Out-of-Distribution Detection Using an Ensemble of Self Supervised Leave-out Classifiers

Apoorv Vyas[13⋆], Nataraj Jammalamadaka[1⋆], Xia Zhu[2⋆], Dipankar Das[1], Bharat Kaul[1], and Theodore L. Willke[2]

[1] Intel labs, Bangalore, India
apoorv.vyas@idiap.ch natraj.j@gmail.com {dipankar.das,bharat.kaul}@intel.com
[2] Intel labs, Hillsboro, OR 97124, USA
{xia.zhu,ted.willke}@intel.com
[3] Idiap Research Institute, Switzerland

Abstract. As deep learning methods form a critical part in commercially important applications such as autonomous driving and medical diagnostics, it is important to reliably detect out-of-distribution (OOD) inputs while employing these algorithms. In this work, we propose an OOD detection algorithm which comprises of an ensemble of classifiers. We train each classifier in a self-supervised manner by leaving out a random subset of training data as OOD data and the rest as in-distribution (ID) data. We propose a novel margin-based loss over the softmax output which seeks to maintain at least a margin m between the average entropy of the OOD and in-distribution samples. In conjunction with the standard cross-entropy loss, we minimize the novel loss to train an ensemble of classifiers. We also propose a novel method to combine the outputs of the ensemble of classifiers to obtain OOD detection score and class prediction. Overall, our method convincingly outperforms Hendrycks *et al.* [7] and the current state-of-the-art ODIN [13] on several OOD detection benchmarks.

Keywords: anomaly detection · out-of-distribution

1 Introduction

Deep learning has significantly improved the performance of machine learning systems in fields such as computer vision, natural language processing, and speech. In turn, these algorithms are integral in commercial applications such as autonomous driving, medical diagnosis, and web search. In these applications, it is critical to detect sensor failures, unusual environments, novel biological phenomena, and cyber attacks. To accomplish this, systems must be capable of detecting when inputs are anomalous or out-of-distribution (OOD). In this work, we propose an out-of-distribution detection method for deep neural networks and demonstrate its performance across several out-of-distribution classification tasks on the state-of-the-art deep neural networks such as DenseNet[8] and Wide ResNet(WRN)[22].

[⋆] Equal contribution. Work done when the authors were working at Intel labs.

We propose a novel margin-based loss term, added to cross-entropy loss over in-distribution samples, which maintains a margin of at least m between the average entropy of OOD and ID samples respectively. We propose an ensemble of K leave-out classifiers for OOD detection. The training dataset with N classes is partitioned into K subsets such that the classes of each partition are mutually exclusive with respect to each other. Each classifier samples one of the K subsets without replacement as *out-of-distribution training data* and the rest of the $K-1$ subsets as *in-distribution training data*. We also propose a new OOD detection score which combines both softmax prediction score and entropy with temperature scaling [13]. We demonstrate the efficacy of our method on standard benchmarks proposed in ODIN [13] and *outperform them*. Our contributions are (i) proposing a novel loss for OOD detection, (ii) demonstrating a self-supervised OOD detection method, and (iii) moving the state-of-the-art by outperforming the current best methods.

The rest of the paper is organized as follows. Section 2 describes the previous work on the OOD detection. Section 3 describes our method in detail. Section 4 describes various evaluation metrics to measure the performance of OOD detection algorithms. The ablation results of various design choices and hyper-parameters are also presented. We then compare our method against the recently proposed ODIN algorithm [13] and demonstrate that it outperforms it on various OOD detection benchmarks. Finally, section 5 discusses observations about our method, future directions and conclusions.

2 Related Work

Traditionally, based on the availability of the data labels, OOD detection methods can be categorized into supervised [16], semi-supervised [4] and unsupervised methods [15], [3]. All these classes of methods have access to the OOD data while training but differ in access to labels. It is assumed that the classifier has labels for normal as well as OOD classes during training for supervised OOD detection, while labels for only the normal classes are available in case of semi-supervised methods, and no labels are provided for unsupervised OOD detection methods which typically rely on the fact that anomalies occur in much less frequency than normal data. Our method is able to detect anomalies in test OOD datasets the very first time it encounters them during testing. We use one OOD dataset as validation set to search for hyper-parameters.

Notable OOD detection algorithms which work in the same setting as ours are isolation forests [14], Hendrycks and Gimpel [7], ODIN [13] and Lee *et.al,* [12]. The work reported in isolation forests [14] exploits the fact that anomalies are scarce and different and while constructing the isolation tree, it is observed that the anomalous samples appear close to the root of the tree. These anomalies are then identified by measuring the length of the path from the root to a terminating node; the closer a node is to the root, the higher is its chance of representing an OOD. Hendrycks and Gimpel [7] is based on the observation that prediction probability of incorrect and out-of-distribution samples tends to be lower than the prediction probability of correct samples. Lee *et al.* modify the formulation of generative adversarial networks [6] to generate OOD samples for the given in-distribution. They achieve this by simultaneously training GAN [6] and standard supervised neural network. The joint loss consists of individual losses and an additional connecting term which reduces the KL divergence between the generated sample's softmax distribution and the uniform distribution.

Another set of related works are open set classification methods [17], [18], [19], [1], [2]. Scheirer *et.al,* [19] introduces and formalizes "open space risk" which intuitively is the risk associated with labeling those areas in the output feature space as positive where there is no density support from the training data. Thus the approximation to the ideal risk is defined as a linear combination of "open space risk" and the standard "empirical risk". Bendale and Boult [1] extend the definition of "open set risk" to open world recognition where the unknown samples are not static set. The open world recognition defines a *multi-class open set recognition function*, a *labeling process* and an *incremental learning function*. The *multi-class open set recognition function* detects novel classes which are labeled using the *labeling process* and finally are fed to *incremental learning function* which updates the model. The OSDN [2] work proposes openMax function which extends the softmax function by adding an additional unknown class to the classification layer. The value for unknown class is computed by taking the weighted average of all other classes. The weights are obtained from Weibull distribution learnt over the pairwise distances between penultimate activation vectors (AV) of the top farthest correctly classified samples. For an OOD test sample these weights will be high while for an in-distribution sample these scores will be low. The final activation vector is re-normalized using softmax function.

The current state-of-the-art is ODIN [13] which proposes to increase the difference between the maximum softmax scores of in-distribution and OOD samples by (i) calibrating the softmax scores by scaling the logits that feed into softmax by a large constant (referred to as temperature) and (ii) pre-processing the input by perturbing it with the loss gradient. ODIN [13] demonstrated that at high temperature values, the softmax score for the predicted class is proportional to the relative difference between largest unnormalized output (logit) and the remaining outputs (logits). Moreover, they empirically showed that the difference between the largest logit and the remaining logits is higher for the in-distribution images than for the out-of-distribution images. Thus temperature scaling pushes the softmax scores of in- and out-of-distribution images further apart when compared to plain softmax. Perturbing the input image through gradient ascent w.r.t to the score of predicted label was demonstrated [13] to have stronger effect on the in- distribution images than that on out-of-distribution images, thereby, further pushing apart the softmax scores of in- and out-of-distribution images. We leverage the effectiveness of both these methods. The proposed method outperforms all the above methods by considerable margins.

3 Out-of-Distribution (OOD) Classifier

In this section, we introduce three important components of our method: entropy based margin-loss function (3.1), training ensemble of leave-out classifiers (3.2), and OOD detection scores (3.3).

Algorithm 1: Algorithm to train K Leave Out Classifiers

```
Input  : Training Data $X$, Number of classes $N$, $K$ Partitions, $\delta$ accuracy
         bound, Validation OOD Data $X_{valOOD}$
Output : $K$ Leave Out Classifiers
for $i \leftarrow 1$ to $K$ do
    $X_{ood} \leftarrow X_i$, $X_{in} \leftarrow X - X_i$;
    while Not Converged do
        ood_batch $\leftarrow$ Sample OOD minibatch;
        in_batch $\leftarrow$ Sample in-distribution minibatch;
        update the classifier $F_i$ by minimizing loss (Equation 1) using SGD;
        save model with least OOD error on $X_{valOOD}$ within $\delta$ accuracy of
        current best accuracy.;
    end
end
return $\{F_i\}$;
```

3.1 Entropy based Margin-Loss

Given a labeled set $(x_i \in X_{in}, y_i \in Y_{in})$ of in-distribution (ID) samples and $(x_o \in X_{ood})$ of out-of-distribution (OOD) samples, we propose a novel loss term in addition to the standard cross-entropy loss on ID samples. This loss term seeks to maintain a margin of at least m between the average entropy of OOD and ID samples. Formally, a multi-layer neural network $F : x \rightarrow p$ which maps an input x to probability over classes and parametrized by W is learned by minimizing the margin-loss over the difference of average entropies over OOD samples and ID samples, and cross entropy loss on ID samples. The loss function is given by Equation 1,

$$\mathcal{L} = -\frac{1}{|X_{in}|}\sum_{x_i \in X_{in}} \log(F_{y_i}(x_i)) + \beta * \max\left(m + \frac{\sum\limits_{x_i \in X_{in}} H(F(x_i))}{|X_{in}|} - \frac{\sum\limits_{x_o \in X_{ood}} H(F(x_o))}{|X_{ood}|}, 0\right) \quad (1)$$

where $F_{y_i}(x_i)$ is the predicted probability of sample x_i whose ground truth class y_i, $H(\cdot)$ is the entropy over the softmax distribution, m is the margin and β is the weight on margin entropy loss.

The new loss term evaluates to its minimum value *zero* when the difference between the average entropy of OOD and ID samples is greater than the margin m. For ID samples, the entropy loss encourages the softmax probabilities of non ground-truth classes to decrease and the cross-entropy loss encourages the softmax probability of ground-truth class to increase. For OOD samples, the entropy loss encourages the probabilities of all the classes to be equal. When the OOD entropy is higher than ID entropy by a margin m, the new loss term evaluates to *zero*. Our experiments suggest that maximizing OOD entropy leads to overfitting. Bounding the difference of average entropies of ID samples and OOD samples entropies with margin has helped in preventing overfitting, and thus is better for model generalization[11].

Algorithm 2: Algorithm for OOD Detection using K Leave Out Classifiers

```
Input  : Test Image $x_t$, $K$ leave-out Classifiers $F_i, i \in 1, ..., K$ and their
         temperature scaled versions $F_i(x_t; T)$, perturbation factor $\epsilon$, number of
         classes $N$
Output : Class prediction $C_t$, OOD score $O_t$
$S_t \leftarrow \{0\}^N$, $O_t \leftarrow 0$;
for $i \leftarrow 1$ to $K$ do
    $S_t \leftarrow S_t + F_i(x_t)$;
    $\hat{x}_t \leftarrow x_t - \epsilon * \text{sign}(\frac{\partial H(F_i(\hat{x}_t;T))}{\partial x_t})$;
    $O_t \leftarrow O_t + \max_N(F_i(\hat{x}_t;T)) - H(F_i(\hat{x}_t;T))$;
end
$C_t \leftarrow \text{argmax}(S_t)$;
return $C_t$, $O_t$;
```

3.2 Training Ensemble of leave-out classifiers

Given an in-distribution training data X which consists of N classes, the data is divided into K partitions $X_i, i \in \{1, ..., K\}$ such that the classes of each partition are mutually exclusive to all other partitions. A set of K classifiers are learned where classifier $F_i, i \in \{1, ..., K\}$ uses the partition X_i as OOD data X_{ood} and rest of the data $X - X_i$ as in-distribution data X_{in}. A particularly simple way of partitioning the classes is to divide them into partitions with equal number of classes. For example, dividing a dataset of $N = 100$ classes into $K = 5$ random and equal partitions gives us a partition with size of 20 classes. Each of the K classifiers would then use 20 classes for OOD and 80 classes as ID. Each classifier F_i is learned by minimizing the proposed margin entropy loss (eqn 1) using the assigned OOD and ID data. During the training, we also assume a small number of out-of-distribution images to be available as a validation dataset. At every epoch, we save the model with best OOD detection rate on this small OOD validation data and within a δ accuracy bound of the current best accuracy. The complete algorithm for training the leave-out classifiers in presented in Algorithm 1.

3.3 OOD Detection Score for Test Image

At the time of testing, an input image is forward propagated through all the K leave-out classifiers and the softmax vectors of all the networks are remapped to their original class indices. For the left-out classes, a score of zero is assigned. For classification of an input sample, first the softmax vectors of all the classifiers are averaged and the class with the highest averaged softmax score is considered as the prediction. For the OOD detection score, for each of the K classifiers, we first compute both the maximum value and negative entropy of the softmax vectors with temperature scaling. We then compute the average of all these values to obtain the OOD detection score.

An in-distribution sample with class labels y_i acts as an OOD for exactly one of the K classifiers. This is because the classes are divided into K mutually exclusive partitions and class y_i can be part of only one of these partitions. When an in-distribution sample (x_i, y_i) is forward propagated through these K classifiers, we would expect the negative entropy and maximum softmax score to be high for $K-1$ classifiers where it was sampled as in-distribution dataset. However, for an OOD sample x_o we expect the negative entropy and maximum softmax score to be relatively low for all the K classifiers. We thus expect a higher OOD detection score for ID samples than the OOD samples thus differentiating them.

Following the work of ODIN [13], we use both temperature scaling and input preprocessing while testing. In temperature scaling, the logits feeding into softmax layer are scaled by a constant factor T. It has been established that temperature scaling can calibrate the classification score and in the context of OOD detection [13], it pushes the softmax scores of in- and out-of-distribution samples further apart when compared to plain softmax. We modify input preprocessing by perturbing over entropy loss instead of cross-entropy loss used by ODIN [13]. Perturbing using the entropy loss decreases the entropy of the ID samples much more than the OOD samples. For an input test image x_t, after it is forward propagated through the neural network F_i, the gradient of the entropy loss with respect to x_t is computed and the input is perturbed with Equation 2. The OOD detection score is then calculated by the combination of maximum softmax value and entropy as described previously, both with temperature scaling.

$$\hat{x_t} = x_t - \epsilon * \mathrm{sign}(\frac{\partial L(F_i(x_t;T))}{\partial x_t}) \tag{2}$$

The complete algorithm for OOD detection on a test image is presented in Algorithm 2.

4 Experimental Results

In this section, we describe our experimental results. The details such as in-distribution and OOD datasets, the neural network architectures and evaluation metrics are described in detail. The ablation studies on various hyper-parameters

Architecture	**CIFAR-10**	**CIFAR-100**
DenseNet-BC	5.0	19.9
WRN-28-10	5.0	20.4

Table 1: Test error rates on CIFAR-10 and CIFAR-100

of the algorithms are described and conclusions are drawn. Finally, our method is compared against the current state-of-the-art ODIN [13] and is shown to significantly outperform it.

4.1 Experimental Setup

We use CIFAR-10 (contains 10 classes) and CIFAR-100 (contains 100 classes) [10] datasets as in-distribution datasets to train deep neural networks for image classification. They both consist of 50,000 images for training, and 10,000 images for testing. The dimensions of an image in both the datasets is 32×32. The classes of both CIFAR-10 and CIFAR-100 are randomly divided into five parts. As described in Section 3.2, each part is assigned as OOD to a unique network which is then trained. For each network, the other parts act as in-distribution samples.

Following the benchmarks given in [13], the following OOD datasets are used in our experiments. The datasets are described in ODIN[13] and provided as a part of their code release; here we are simply restating the description for comprehensiveness.

- **TinyImageNet[9] (TIN)** is a subset of ImageNet dataset[13]. Tiny ImageNet contains 200 classes which is drawn from original 1,000 classes of ImageNet. In total, there are 10,000 images in the Tiny ImageNet. By randomly cropping and downsampling each image to 32×32, two datasets *TinyImageNetcrop* (*TINc*) and *TinyImageNetresize* (*TINr*) are constructed.
- **LSUN** is the Large Scale UNderstanding dataset (LSUN)[21] created by Princeton, using deep learning classifiers with humans in the loop. It contains 10,000 images of 10 scene categories. By randomly cropping and downsampling each image to size 32×32, two datasets *LSUNc* and *LSUNr* are constructed.
- **iSUN[20]** is collected by gaze tracking from Amazon Mechanical Turk using a webcam. It contains 8925 scene images. Similar to the above dataset as other datasets, images are down-sampled to size 32×32.
- **Uniform Noise (UNFM)** is synthetic dataset consists of 10,000 noise images. The RGB value of each pixel in an image is drawn from uniform distribution in the range $[0, 1]$.
- **Gaussian Noise (GSSN)** is synthetic dataset consists of 10,000 noise images. The RGB value of each pixel is drawn from independent and identically distributed Gaussian with mean 0.5 and unit variance and each pixel value is clipped to the range $[0, 1]$.

Neural network architecture Following ODIN[13], two state-of-the-art neural network architectures, *DenseNet* [8] and *Wide ResNet*(WRN) [22], are adopted to evaluate our method. For DenseNet, we use the DenseNet-BC setup as in [8], with depth $L = 100$, growth rate $k = 12$ and dropout rate 0. For Wide ResNet, we use WRN-28-10 setup, with depth 28, width 10 and dropout rate of 0.3. We train both DenseNet-BC and Wide ResNet on CIFAR-10 and CIFAR-100 for 100 epochs with batch size 100, momentum 0.9, weight decay 0.0005, and margin 0.4. The initial training rate is 0.1 and it is linearly dropped to 0.0001 over the whole training process. During training, we augment our training data with random flip and random cropping. We use the smallest OOD dataset, iSUN, as validation data for hyper-parameter search. We test the rest four out-of-distribution datasets except iSUN on our trained network. During testing, we use batch size 100. Similar to ODIN[13], input preprocessing with $\epsilon = 0.002$ is used.

Table 1 shows the test error rates when our method is trained and tested on CIFAR-10 and CIFAR-100 respectively using the algorithms 1 and 2. For CIFAR-10, the vanilla DenseNet-BC [8] and the proposed method gives error rates of 4.51% and 5.0% respectively. For CIFAR-100, the error rates are 22.27% and 19.9% respectively. On both these datasets, the difference in error rates is marginal. For WRN [22] with depth 40, $k = 10$, the test error rate on CIFAR-10 for the vanilla network is 4.17% and for the proposed network is 5.0%. For CIFAR-100, the error rates are 20.5% and 20.4% for the vanilla network and the proposed network respectively. The small difference in the performance on CIFAR-10 can be explained by the fact that the our method did not use the ZCA whitening preprocessing while the vanilla network did.

4.2 Evaluation Metrics

To measure the effectiveness of our method to distinguish between in-distribution and out-of-distribution samples, we adopt five different metrics, same as what was used in ODIN[13] paper. We restate these metrics below for comprehensiveness. In the rest of manuscript, TP, TN, FP, FN are used to denote true positives, true negatives, false positives and false negatives respectively.

FPR at 95% TPR measures the probability that an out-of-distribution sample is misclassified as in-distribution when the true positive rate (TPR) is 95%. In this metric, TPR is computed by $TP/(TP + TN)$, and FPR is computed by $FP/(FP + TN)$.

Detection Error measures the minimum misclassification probability over all possible score thresholds, as defined in ODIN[13]. To have a fair comparison with ODIN, the same number of positive and negative samples are used during testing.

AUROC is the Area Under the Receiver Operating Characteristic curve. In a ROC curve, the TPR is plotted as a function of FPR for different threshold settings. AUROC equals to the probability that a classifier will rank a randomly chosen positive sample higher than a randomly chosen negative one. AUROC score of 100% means perfect separation between positive and negative samples.

Ablation Studies	Parameters	FPR at 95% TPR ↓	Detection Error ↓	AUROC ↑	AUPR In ↑	AUPR Out ↑	CLS Acc ↑
Number of Splits	3	32.37	13.94	93.50	94.39	92.22	76.41
	5	**22.95**	**10.79**	**95.69**	**96.55**	94.3	80.01
	10	28.71	12.53	94.48	95.37	93.26	81.94
	20	23.85	10.95	95.49	96.24	**94.36**	**82.33**
Type of splits	Random	**22.95**	**10.79**	**95.69**	**96.55**	**94.3**	**80.01**
	Manual	40.16	16.26	91.57	92.90	89.57	79.79
Epsilon	0.000000	53.51	16.37	90.75	93.16	86.71	**80.32**
	0.000313	41.62	14.37	92.8	94.52	90.16	80.22
	0.000625	34.64	12.83	94.09	95.4	92.19	80.17
	0.001250	25.74	11.19	95.38	96.31	94.04	80.08
	0.002000	**22.95**	**10.79**	**95.69**	**96.55**	**94.3**	80.01
	0.003000	29.07	11.79	94.73	95.9	92.43	79.97
Temp-rature	1	38.57	17.32	91.44	92.7	90.12	**80.01**
	10	27.84	11.93	94.86	95.81	93.39	**80.01**
	100	24.44	10.86	95.6	96.5	94.17	**80.01**
	1000	22.95	**10.79**	**95.69**	**96.55**	**94.3**	**80.01**
	5000	**22.7**	10.81	95.66	96.53	94.28	**80.01**
Loss Function	SFX	84.09	36.55	68.96	72.38	63.77	54.18
	SFX+MaxEntropyDiff	50.70	19.65	88.26	89.71	86.18	72.99
	SFX+MarginEntropy	**22.95**	**10.79**	**95.69**	**96.55**	**94.3**	**80.01**
OOD Detection Score	SFX	50.52	19.91	88.69	90.91	86.19	**80.01**
	Entropy	36.23	16.48	91.92	93.03	90.74	**80.01**
	SFX+Entropy	38.57	17.32	91.44	92.7	90.12	**80.01**
	SFX@Temp	**22.71**	10.83	95.65	96.52	94.26	**80.01**
	Entropy@Temp	37.0	14.05	93.33	94.76	91.04	**80.01**
	(SFX+Entropy)@Temp	22.95	**10.79**	**95.69**	**96.55**	**94.3**	**80.01**

Table 2: Ablation Studies on CIFAR-100 as in-distribution data and iSUN as out-of-distribution data on DenseNet-100 network. All values are percentages. ↑ indicates larger value is better, and ↓ indicates lower value is better.

AUPR-In measures the Area Under the Precision-Recall curve. In a PR curve, the *precision*$= TP/(TP+FP)$, is plotted as a function of *recall*$= TP/(TP+FN)$, for different threshold settings. Since precision is directly influenced by class imbalance (due to FP), PR curves can highlight performance differences that are lost in ROC curves for imbalanced datasets[5]. AUPR score of 100% means perfect distinguish between positive and negative samples. For AUPR-In metric, in-distribution images are specified as positive.

AUPR-Out is similar to the metric AUPR-In. The difference lies in that for AUPR-Out metric, out-of-distribution images are specified as positive.

CLS Acc is the classification accuracy for ID samples.

4.3 Ablation Studies

In this section, we perform ablation studies to study the effects of various hyper parameters used in our model. We perform the ablation studies on DenseNet-BC [8] network with CIFAR-100 [10] as in-distribution while training and iSUN [20] as the OOD validation data while testing. By default, we use 5 random splits for CIFAR-100, $\epsilon = 0.002$, *SFX+MarginEntropy* loss to train network, accuracy bound $\delta = 2\%$ to save the models, and use *(Softmax + Entropy)@Temperature* with $T = 1000$ to detect out-of-distribution samples. Results are given in Table 2.

(1) **Number of splits:** This analysis characterizes the sensitivity of our algorithm to the number of splits of the training classes which is same as the number of classifiers in the ensemble. As the number of splits increase, the number of times a particular training class being in-distribution for the leave-out classifiers increases too. This enables the ensemble to discriminate an in-distribution sample from the OOD sample. But it also increases the computational cost. For CIFAR-100, we studied 3, 5, 10 and 20 splits. Our results show that while 5 splits gave the best result, 3 splits also provides a good trade-off between accuracy and computational cost. We choose the number of splits as 5 as default value.

(2) Type of splits: This study characterizes the way in which the classes are split into mutually exclusive sets. We experiment with splitting the classes manually using prior knowledge and splitting randomly. For the manual split, the class labels are first clustered into semantically consistent groups and classes from each group are then distributed across the splits. The results show that the OOD detection rates for random selections are better than the manual selection.This ensures that we can achieve good OOD detection rates even by random selection of classes when the number of classes is huge.

(3) Different ϵ for input preprocessing: For input preprocessing, we sweep over $\epsilon \in [0, 0.000313, 0.000625, 0.00125, 0.002, 0.003]$. Our results show that as ϵ increases from 0, the performance of out-of-distribution detector increases, and it reaches the best performance at 0.002. The further increase of ϵ does not help performance.

(4) Different T for temperature scaling: For temperature scaling, we sweep over $T \in [1, 10, 100, 1000, 5000]$. Our results show that for DenseNet-BC with CIFAR-100, as T increases from 1 to 1000, the performance of out-of-distribution detector increases. Beyond $T = 1000$, the performance does not change much.

(5) Loss function variants: We study the effects of training our method with different types of losses. The training regime follows the strategy given in section 3.2, where the training data X is split into K partitions $X_i, i \in \{1, ..., K\}$. A total of K classifiers are trained where classifier F_i uses the partition X_i as OOD data and $X - X_i$ as in-distribution data.

- *SFX*: We assign an additional label to all the OOD samples and train classification network using the cross entropy loss.

- *SFX+MaxEntropyDiff*: Along with the cross entropy loss, we maximize the difference between the entropy of in- and out-of-distribution samples across all in-distribution classes.
- *SFX+MarginEntropy*: Along with the cross entropy, we maximize the difference between the entropy of in- and out-of-distribution samples across all in-distribution classes, but is bounded by a margin as given in Equation 1.

Our results show that the proposed *SFX+MarginEntropy* loss works dramatically better than all other types of losses for detecting out-of-distribution samples as well as for accurate classification. The results demonstrate that the proposed novel loss function *SFX+MarginEntropy* (equation 1) is the major factor in significant improvements over the current state-of-the-art ODIN [13].

(6) Out-of-distribution detector: We study different OOD scoring methods to discriminate out-of-distribution samples from in-distribution samples.

- *Softmax score*: Given an input image, the score is given by the average of maximum softmax outputs over all the classifiers in the ensemble.
- *Entropy score*: Given an input image, the score is given by the average of entropy of softmax vector over all the classifiers in the ensemble.
- *Softmax + Entropy*: Given an input image, both the above scores are added.
- *Softmax@Temperature*: Given an input image, the above described Softmax score is computed on temperature scaled ($T = 1000$) softmax vectors.
- *Entropy@Temperature*: Given an input image, the above described Entropy score is computed on temperature scaled ($T = 1000$) softmax vectors.
- *(Softmax + Entropy)@Temperature*: Given an input image, the above described Entropy@Temperature and Softmax@Temperature are computed ($T = 1000$) on softmax vectors and then added.

Among the above OOD scoring methods, the *(Softmax + Entropy)@Temperature* ($T = 1000$) achieved the best performance. *Softmax@Temperature* ($T = 1000$) achieved the second best performance.

4.4 Results and Analysis

Table 3 shows the comparison between our results and ODIN [13] on various benchmarks. The results are reported on all neural network, in-dataset and OOD dataset combinations. Our hyperparameters are tuned using iSUN dataset. From the Table 3, it is very clear that our approach significantly outperforms ODIN [13] across all neural network architectures on almost all of the dataset pairs. The combination of novel loss function, OOD scoring method, and the ensemble of models has enabled our method to significantly improve the performance of OOD detection on more challenging datasets, such as LSUN (resized), iSUN and ImageNet(resized), where the images contain full objects as opposed to the cropped parts of objects. The proposed method is slightly worse on the uniform and some of Gaussian distribution results. Moreover, our method achieves significant gains on both CIFAR-10 and CIFAR-100 with the same number of splits which is 5, even though the number of classes have increased by a factor of

	OOD Dataset	FPR at 95% TPR ↓	Detection Error ↓	AUROC ↑	AUPR In ↑	AUPR Out ↑
		each cell in **ODIN[13]/Our Method** format				
DenseNet-BC CIFAR-10	TINc	4.30/**1.23**	4.70/**2.63**	99.10/**99.65**	99.10/**99.68**	99.10/**99.64**
	TINr	7.50/**2.93**	6.10/**3.84**	98.50/**99.34**	98.60/**99.37**	98.50/**99.32**
	LSUNc	8.70/**3.42**	6.00/**4.12**	98.20/**99.25**	98.50/**99.29**	97.80/**99.24**
	LSUNr	3.80/**0.77**	4.40/**2.1**	99.20/**99.75**	99.30/**99.77**	99.20/**99.73**
	UNFM	**0.00**/2.61	**0.20**/3.6	**100**/98.55	**100**/98.94	**100**/97.52
	GSSN	**0.00**/**0.00**	**0.50**/0.2	**99.90**/99.84	**100**/99.89	**99.90**/99.6
DenseNet-BC CIFAR-100	TINc	17.30/**8.29**	8.80/**6.27**	97.10/**98.43**	97.40/**98.58**	96.80/**98.3**
	TINr	44.30/**20.52**	17.50/**9.98**	90.70/**96.27**	91.40/**96.66**	90.10/**95.82**
	LSUNc	17.60/**14.69**	9.40/**8.46**	96.80/**97.37**	97.10/**97.62**	96.50/**97.18**
	LSUNr	44.00/**16.23**	16.80/**8.77**	91.50/**97.03**	92.40/**97.37**	90.60/**96.6**
	UNFM	**0.50**/79.73	**2.50**/9.46	**99.50**/92.0	**99.60**/94.77	**99.00**/83.81
	GSSN	**0.20**/38.52	**1.90**/8.21	**99.60**/94.89	**99.70**/96.36	**99.10**/90.01
WRN-28-10 CIFAR-10	TINc	23.40/**0.82**	11.60/**2.24**	94.20/**99.75**	92.80/**99.77**	94.70/**99.75**
	TINr	25.50/**2.94**	13.40/**3.83**	92.10/**99.36**	89.00/**99.4**	93.60/**99.36**
	LSUNc	21.80/**1.93**	9.80/**3.24**	95.90/**99.55**	95.80/**99.57**	95.50/**99.55**
	LSUNr	17.60/**0.88**	9.70/**2.52**	95.40/**99.7**	93.80/**99.72**	96.10/**99.68**
	UNFM	**0.00**/16.39	**0.20**/5.39	**100**/96.77	**100**/97.78	**100**/94.18
	GSSN	**0.00**/**0.00**	**0.10**/1.03	**100**/99.58	**100**/99.71	**100**/99.2
WRN-28-10 CIFAR-100	TINc	43.90/**9.17**	17.20/**6.67**	90.80/**98.22**	91.40/**98.39**	90.00/**98.07**
	TINr	55.90/**24.53**	23.30/**11.64**	84.00/**95.18**	82.80/**95.5**	84.40/**94.78**
	LSUNc	39.60/**14.22**	15.60/**8.2**	92.00/**97.38**	92.40/**97.62**	91.60/**97.16**
	LSUNr	56.50/**16.53**	21.70/**9.14**	86.00/**96.77**	86.20/**97.03**	84.90/**96.41**
	UNFM	**0.10**/99.9	**2.20**/14.86	**99.10**/83.44	**99.40**/89.43	**97.50**/71.2
	GSSN	**1.00**/98.26	**2.90**/16.88	**98.50**/93.04	**99.10**/88.64	**95.90**/71.62

Table 3: Distinguishing in- and out-of-distribution test set data for the image classification. All values are percentages. ↑ indicates larger value is better, and ↓ indicates lower value is better. Each value cell is in "ODIN[13]/Our Method" format.

ten from CIFAR-10 to CIFAR-100. Thus the number of splits need not be scaled linearly with the number of classes, making our method practical. We implicitly outperform Hendrycks and Gimpel [7] and Lee *et.al,* [12] as ODIN outperforms both these works and our method outperform ODIN on all but two benchmarks.

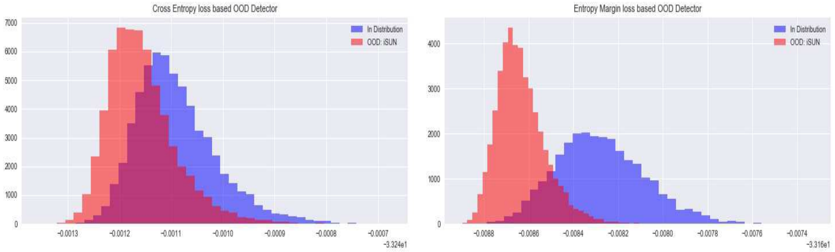
All three components in our method, namely novel loss function, the ensemble of leave-out classifiers and improved OOD detection metric contributed to the improvement in performance over state-of-the-art ODIN (refer to table 2). The contribution of these components can be seen in Table 2 in the rows marked as “Loss function”, “Number of splits” and “OOD detection scores”.

Our algorithm has stochasticity in the form of random splits of the classes. Given 100 classes in CIFAR-100, there are many ways to split 100 classes into 5 partitions. Table 4 gives the mean and standard deviation across five random ways to partition data when we use 5 number of splits for training. We note that even our worst case results outperform ODIN [13] on more challenging datasets.

Figure 1 compares the histogram of OOD detection scores on ID and OOD samples when different loss functions are used for training. Figure 1(a) is trained

OOD **Dataset**	**DenseNet-BC** **CIFAR-10**	**DenseNet-BC** **CIFAR-100**	**WRN-28-10** **CIFAR-10**	**WRN-28-10** **CIFAR-100**
TINc	1.49 ± 0.23	10.26 ± 1.33	1.3 ± 0.33	10.35 ± 2.21
TINr	3.95 ± 0.75	26.58 ± 4.16	4.56 ± 1.29	29.84 ± 5.12
LSUNc	4.54 ± 1.42	16.95 ± 1.27	3.81 ± 1.22	15.51 ± 1.4
LSUNr	1.3 ± 0.6	20.22 ± 2.79	1.53 ± 0.41	22.51 ± 6.08
UNFM	14.37 ± 31.84	38.79 ± 19.41	0.75 ± 1.24	47.67 ± 47.19
GSSN	27.09 ± 40.02	82.24 ± 12.81	31.47 ± 33.95	67.48 ± 44.33

Table 4: Mean and standard deviation of FPR at 95% TPR



(a) Cross entropy loss.

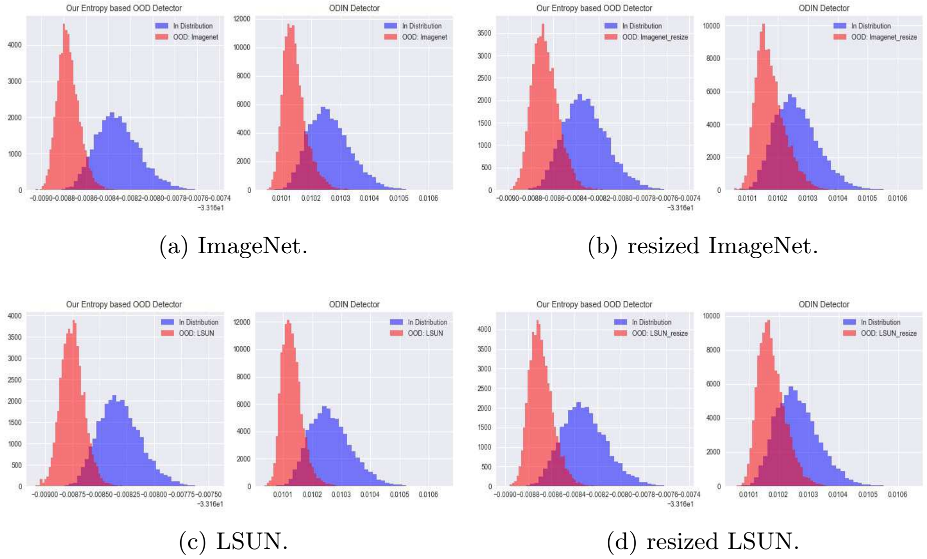
(b) Margin entropy loss.

Fig. 1: Histogram of ID and OOD detection scores of the proposed method and ODIN [13]

with only cross entropy loss, while Figure 1(b) is trained with proposed margin entropy loss and cross entropy, the proposed OOD detector is used in both bases. As shown in Figure 1, the proposed margin entropy loss helps to better separate ID and OOD distributions than using cross entropy loss alone. Figure 2 presents the histogram plot of OOD detection scores on ID and OOD samples for both our method and ODIN [13]. As shown in Figure 2, the proposed method has less overlap between OOD samples and ID samples compared to ODIN [13] and thus separates ID and OOD distributions better.

5 Conclusion and Future Work

As deep learning is widely adopted in many commercially important applications, it is very important that anomaly detection algorithms are developed for these algorithms. In this work, we have proposed an anomaly detection algorithm for deep neural networks which is an ensemble of leave-out-classifiers. These classifiers are learned by maximizing the margin-loss between the entropy of OOD samples and in-distribution samples. A random subset of training data serves as OOD data while the rest of the data serves as in-distribution. We show our algorithm significantly outperforms the current state-of-art methods [7], [12] and [13] across almost all the benchmarks. Our method contains three important components, novel loss function, the ensemble of leave-out classifiers, and novel



(a) ImageNet.

(b) resized ImageNet.

(c) LSUN.

(d) resized LSUN.

Fig. 2: Histogram of ID and OOD detection scores with proposed OOD detector v.s. ODIN OOD detector

out-of-distribution detector. Each of this component improves OOD detection performance. Each of them can be applied independently on top of other methods.

We also note that this method opens up several directions of research to pursue. First, the proposed method of the ensemble of neural networks requires large memory and computational resources. This can potentially be alleviated by all the networks sharing most of the parameters and branch away individually. Also, the number of splits can be used to trade off between detection performance and computational overhead. Notice that based on ablation study (Table 2) and detailed 3 splits results in supplementary document, even 3 splits outperform ODIN [13]. For use cases where reducing computational time is critical, we recommend to use 3 splits. Please see supplementary material for detailed results on 3 splits. Our current work requires an OOD dataset for hyper-parameter search. This problem can potentially be solved by investigating other surrogate functions for entropy which are better behaved with the epochs.

References

1. Bendale, A., Boult, T.E.: Towards open world recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1893–1902 (2015)
2. Bendale, A., Boult, T.E.: Towards open set deep networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1563–1572 (2016)
3. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Comput. Surv. **41**(3), 15:1–15:58 (2009)

4. Fujimaki, R., Yairi, T., Machida, K.: An approach to spacecraft anomaly detection problem using kernel feature space. In: KDD. pp. 401–410 (2005)
5. Goadrich, M., Oliphant, L., Shavlik, J.: Creating ensembles of first-order clauses to improve recall-precision curves. Machine Learning **64**, 231–262 (2006)
6. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems (NIPS) (2014)
7. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. In: International Conference on Learning Representations (ICLR) (2017)
8. Huang, G., Liu, Z., Weinberger, K.Q.: Densely connected convolutional networks (2016), arXiv preprint arXiv:1608.06993
9. https://tiny imagenet.herokuapp.com:
10. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images
11. LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F.: A tutorial on energy-based learning. In: Predicting structured data (2006)
12. Lee, K., Lee, H., Lee, K., Shin, J.: Training confidence-calibrated classifiers for detecting out-of-distribution samples. In: International Conference on Learning Representations (ICLR) (2018), `https://openreview.net/forum?id=ryiAv2xAZ`
13. Liang, S., Li, Y., Srikant, R.: Enhancing the reliability of out-of-distribution image detection in neural networks. In: International Conference on Learning Representations (ICLR) (2018)
14. Liu, F.T., Ting, K.M., Zhou, Z.: Isolation forest. In: ICDM. pp. 413–422 (2008)
15. Lu, W., Traoré, I.: Unsupervised anomaly detection using an evolutionary extension of k-means algorithm. IJICS **2**(2), 107–139 (2008)
16. Phua, C., Alahakoon, D., Lee, V.C.S.: Minority report in fraud detection: classification of skewed data. SIGKDD Explorations **6**(1), 50–59 (2004)
17. Rudd, E.M., Jain, L.P., Scheirer, W.J., Boult, T.E.: The extreme value machine. IEEE Trans. Pattern Anal. Mach. Intell. **40**(3), 762–768 (2018)
18. Scheirer, W.J., Jain, L.P., Boult, T.E.: Probability models for open set recognition. IEEE Trans. Pattern Anal. Mach. Intell. **36**(11), 2317–2324 (2014)
19. Scheirer, W.J., de Rezende Rocha, A., Sapkota, A., Boult, T.E.: Toward open set recognition. IEEE Trans. Pattern Anal. Mach. Intell. **35**(7), 1757–1772 (2013)
20. Xu, P., Ehinger, K.A., Zhang, Y., Finkelstein, A., Kulkarni, S.R., Xiao, J.: Turkergaze: Crowdsourcing saliency with webcam based eye tracking (2015), arXiv preprint arXiv:1504.06755
21. 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 (2015), arXiv preprint arXiv:1506.03365
22. Zagoruyko, S., Komodakis, N.: Wide residual networks (2016), arXiv preprint arXiv:1605.07146