}$ and the prediction is solely based on its direction. We also present another modification: cosine with mean (**CM**) which replaces $\mathbf{w}_k$ by the mean $\boldsymbol{\mu}_k$

of the training feature of class k. For this experiment we use two architecture: WideResNet-40-2 (Zagoruyko & Komodakis, 2016) and DenseNet-101 (Huang et al., 2017), and two datasets: CIFAR-10 and CIFAR-100. The result, reported in table 3, suggests that using only angular information $\cos(\mathbf{w}_k, \mathbf{z})$ can retain the most performance of the OOD detection task without much degradation on the classification task. Notice that CM increases the OOD detection performance by a large margin. These results verify our motivation in section 2.2 that cosine similarity is informative for both classifying ID examples and detecting OOD examples.

Table 3. **Cosine similarity is effective.** Test accuracy and OOD Detection performance (AUROC) of models before and after modification.

Model & Dataset	Test Accuracy	AUROC
	Standard/CW/CM	**Standard/CW/CM**
WideResNet-40-2 + CIFAR-10	94.84/94.82/**95.02**	91.29/**92.49**/**92.49**
WideResNet-40-2 + CIFAR-100	**75.95**/75.93/75.03	77.39/79.77/**86.95**
DenseNet + CIFAR-10	94.52/**94.55**/94.40	94.62/94.40/**96.40**
DenseNet + CIFAR-100	**75.08**/74.69/71.66	80.28/75.01/**89.11**

4. An analysis of cosine similarity from influence perspective

In this section, we analyze why using cosine similarity between the input feature and the mean is effective for out-of-distribution detection. We show that cosine similarity naturally arises from the influence perspective, which characterizes how a function's value at one input changes when we modify its value at another input. In particular, given a scalar output function g_W parameterized by W, Charpiat et al. (2019) proposes a kernel measuring the influence between two input $\mathbf{z}$ and $\mathbf{z}'$:

$$K_g(\mathbf{z}, \mathbf{z}') = \frac{\langle \nabla_W g_W(\mathbf{z}), \nabla_W g_W(\mathbf{z}') \rangle}{\|\nabla_W g_W(\mathbf{z})\| \|\nabla_W g_W(\mathbf{z}')\|}.$$

This kernel measures how similar output of g at $\mathbf{z}$ and $\mathbf{z}'$ change if the weight W is perturbed. Notice that the kernel is bounded between $[-1, 1]$. If the value $K_g(\mathbf{z}, \mathbf{z}')$ closes to 1 then $g_W(\mathbf{z})$ and $g_W(\mathbf{z}')$ response similar to each other for a perturbation on W. Intuitively, large $K_g(\mathbf{z}, \mathbf{z}')$ indicates that $\mathbf{z}$ and $\mathbf{z}'$ are similar under the point of view of g.

For the choice of g, inspired by Huang et al. (2021), we let g be the Kullback–Leibler (KL) divergence between the softmax output and a uniform distribution. Formally, denote $\mathbf{p} = \mathrm{softmax}(W\mathbf{z} + \mathbf{b})$ and $\mathbf{p}' = \mathrm{softmax}(W\mathbf{z}' + \mathbf{b})$ as predicted label probabilities assign to feature $\mathbf{z}$ and $\mathbf{z}'$ by the trained model. Then we have $g = D_{\mathrm{KL}}(\mathbf{u}||\mathbf{p})$, where D_{KL} is the KL divergence, and $\mathbf{u}$ is a uniform distribution on labels, i.e $\mathbf{u} = [1/C, 1/C, \ldots, 1/C] \in \mathbb{R}^C$. The gradient of $g(\mathbf{z}')$ w.r.t W points to direction which increase model's uncertainty at $\mathbf{z}'$. The kernel K_g now measure how much the predicted distribution at $\mathbf{z}$ become uniform if the weight W is perturbed such that increasing the uncertainty at $\mathbf{z}'$. Intuitively, if $\mathbf{z}'$ is a typical ID point then this perturbation will affect the model's prediction on an ID input more than on an OOD input. This also makes K_g agnostic to the true label of $\mathbf{z}$ or $\mathbf{z}'$.

Given the kernel K_g, for each test point $\mathbf{z}$ which is predicted with label k, we chose the point $\mathbf{z}'$ as the reference point such that it represents class k and measure the influence between $\mathbf{z}'$ and $\mathbf{z}$. To get the value $K_g(\mathbf{z}, \mathbf{z}')$, we first compute the gradient of g w.r.t. W. We have

$$\begin{aligned} \nabla_W g_W(\mathbf{z}) &= \frac{\partial D_{\mathrm{KL}}(\mathbf{u}||\mathbf{p}_W)}{\partial \operatorname{vec} W} \\ &= (\mathbf{u} - \mathbf{p})^\top \otimes \mathbf{z}^\top \in \mathbb{R}^{Cd}. \end{aligned}$$

where $\otimes$ is Kronecker product. Assume that $\mathbf{z}$ is not a zero vector and $\mathbf{p}$ is not uniform then $\|\nabla_W g_W(\mathbf{z})\| > 0$. Apply the same assumption for $\mathbf{z}'$ and $\mathbf{p}'$ then

$$\begin{aligned} K_g(\mathbf{z}', \mathbf{z}) &= \frac{\langle \nabla_W g_W(\mathbf{z}'), \nabla_W g_W(\mathbf{z}) \rangle}{\|\nabla_W g_W(\mathbf{z}')\| \|\nabla_W g_W(\mathbf{z})\|} \\ &= \frac{\langle \mathbf{u} - \mathbf{p}', \mathbf{u} - \mathbf{p} \rangle}{\|\mathbf{u} - \mathbf{p}'\| \|\mathbf{u} - \mathbf{p}\|} \cdot \frac{\langle \mathbf{z}', \mathbf{z} \rangle}{\|\mathbf{z}'\| \|\mathbf{z}\|} \\ &= \frac{\langle \mathbf{p}', \mathbf{p} \rangle - 1/C}{\|\mathbf{u} - \mathbf{p}'\| \|\mathbf{u} - \mathbf{p}\|} \cdot \frac{\langle \mathbf{z}', \mathbf{z} \rangle}{\|\mathbf{z}'\| \|\mathbf{z}\|} \end{aligned} \tag{2}$$

For $\mathbf{z}' = \boldsymbol{\mu}_k$, we observe that $\mathbf{p}'_W$ is approximately one-hot vector with $(\mathbf{p}'_W)_k = 1$. This observation is consistent with different architectures (DenseNet and ResNet) and training datasets (CIFAR-10, CIFAR-100, and ImageNet-1k). Substitute this one-hot vector $\mathbf{p}'$ to equation 2, we get

$$K_g(\boldsymbol{\mu}_k, \mathbf{z}) = \frac{p_k - 1/C}{(1 - 1/C)(\|\mathbf{p}\|^2 - 1/C)} \cdot \cos(\boldsymbol{\mu}_k, \mathbf{z})$$

Since $\mathbf{p}$ is not uniform and $p_k = \max_i(\mathbf{p}_W)_i$ we have $(p_k - 1/C) > 0$ and $(\|\mathbf{p}\|^2 - 1/C > 0)$. This suggests that $K_g(\boldsymbol{\mu}_k, \mathbf{z})$ and $\cos(\boldsymbol{\mu}_k, \mathbf{z})$ are positively correlated and smaller $\cos(\boldsymbol{\mu}_k, \mathbf{z})$ indicates less influence between the typical ID feature $\boldsymbol{\mu}_k$ and the test input's feature $\mathbf{z}$ and can be signal of a OOD input.

5. Conclusion

This paper introduces CTM, a post hoc approach to out-of-distribution detection based on the use of cosine similarity. The comprehensive experimental evaluation on 3 ID datasets, 10 OOD datasets, and using 3 performance metrics proves the efficacy of CTM. The importance of both feature embedding and an appropriate similarity measure for effective OOD detection was also confirmed through experiments. After theoretical analysis, we have demonstrated that cosine similarity is a suitable indicator for identifying OOD samples. We hope our work will encourage future research into using angular information for OOD detection.

References

Amodei, D., Olah, C., Steinhardt, J., Christiano, P. F., Schulman, J., and Mané, D. Concrete problems in AI safety. *CoRR*, abs/1606.06565, 2016. URL http://arxiv.org/abs/1606.06565.

Charpiat, G., Girard, N., Felardos, L., and Tarabalka, Y. Input similarity from the neural network perspective. *Advances in Neural Information Processing Systems*, 32, 2019.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Fang, Z., Li, Y., Lu, J., Dong, J., Han, B., and Liu, F. Is out-of-distribution detection learnable? In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=sde_7ZzGXOE.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of International Conference on Learning Representations*, 2017.

Hendrycks, D., Basart, S., Mazeika, M., Zou, A., Kwon, J., Mostajabi, M., Steinhardt, J., and Song, D. Scaling out-of-distribution detection for real-world settings. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8759–8773. PMLR, 17–23 Jul 2022.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.

Huang, R., Geng, A., and Li, Y. On the importance of gradients for detecting distributional shifts in the wild. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=fmiwLdJCmLS.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. *Tech Report*, 2009.

Lee, K., Lee, K., Lee, H., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018a. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/abdeb6f575ac5c6676b747bca8d09cc2-Paper.pdf.

Lee, K., Lee, K., Lee, H., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018b.

Liang, S., Li, Y., and Srikant, R. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

Liu, W., Wang, X., Owens, J., and Li, Y. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Malinin, A. and Gales, M. Predictive uncertainty estimation via prior networks. In *Proceedings of the Advances in Neural Information Processing Systems*, pp. 7047–7058, 2018.

Ming, Y., Sun, Y., Dia, O., and Li, Y. How to exploit hyperspherical embeddings for out-of-distribution detection? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=aEFaE0W5pAd.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Nguyen, A. M., Yosinski, J., and Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *CoRR*, abs/1412.1897, 2014. URL http://arxiv.org/abs/1412.1897.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on* *computer vision and pattern recognition*, pp. 4510–4520, 2018.

Sun, Y., Ming, Y., Zhu, X., and Li, Y. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning (ICML)*, 2022.

Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., and Belongie, S. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8769–8778, 2018.

Wei, H., Xie, R., Cheng, H., Feng, L., An, B., and Li, Y. Mitigating neural network overconfidence with logit normalization. *Proceedings of the International Conference on Machine Learning*, 2022.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010.

Xu, P., Ehinger, K. A., Zhang, Y., Finkelstein, A., Kulkarni, S. R., and Xiao, J. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, 2015.

Yang, J., Zhou, K., Li, Y., and Liu, Z. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021.

Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., and Xiao, J. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2017.

A. Implimentation

Software and Hardware. All experiments are run with PyTorch and NVIDIA RTX3090 GPUs.

Number of evaluation. In each run, we select a subset from the OOD dataset such that its size is equal to the size of the ID dataset. We run 5 times for each combination of method, ID data, and OOD data and report the average result.

Note on CIFAR results

- For the reimplementation of KNN with DenseNet-101, we tried a grid search, where the number of neigboors k include $\{10, 20, 50, 100, 200, 400\}$. We report the best result out of all hyperparameter combinations, given by $k = 50$ for CIFAR-10 and $k = 200$ for CIFAR-100.

B. Detailed CIFAR results

Table 4 and 5 present the full results on 6 OOD datasets for CIFAR-10 and CIFAR-100 benchmarks respectively.

C. Additional results

We include the result using MobileNetV2 (Sandler et al., 2018) for the ImageNet experiment in table 6.

D. Feature layers

The main results shown in the paper are generated by using features from the penultimate layer. In table 7, we show if we use feature from different layers of the network (Illustrated in Figure 1).

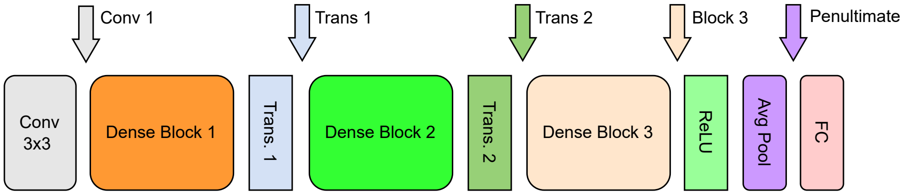

Figure 1. Diagram of DenseNet-101 architecture and indications of feature extraction layers.

Table 4. Detailed results on six common OOD benchmark datasets: `Textures` (Cimpoi et al., 2014), `SVHN` (Netzer et al., 2011), `Places365` (Zhou et al., 2017), `LSUN-Crop` (Yu et al., 2015), `LSUN-Resize` (Yu et al., 2015), and `iSUN` (Xu et al., 2015). For each ID dataset, we use the same DenseNet pretrained on **CIFAR-10**. ↑ indicates larger values are better and ↓ indicates smaller values are better.

Method	**SVHN**		**LSUN-c**		**LSUN-r**		**iSUN**		**Textures**		**Places365**		**Average**	
	FPR95 ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑
Softmax score	47.24	93.48	33.57	95.54	42.10	94.51	42.31	94.52	64.15	88.15	63.02	88.57	48.73	92.46
ODIN	25.29	94.57	4.70	98.86	3.09	99.02	3.98	98.90	57.50	82.38	52.85	88.55	24.57	93.71
Mahalanobis	6.42	98.31	56.55	86.96	9.14	97.09	9.78	97.25	21.51	92.15	85.14	63.15	31.42	89.15
Energy score	40.61	93.99	3.81	99.15	9.28	98.12	10.07	98.07	56.12	86.43	39.40	91.64	26.55	94.57
KNN	4.14	99.22	6.97	98.74	11.26	97.88	11.50	98.03	20.48	96.13	45.28	90.25	16.61	96.71
CTM	5.14	99.08	10.94	98.11	9.71	98.30	10.41	98.11	17.62	96.70	55.56	88.11	18.23	96.40

Table 5. Detailed results on six common OOD benchmark datasets: `Textures` (Cimpoi et al., 2014), `SVHN` (Netzer et al., 2011), `Places365` (Zhou et al., 2017), `LSUN-Crop` (Yu et al., 2015), `LSUN-Resize` (Yu et al., 2015), and `iSUN` (Xu et al., 2015). For each ID dataset, we use the same DenseNet pretrained on **CIFAR-100**. ↑ indicates larger values are better and ↓ indicates smaller values are better.

Method	**SVHN**		**LSUN-c**		**LSUN-r**		**iSUN**		**Textures**		**Places365**		**Average**	
	FPR95 ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑
Softmax score	81.70	75.40	60.49	85.60	85.24	69.18	85.99	70.17	84.79	71.48	82.55	74.31	80.13	74.36
ODIN	41.35	92.65	10.54	97.93	65.22	84.22	67.05	83.84	82.34	71.48	82.32	76.84	58.14	84.49
Mahalanobis	22.44	95.67	68.90	86.30	23.07	94.20	31.38	93.21	62.39	79.39	92.66	61.39	55.37	82.73
Energy score	87.46	81.85	14.72	97.43	70.65	80.14	74.54	78.95	84.15	71.03	79.20	77.72	68.45	81.19
KNN	17.93	96.35	31.44	92.85	47.31	90.41	39.70	91.90	24.27	93.70	93.41	60.13	42.34	87.56
CTM	10.03	97.87	31.90	93.16	54.41	88.11	45.43	90.28	20.43	95.44	88.38	69.82	41.76	89.11

Table 6. **OOD detection results on ImageNet.** Proposed and baseline methods are based on a MobilenetV2 model trained on ImageNet-1k (Deng et al., 2009) only. ↓ indicates smaller values are better and ↑ indicates larger values are better.

	OOD Datasets									
Methods	**iNaturalist**		**SUN**		**Places**		**Textures**		**Average**	
	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
Softmax score	64.29	85.32	77.02	77.10	79.23	76.27	73.51	77.30	73.51	79.00
ODIN	55.39	87.62	54.07	85.88	57.36	84.71	49.96	85.03	54.20	85.81
Mahalanobis	62.11	81.00	47.82	86.33	52.09	83.63	92.38	33.06	63.60	71.01
Energy score	59.50	88.91	62.65	84.50	69.37	81.19	58.05	85.03	62.39	84.91
CTM (Our)	46.66	90.41	71.28	77.55	78.86	73.03	14.49	96.60	52.82	84.39

Table 7. **OOD detection results on CIFAR benchmarks.** The results were averaged from 6 OOD datasets and measured in terms of FPR95 and AUROC. All values are percentages. All methods are based on a DenseNet-101 (Huang et al., 2017) model trained on ID data only.

	CIFAR-10		**CIFAR-100**	
Method	**FPR95** ↓	**AUROC** ↑	**FPR95** ↓	**AUROC** ↑
CTM before dense blocks	95.07	53.59	95.20	54.79
CTM after 1st Transition Block	66.10	81.73	68.21	76.15
CTM after 2nd Transition Block	25.91	92.24	62.87	78.09
CTM after 3rd Dense Block	16.29	96.25	58.69	80.35
CTM after penultimate Layer	18.23	96.40	41.76	89.11