Combining Outlierness Scores and Feature Extraction Techniques for Improvement of OoD and Adversarial Attacks Detection in DNNs

Tomasz Walkowiak[0000−0002−7749−4251], Kamil Szyc[0000−0001−6723−271X], and Henryk Maciejewski[0000−0002−8405−9987]

Wroclaw University of Science and Technology
{tomasz.walkowiak,kamil.szyc,henryk.maciejewski}@pwr.edu.pl

Abstract. Out-of-distribution (OoD) detection is one of the challenges for deep networks used for image recognition. Although recent works have proposed several state-of-the-art methods of OoD detection, no clear recommendation exists as to which of the methods is inherently best. Our studies and recent results suggest that there is no universally best OoD detector, as performance depends on the in-distribution (ID) and OoD benchmark datasets. This leaves ML practitioners with an unsolvable problem - which OoD methods should be used in real-life applications where limited knowledge is available on the structure of ID and OoD data. To address this problem, we propose a novel, ensemble-based OoD detector that combines outlierness scores from different categories: prediction score-based, (Mahalanobis) distance-based, and density-based. We showed that our method consistently outperforms individual SoTA algorithms in the task of (i) the detection of OoD samples and (ii) the detection of adversarial examples generated by a variety of attacks (including CW, DeepFool, FGSM, OnePixel, etc.). Adversarial attacks commonly rely on the specific technique of CNN feature extraction (GAP – global average pooling). We found that detecting adversarial examples as OoD significantly improves if we also ensemble over different feature extraction methods(such as GAP, cross-dimensional weighting (CroW), and layer-concatenated GAP). Our method can be readily applied with popular DNN architectures and does not require additional representation retraining for OoD detection.[1]

Keywords: Out-of-distribution detection · adversarial attack · CNN

1 Introduction

Applying deep neural networks in safety-critical systems requires that the models reliably recognize out-of-distribution (OoD) examples. Although many state-of-the-art methods of OoD detection have been proposed in recent works, no consensus has been worked out as to which of the methods is inherently best.

[1] All results are fully reproducible, the source code is available at `https://github.com/twalkowiak/WNN-OOD`

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_41

The methods differ in how the 'outlierness' of the samples is obtained and can be categorized as: classifier prediction score-based [8, 10, 16], distance-based [14, 21], or density-based [3]. A recent comprehensive study [28] suggests that there is no universally best OoD detector, as performance depends on the in-distribution (ID) and OoD benchmark data sets. Our studies summarized in Table 1 confirm this observation. We compare the performance of the methods: distance-based (Mahalanobis), prediction score-based (MSP), and density-based (LOF) and conclude that each of the methods outperforms other methods on some benchmarks. This poses an unsolvable problem - which OoD detection methods should we implement in a real-life deployment under limited knowledge of the structure of ID and OoD. We show that we can avoid this decision by using an ensemble of OoD detectors from different categories (prediction score-based, distance-based, and density-based) - as this combined OoD detector consistently outperforms popular detectors. We also find that ensembling over different representations (feature extraction procedures) can further improve OoD detection. We considered the following feature extractors: global average pooling (GAP), cross-dimensional weighting (CroW), layer-concatenated GAP (lcGAP), Global Maximum Pooling (GMP), and Selective Convolutional Descriptor Aggregation (SCDA).

Our contributions are the following. (i) We proposed an ensemble approach to OoD detection by (a) combining prediction score-based, distance-based, and density-based OoD detectors and (b) by combining OoD scores obtained under different feature extraction methods (GAP, CroW, lcGAP, GMP, SCDA). (ii) We showed on a comprehensive set of ID and OoD benchmarks that this approach consistently improves OoD detection compared with the individual SoTA algorithm. This holds for different DNN models (VGG16, ResNet, WideResNet, DenseNet, ShuffleNetV2, and MobileNetV2). (iii) We also showed that the proposed method detects adversarial examples generated by a wide range of adversarial attacks as OoD. (iv) We performed a sensitivity analysis of the ensemble method to quantify the contribution of the individual ensemble members to the performance of the proposed OoD detector.

2 Related work

OoD detection based on standard representations. Many OoD methods rely on representations trained to discriminate between classes given in the training data set. These methods differ in the technique used to estimate outlierness scores to detect OoD examples. Popular methods include: the OpenMax score proposed by [2], the MSP (maximum softmax probability) score [8], the ODIN score [16], generalized ODIN [10], the Mahalanobis distance-based score [14], or density-related scores such as kNN-based [26], or LOF-based [3, 30]. These methods allow for post hoc detection of OoD inputs without requiring dedicated training of DNNs for OoD detection. Our proposed method belongs to this line of research.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_41

Table 1. Comparison of different OoD detection methods for CIFAR-10 as ID. There is no one universal OoD method for all benchmarks. A different strategy is better for different CNN models and pairs of ID and OoD. The tested OoD methods operate according to different principles: distance-based (Mahalanobis), prediction score-based (MSP), and density-based (LOF). Our results are in line with recent works[28]. This poses an unsolvable problem: which OoD detection method we should implement in a real-life deployment with limited knowledge of ID and OoD.

Model	OoD	AUROC	DATACC	TNR at TPR 95%
		Mahalanobis/MSP/LOF		
ResNet	SVHN	92.6/92.2/**95.8**	85.2/87.2/**89.4**	61.1/43.6/**79.5**
	CIFAR-100	82.0/**87.2**/82.5	74.6/**80.8**/75.4	29.3/34.6/**35.5**
WideResNet	SVHN	**97.8**/93.0/95.3	**92.9**/88.2/90.2	**88.0**/60.0/69.7
	CIFAR-100	**90.3**/86.8/89.0	**83.2**/81.6/81.9	50.4/44.9/**50.8**
MobileNetV2	SVHN	85.4/**86.3**/76.1	80.6/**81.7**/69.8	20.6/**31.4**/20.9
	CIFAR-100	**86.9**/82.3/85.0	**80.4**/78.1/77.3	36.9/33.0/**40.9**

Table 2. Comparison of different feature extraction detection methods using Mahalanobis and LOF as OoD methods and CIFAR-10 as ID. We propose to use different feature extraction techniques in OoD detection problems. It may significantly improve achieved results compared to the default GAP. Various approaches focused on different components (e.g., on edges, patterns, or whole objects), so different features can effectively separate data for other pairs of ID and OoD. So, as with choosing an OoD method, there is no universal feature extraction strategy.

Model	OoD	Method	AUROC	DATACC	TNR at TPR 95%
			CroW/GAP/lcGAP		
ResNet	SVHN	Mah	94.0/92.6/**94.6**	86.4/85.2/**87.8**	67.7/61.1/**70.5**
		LOF	**96.5**/95.8/96.5	**90.3**/89.4/**90.3**	**83.0**/79.5/81.0
	CIFAR-100	Mah	**83.2**/82.0/80.5	**75.8**/74.6/73.0	**32.1**/29.3/28.8
		LOF	83.1/82.5/**83.4**	**75.9**/75.4/75.7	37.1/35.5/**37.5**
WideResNet	SVHN	Mah	**98.0**/97.8/97.0	**93.1**/92.9/91.1	**88.8**/88.0/85.3
		LOF	95.4/95.3/**98.2**	90.3/90.2/**93.7**	70.9/69.7/**90.0**
	CIFAR-100	Mah	**90.5**/90.3/84.8	**83.5**/83.2/77.3	**51.5**/50.4/40.3
		LOF	89.0/89.0/**89.4**	81.8/81.9/**82.2**	51.5/50.8/**52.9**
MobileNetV2	SVHN	Mah	84.9/85.4/**86.2**	80.2/80.6/**81.6**	19.5/20.6/**20.9**
		LOF	76.6/76.1/**92.7**	70.0/69.8/**87.6**	20.9/20.9/**44.6**
	CIFAR-100	Mah	86.9/86.9/**87.0**	80.3/80.4/**80.5**	**37.2**/36.9/37.0
		LOF	85.0/85.0/**86.8**	77.3/77.3/**79.2**	40.4/40.9/**43.8**

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_41

OoD detection for classification with additional training. The methods in this group include outlier-exposed techniques, for example, [9], where models are trained to produce uniformly distributed predictions for the OoD data available during training. [5] propose to train with outliers synthesized in the feature space (rather than the image). Another line of methods attempts to improve representations for OoD detection using contrastive learning and self-supervised techniques [27, 32, 23]. These methods learn representations in which transformed (augmented) versions of an image are pulled closer to each other while pushing other images further away. This modified feature space often leads to better OoD detection.

3 Method

3.1 Different categories of OoD detection methods

Here we briefly present the OoD detectors used in the proposed ensemble procedure: predictive score-based (MSP), distance-based (Mahalanobis), and density-based (LOF).

Maximum Softmax Prediction (MSP) [8] quantifies outlierness based on the prediction score from the neural network. More specifically, the confidence score for OoD detection is based on the maximum output of the softmax layer of the DNN.

OoD detection based on the Mahalanobis distance is based on the estimation of the multivariate Gaussian (MVN) distribution as a model of the class-conditional posterior distribution. Given the ID training dataset $X_c \subset R^d$ for class $c \in C = \{1, 2, \ldots, m\}$, we estimate the MVN model for class $\mathcal{N}(\mu_c, \Sigma_c)$ with the mean vector μ_c and the covariance matrix Σ_c estimated from X_c (some methods estimate Σ_c from $\bigcup_{c \in C} X_c$, see, e.g., [14]). The confidence score for the detection of OoD of a test sample u is then obtained as the shortest negative Mahalanobis distance for all known classes: $cs_{Mah}(u) = -\min_c \sqrt{(u-\mu_c)^\top \Sigma_c^{-1}(u-\mu_c)}$.

The Local Outlier Factor (LOF) [3] obtains OoD scores based on nonparametric estimates of density. More specifically, it calculates the local reachability density $LRD_k(u, X)$ of a sample u with respect to the known dataset X. LRD is the ratio of an average reachability distance between a given point, its k neighbors, and their neighbors. K-neighbors ($N_k(u, X)$) includes a set of points that lie in the circle of radius k-distance, where k-distance is the distance between the point, and it is the farthest k^{th} nearest neighbor ($||N_k(u, X)|| >= k$). The confidence score for the detection of OoD is then the inverse of LOF defined in [3], i.e. $cs_{LOF}(u, X) = -\frac{\sum_{x \in N_k(u, X_i)} LRD_k(x, X)}{||N_k(u,X)|| LRD_k(u,X)}$.

3.2 Feature extraction

The standard method for CNN feature extraction is Global Average Pooling (GAP), proposed by [17]. This approach is widely used in networks designed for

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_41

image classification since it is robust to spatial translations of the input data. However, many works, especially on image retrieval, propose different feature extraction strategies, focusing on local (object details) or global (whole object) descriptions and low-level (e.g., shapes, textures) or high-level (whole image meaning) features. Different feature extractors focus on specific image components, so various features may prove helpful for other pairs of ID and OoD datasets to detect OoD samples effectively.

Therefore, we also propose to analyze other feature extraction methods for OoD detection problems, such as CroW[12], GAP, lcGAP (our, inspired by [15]), GMP, and SCDA[31]. Most methods work based on the selected convolutional layer (usually the last) T with shape (w, h, c), where w refers to the width, h to height, and c to channels. A feature vector V (with length c) is calculated as follows.

For GAP as $\frac{1}{wh}\sum_{i=0}^{w}\sum_{j=0}^{h} T_{i,j,k}$ and $max_{i,j}^{w,h} T_{i,j,k}$ for GMP (Global Maximum Pooling) for each channel k independently.

For CroW (cross-dimensional weighting), T is first weighted channel-wise by weight vectors β_k and then location-wise by a weight matrix α, that is, we define weighted T as $T'_{i,j,k} = \alpha_{ij}\beta_k T_{i,j,k}$. Next, sum-pooled is performed to aggregate T' features.

SCDA (Selective Convolutional Descriptor Aggregation) is based on an activation feature map. First, the aggregation map $A_{i,j}$ is obtained as $\sum_{k=1}^{c} T_{i,j,k}$. For the aggregation map A, there are w, h summed activation responses corresponding to positions w, h. Next, the mask map M of the same size as A is obtained as $M_{i,j}^{w,h} = 1$ if $A_{i,j} > \bar{a}; 0$ otherwise, where $\bar{a}$ is the mean value of all positions in A. The final feature vector is selected based on the largest connected component of the mask map M. The layer-concatenated GAP (lcGAP) method (our original proposition) uses concatenated features from different convolutional layers after each block using GAP.

These methods have different characteristics; most are global descriptions with high-level features. The GAP method is more robust to scale changes because the GMP response of the feature map does not change rapidly with scale change. CroW uses a weighting and aggregation scheme to transform convolutional image features into compact global image features. SCDA focuses on local descriptions. lcGAP considers low-level features.

We show that the feature extraction strategy can affect the defense against adversarial attacks. These attacks aim to cheat the CNN classifier that commonly relies on GAP features. Different feature extraction methods can make features more resilient and less vulnerable to attacks. See Section 4.2 for more information on attacks.

3.3 Proposed ensemble OoD detector

The results in Tables 1 and 2 suggest no best method among feature extractors and OoD detectors exists. Therefore, we consider combining confidence scores from all available sources using an MLP regressor. The idea of using an OoD

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_41

Algorithm 1: WNN ensemble OoD detector. WNN.fit method trains the detector, WNN.score computes the confidence score for OoD detection based on the ensemble of OoD detection and feature extraction methods. See Section 3.3 for further details.

```
Inputs  : data_IN_train, data_IN_test, data_OOD
Outputs: final_confidence_scores
WNN - ensemble OoD detector
    scores = []
    foreach feature_method in [feature extraction methods] do
        features_IN_train = feature_method.extract_from(data_IN_train)
        features_IN_test = feature_method.extract_from(data_IN_test)
        features_OOD = feature_method.extract(data_OOD)
        foreach ood_method in [OoD detection methods] do
            ood_method.fit(features_IN_train)
            score_IN_test = ood_method.score_and_norm(features_IN_test)
            score_OOD = ood_method.score_and_norm(features_OOD)
            scores.append([score_IN_test, score_OOD])
    calibration_scores, test_scores = split_data(scores)
    WNN = new MLP_Model()
    WNN.fit(calibration_scores)
    final_confidence_scores = WNN.score(test_scores)
```

ensemble was inspired by [14] (although they used linear regression) and, in general, by the well-known concept of stack generalization ensemble [29].

The pseudocode of the proposed ensemble OoD detector is shown in the Algorithm 1.

Given the training data, we extract CNN features using all the methods discussed in Section 3.2. We then build OoD detectors for each OoD type (see Section 3.1) and each feature extractor. This results in a set of 15 different OoD detectors. The idea of the proposed method is to combine the confidence scores returned by these OoD detectors into the final ensemble-based score.

The confidence score ranges for each OoD method are very different. Mahalanobis gives values below 0 (lower limits depend on the data), MSP values range from 0 to 1 (inclusive), whereas LOF shows values below or equal to -1. Therefore, we scale confidence scores to the range 0-1 using the estimated cumulative density on the validation set of inliers (estimation is done using step functions).

Next, we train the MLP that combines the weighted confidence scores from individual OoD detectors. Similarly to [14, 33], we use a calibration set consisting of images from in- and out-of-distribution datasets. We use a fully connected NN consisting of one hidden layer (100 neurons) and a single output. We use ReLU activations for the hidden layer and linear activation in the final layer. We employ Adam optimizer during backpropagation. The output of this network serves as

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_41

the final confidence score. The method is named WNN from the weighted neural network.

4 Experiments and Results

4.1 OoD Detection Problem

We run OoD detection problems using widely used benchmark datasets and CNNs models such as VGG16 [24], ResNet [7], WideResNet, DenseNet [11], ShuffleNetV2 [18] and MobileNetV2 [22]. All models were trained on CIFAR-10. To evaluate the OoD methods, we used CIFAR-10 as ID data and CIFAR-100 or SVHN as OoD data. We kept a 1:1 proportion of known and unknown samples (10,000:10,000). We used the standard metrics in the results: Area Under Receiver Operating Characteristic curve (AUROC), detection accuracy (DTACC) that defines the ratio of correct classification, and True Negative Rate at 95% True Positive Rate (TNR at TPR 95%). The higher the values of all metrics, the better the detection of OoD. As the measures are strongly correlated, for some results, we present only AUROC. All our tests can be replicated using the standard workstations for deep learning; we used Nvidia GeForce GTX 1080/2080 Ti.

No one universal OoD method In the first experiment, we show that there is no universally best OoD method for all benchmarks. In Table 1, we compare the OoD methods described in Section 3.2 and show that for different pairs of IN and OoD datasets, different OoD detectors are the best.

For example, for WideResNet, the parametric Mahalanobis method is the best for all benchmarks. For ResNet, depending on the OoD data and chosen metrics, the best performance is achieved for the density-based LOF or the logits-based MSP. These results are consistent with the recent literature [28].

No one universal feature extraction technique We propose using different feature extraction strategies (see Section 3.2) in the OoD detection problem rather than relying on a single method. In Table 2, we show that for different pairs of IN and OoD samples, different strategies work better, as different image features are needed to separate these image categories. For example, WideResNet using LOF and SVHN performed best for lcGAP for AUROC (2.9 p.p. better than GAP), and ResNet using Mahalanobis and CIFAR-100 performed best for CroW for AUROC (1.2 p.p. better than GAP).

Proposed method results In Table 3, we report the performance of WNN - our ensemble method introduced in Section 3.3. We compare our method with individual OoD detectors used in the ensemble (LOF, Mahalanobis, MSP) and with two popular SoTA OoD detectors: KNN [26] (a density-based non-parametric method), and UF [14] (Mahalanobis distance-based, with a common covariance

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_41

matrix and feature ensemble from different layers of a CNN). Since the WNN results depend on the distribution of the validation data (20% random samples from the original data were used to build the WNN) and the MLP training process, we repeated the experiments ten times and showed the mean and std results. Due to a random split on validation and test data, the results for all other methods (including those not requiring validation) are also presented as mean and standard deviation. The proposed WNN method gives the best results for most benchmarks. To verify the claim that adding other feature extraction strategies than GAP improves OoD detection, we also considered the results for WNN^{GAP}, the proposed ensemble that works only on GAP features. We find that WNN consistently outperforms WNN^{GAP}, which is often the second best.

Experiments on other OoD detection benchmark datasets We also verified the performance of the proposed method on other OoD datasets (i.e., Tiny ImageNet, ImageNet-O, and Textures) and also used CIFAR-100 as in-distribution (with SVHN and CIFAR-10 as OoD).[2] We found the same conclusions as reported in the previous section. The results on CIFAR-100 as in-distribution show that WNN outperforms the MSP, Mahalanobis, LOF, and kNN outlier detectors in all test cases (by ca. 4 pp over the second best). Taking different OoD detectors and different feature generators into an ensemble leads to a significant improvement over individual non-ensemble OoD detectors. Thus, the practitioner using the proposed WNN is freed from the need to select the best / most appropriate OoD detector for a given study (which is difficult to do in practice).

4.2 Attacks

We used our method as a defensive approach against adversarial attacks. The goal is to use adversarial examples as OoD samples similar to [14]. These attacks modify the input image by adding unique noise, making the network fooled. Noise usually does not change the human classification assessment. Although there are numerous dedicated defense methods[34], our approach allows for initial protection against many kinds of adversarial attacks.

We tested various adversarial attack methods: CW, DeepFool, FGSM, One-Pixel, PGD, and Square. One of the fundamental methods is FGSM[6] (Fast Gradient Sign Method), where a small perturbation is added to maximize the loss function. The PGD[19] (Projected Gradient Descent) is an extension of the FGSM by repeating the addition of those perturbations multiple times. The Deep Fool[20] method is another iterative approach. This method calculates the distributions based on the willingness to move the input across the decision boundaries with minimal changes. The first-order approximation of Taylor's expansion is used on a linear model to find these distributions.

CW[4] focused on solving the optimization problem of finding the minimized distance between two images (standard and attacked) so that the classification

[2] The detailed results are available `https://github.com/twalkowiak/WNN-OOD`

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_41

Table 3. OoD detection results for CIFAR-10 (as inliers) versus SVHN/CIFAR-100 (as outliers). Mahalanobis, MSP[8] and LOF[3] are working on features extracted from the last layer of CNN using the classic GAP method. WNN (our method) combines 15 OoD detectors (obtained by three OoD methods and five feature extractors: GAP, CroW, lcGAP, SCDA, and GMP). WNN^{GAP} represents the combination limited to GAP-based features only (i.e., it is a limited version of WNN). We can see that adding information from additional feature extractors increases the OoD detection. The results show the mean and std values of the metrics achieved.

	Method	AUROC	DATACC SVHN	TNR	AUROC	DATACC CIFAR-100	TNR
VGG16	Mah	90.0±0.09	84.5±0.13	37.8±0.35	87.4±0.08	80.6±0.16	39.5±0.39
	MSP	89.5±0.08	85.1±0.08	30.1±0.38	86.4±0.10	80.1±0.09	32.5±0.48
	LOF	87.2±0.12	80.3±0.10	37.7±0.88	83.6±0.13	76.8±0.12	36.2±0.76
	WNN^{GAP}	90.4±0.11	85.5±0.08	34.4±0.82	87.6±0.13	80.6±0.16	37.8±0.84
	WNN(our)	97.5±0.10	92.2±0.18	86.2±0.92	**88.3±0.13**	**81.2±0.09**	41.3±1.18
	KNN	91.3±0.15	86.4±0.13	39.3±0.88	87.5±0.15	80.5±0.16	38.4±0.35
	UF	**98.9±0.10**	**95.2±0.19**	**95.2±0.35**	86.8±0.70	79.6±0.87	**44.3±1.02**
ResNet	Mah	92.7±0.06	85.3±0.04	61.3±0.32	82.1±0.09	74.7±0.10	29.3±0.39
	MSP	92.3±0.10	87.2±0.10	44.0±0.63	87.2±0.07	80.8±0.11	34.6±0.59
	LOF	95.8±0.05	89.4±0.10	79.8±0.41	82.6±0.12	75.5±0.11	36.5±0.88
	WNN^{GAP}	96.3±0.07	90.5±0.11	79.8±0.48	88.3±0.16	81.1±0.13	40.6±1.06
	WNN(our)	**99.3±0.03**	**95.8±0.10**	**96.2±0.12**	**89.3±0.14**	**82.0±0.25**	**45.1±0.75**
	KNN	96.4±0.05	90.4±0.10	78.9±0.52	87.5±0.08	80.2±0.09	41.5±0.52
	UF	97.6±0.31	92.4±0.46	88.5±1.61	75.7±0.49	68.9±0.41	22.6±0.93
WideResNet	Mah	97.9±0.03	93.0±0.07	88.0±0.30	90.3±0.11	83.1±0.12	50.6±0.56
	MSP	93.0±0.12	88.2±0.13	59.6±0.56	86.8±0.12	81.6±0.12	45.1±0.65
	LOF	95.3±0.05	90.2±0.07	70.0±0.53	89.0±0.12	82.0±0.13	50.9±0.80
	WNN^{GAP}	97.4±0.10	92.7±0.13	85.4±1.26	90.4±0.11	83.4±0.17	51.6±0.55
	WNN(our)	**99.2±0.02**	**95.4±0.11**	**95.8±0.23**	**91.1±0.08**	**83.9±0.14**	**53.9±0.62**
	KNN	96.9±0.06	91.8±0.12	80.0±0.51	90.9±0.10	83.8±0.10	52.7±0.49
	UF	97.4±0.18	92.3±0.42	89.4±1.06	67.9±1.82	63.1±1.48	17.5±0.48
DenseNet	Mah	99.0±0.03	95.1±0.06	94.7±0.17	91.1±0.09	84.3±0.11	57.6±0.43
	MSP	82.5±0.16	79.4±0.13	46.1±0.63	88.5±0.12	82.7±0.12	47.7±1.19
	LOF	97.7±0.03	93.0±0.08	86.3±0.19	89.8±0.10	83.1±0.15	54.5±0.49
	WNN^{GAP}	98.8±0.05	94.8±0.09	93.7±0.26	91.4±0.08	84.6±0.07	56.6±0.57
	WNN(our)	99.2±0.03	95.8±0.05	96.5±0.09	**92.0±0.13**	**85.1±0.11**	**59.7±0.78**
	KNN	98.4±0.04	94.0±0.09	90.8±0.36	90.9±0.07	84.3±0.07	55.5±0.67
	UF	**99.3±0.12**	**96.2±0.40**	**97.0±0.67**	81.2±1.96	74.8±1.65	38.6±2.88
ShuffleNetV2	Mah	93.3±0.11	88.2±0.13	52.2±1.02	87.2±0.08	80.4±0.11	37.7±0.40
	MSP	90.4±0.12	85.8±0.14	41.5±1.17	82.0±0.10	78.3±0.09	29.6±0.73
	LOF	89.9±0.12	83.5±0.16	39.3±0.69	84.7±0.13	77.6±0.12	36.4±0.26
	WNN^{GAP}	93.3±0.15	88.1±0.15	52.9±1.19	87.2±0.07	80.3±0.09	37.8±0.44
	WNN(our)	**97.2±0.07**	**91.0±0.16**	**82.2±0.66**	**87.6±0.13**	**80.7±0.17**	**39.0±0.87**
	KNN	94.5±0.06	89.2±0.08	60.9±0.60	87.5±0.09	80.5±0.13	38.6±0.45
	UF	98.0±0.09	93.3±0.14	90.0±0.69	86.8±0.28	79.2±0.19	39.7±1.33
MobileNetV2	Mah	85.4±0.11	80.6±0.08	20.7±0.30	87.0±0.11	80.4±0.14	37.2±0.53
	MSP	86.3±0.12	81.7±0.12	31.6±0.57	82.2±0.12	78.1±0.13	33.1±0.50
	LOF	76.0±0.15	69.8±0.14	20.8±0.30	85.0±0.12	77.3±0.15	41.1±0.60
	WNN^{GAP}	89.0±0.16	83.4±0.09	36.9±1.93	87.1±0.09	80.3±0.11	39.1±0.61
	WNN(our)	97.6±0.11	92.0±0.16	86.1±0.93	**87.9±0.11**	**80.9±0.13**	**42.5±0.71**
	KNN	86.7±0.11	81.5±0.11	24.9±0.42	87.5±0.10	80.5±0.13	39.3±0.52
	UF	**98.2±0.14**	**93.3±0.31**	**91.0±0.67**	85.6±0.26	78.6±0.20	35.8±3.20

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_41

result will differ for both examples. The problem is not trivial due to the highly non-linear nature of deep models, so the authors propose defining the objective function instead, which is much more likely to optimize using popular optimizers. Square[1] is based on a randomized search scheme in which we select localized square-shaped updates at random positions. The OnePixel[25] is highly interesting due to the change of only one pixel to fool the network. It uses a differential evolution algorithm to find which pixel should be changed. Candidate solutions contain information with x,y coordinates, and RGB values. During each epoch, the population is randomly modified by a minor factor, and the algorithm works until one of the candidates is an adversarial attack.

In our experiments, we generated 1000 examples for each attack method using the TorchAttacks[13] library. We treated the attacks as OoD data by sampling the same number of images for the test-known subset. Similarly to previous experiments, we used 20% random samples to build the WNN and the regressor in UF. We repeated the experiments ten times (with different test image subsets), presenting the results as mean and standard deviation. We kept high confidence in attack examples, usually ensuring that the closed-set classifier's output would be above 95% certainty for the wrong class.

We have shown the results of our experiments in Table 4. Although we present results for the ResNet model, the results and conclusions obtained for other models are consistent. Popular methods like Mahalanobis, MSP, and LOF are not suitable for defense against adversarial attacks. UF works well only for FGSM and PGD, which are the same family of attacks. KNN works only for PGD in practice. Our method is the best approach for all tested attacks, significantly outperforming other methods like CW (7.2 p.p. better than second-best), OnePixel (6.6 p.p. better), or DeepFool (4.1 p.p. better). See that our proposed method is very stable.

The adversarial attack aims to fool CNN's classifier part based on the GAP features. Using different feature extraction strategies should improve results in defense against adversarial examples. We can compare the two methods WNN^{GAP} and WNN (proposed), where the first uses only GAP features, and the second uses all five feature extraction techniques. We can see that for all attacks, adding new features improves results, with significant improvement for some attacks, like FGSM (16.7 p.p. better than with only GAP), OnePixel (6.7 p.p. better), or CW (6.1 p.p. better).

4.3 Sensitivity Analysis

The proposed WNN method combines different sources of information (different feature extractors and Ood detectors). The question of how these sources affect the final detection performance arises. Comparing the results labeled WNN^{GAP} (WNN only on GAP features) with results labeled WNN (all feature extractors) shown in Tables 3 and 4, we conclude that extending the feature extractors beyond GAP increases OoD performance. The impact of the OoD methods used in the ensemble is shown in Table 5. It clearly shows that more methods (Ma-

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_41

Table 4. Comparison of AUROC (%) for ResNet trained on CIFAR-10 and different types of attacks: CW[4], DeepFool[20], FGSM[6], OnePixel[25], PGD[19], Square[1]. Our method is the best approach for all tested attacks, outperforming others. Moreover, we showed that extending the WNN by other feature extraction techniques than GAP methods significantly improves the ability of defenses.

Attack	Mah	MSP	LOF	WNN^{GAP}	WNN	UF	KNN
CW	65.1±0.83	80.8±1.10	63.5±1.44	81.9±1.68	**88.0±3.68**	59.8±2.03	70.3±0.96
DeepFool	63.6±0.92	79.8±0.88	60.2±1.10	79.6±0.82	**83.9±2.89**	52.7±2.36	67.3±0.61
FGSM	73.4±0.80	68.7±1.50	70.7±1.00	76.3±1.17	**93.0±0.47**	92.2±0.44	71.7±1.23
OnePixel	61.2±1.48	75.5±1.63	55.4±1.16	75.4±1.21	**82.1±2.43**	53.9±1.42	65.2±1.04
PGD	86.2±0.53	0.8±0.05	98.2±0.12	99.3±0.13	**99.7±0.08**	97.1±0.21	96.7±0.20
Square	78.7±1.08	80.2±0.92	76.7±0.64	87.1±1.40	**90.2±1.67**	75.1±1.54	80.2±0.86

halanobis, MSP, and LOF) are better (than only Mahalanobis or Mahalanobis and MSP).

We also checked the effect of the size of the validation set on the WNN results. To test this, we analyzed the proportions of the original test data used for validation in the CIFAR10 vs. CIFAR-100 detection task for Wide Resnet. We show the results in Fig. 1, along with the performance of UF (as another ensemble technique). The validation size affects the performance (AUROC), but WNN still performs better than the MSP method for a small number, such as 20 outlier images (a factor of 0.002) used to build the ensemble. WNN gives significantly lower confidence intervals, even though it uses a larger ensemble model (i.e., MLP regressor) than UF (linear regressor). It is caused by the fact

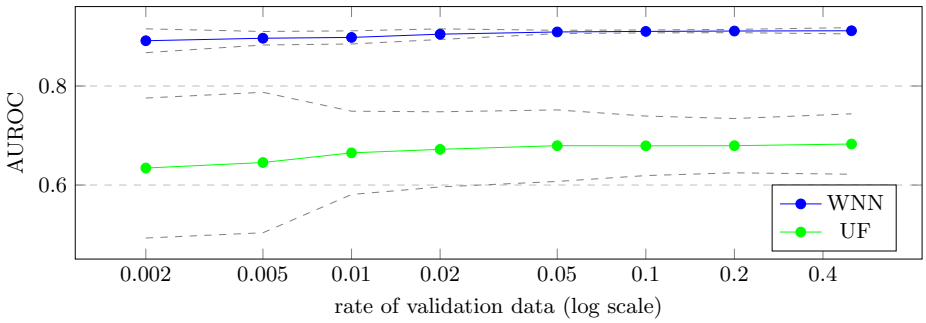

Fig. 1. Effect of validation size on OoD detection for WNN and UF methods. The experiments were performed on Wide ResNet with CIFAR-100 used as outliers. The ratio ranges from 0.002 to 0.5 of the original inliers and outliers images (10,000 in each set). The experiments (splitting the dataset and tuning the ensemble parameters) were repeated 50 times, so the results were random. The solid line shows the mean value of AUROC, whereas the dashed line shows three times the std.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_41

that UF uses both ensemble and the input pre-processing[16] magnitude meta-parameter set up in the validation set. Moreover, the effective data size used to set up the regressor is reduced by half.

4.4 Limitations

We see three limitations of the proposed method: the need for validation samples to tune the ensemble model, the difficulty in choosing the final methods used in the WNN, and the required calculation of many OoD confidence scores, some of which can be computationally heavy.

The first limitation is typical for a large group of OoD methods (such as [14, 16, 9]). The problem is that, in most practical scenarios, either OoD samples are unavailable or cover a small fraction of the OoD sample space. The results in Fig. 1 suggest that WNN performance decreases when the amount of OoD available for training decreases, but for just 20 OoD images, the WNN performance is still better than the baseline (MSP). Furthermore, the ability of validation set-based techniques to detect OoD decreases when the OoD samples are from a different OoD domain from the one used to tune the hyperparameters.

Our WNN method uses OoD detectors obtained by three OoD detection (LOF, Mahalanobis, and MSP) and five feature extraction (CroW, GAP, GMP,

Table 5. The impact of the number of OoD methods used in an ensemble. Results present AUCROC for an ensemble over the feature extraction methods and just the Mahalanobis method (WNN^{Mah}), Mahalanobis plus MSP (WNN^{MahMSP}), and all methods (proposed WNN). A steady increase in performance could be observed by adding more OoD methods. The last column shows the best results for other (non-WNN) methods, i.e., the best among Mahalanobis, MSP, LOF, UF, and KNN. We notice that WNN without LOF (the method with large computational complexity is only slightly worse than the proposed WNN.

Model	OoD	WNN^{Mah}	WNN^{MahMSP}	WNN	other best
VGG16	SVHN	90.1±0.16	90.6±0.22	97.5±0.10	98.9±0.10 (UF)
	CIFAR-100	87.4±0.07	87.7±0.13	88.4±0.12	87.5±0.15 (KNN)
ResNet	SVHN	98.9±0.05	99.2±0.03	99.2±0.02	97.6±0.31 (UF)
	CIFAR-100	85.7±0.13	89.2±0.15	89.4±0.11	87.5±0.08 (KNN)
WideResNet	SVHN	99.2±0.03	99.2±0.03	99.2±0.01	97.9±0.03 (Mah)
	CIFAR-100	90.9±0.11	91.0±0.10	91.1±0.12	90.9±0.10 (KNN)
DenseNet	SVHN	99.1±0.05	99.2±0.04	99.2±0.03	99.3±0.12 (UF)
	CIFAR-100	91.4±0.06	91.8±0.09	92.0±0.10	91.1±0.09 (Mah)
ShuffleNetV2	SVHN	96.8±0.07	96.8±0.11	97.2±0.11	98.0±0.09 (UF)
	CIFAR-100	87.3±0.12	87.2±0.10	87.6±0.09	87.5±0.09 (KNN)
MobileNetV2	SVHN	93.6±0.11	94.0±0.08	97.6±0.08	98.2±0.14 (UF)
	CIFAR-100	87.1±0.13	87.2±0.09	87.8±0.13	87.5±0.10 (KNN)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_41

lcGAP, and SCDA) methods. We chose the above methods because of the differentiated nature of the operating principle (see details in Sections 3.1 and 3.2). However, the proposed solution can be used as a framework, combining confidence scores from any OoD detectors. Finding the optimal set of OoD methods for WNN can be a limitation due to the number of methods available in the literature. We suggest using the most diversified OoD detection methods.

The last limitation is the required calculation of many OoD confidence scores. Some of them might be computationally heavy. For instance, in our proposed set of methods WNN uses, LOF is computationally intensive compared to Mahalanobis or MSP due to the required calculation of the nearest-neighbor aspects. For data of dimensions d and train size N, the LOF model's complexity is $O(d * N^2)$, and OoD detection is $O(d * N)$. However, it can be speed up (for a certain distance metric) by using the k-d tree (as implemented in scikit-learn) or R*-tree (as implemented in the ELKI framework), giving $O(d * NlogN)$ for model building and $O(d * logN)$ for detection. If the reduced time complexity is unacceptable, one can build the WNN only on Mahalanobis and MSP. The results presented in Table 5 (column WNN^{MahMSP}) show that this solution is slightly worse than the proposed WNN but still better than other OoD detectors tested.

5 Conclusion

In this work, we proposed an ensemble procedure for OoD detection that combines OoD detectors based on prediction scores (MSP), distance-based (Mahalanobis), and density-based (LOF). Using several benchmarks, we showed that this procedure outperforms each OoD algorithm used in the ensemble. These results are consistent for different DNN architectures.

We also showed that for OoD detection, different feature extraction strategies are worth considering, as this allows us to broaden the representation of objects. Different DNN feature extraction strategies focus on local (object details) or global (whole object) characteristics and either low-level (e.g., shapes, textures) or high-level (entire image meaning) features. We showed that the ensemble OoD detector that combines different feature extractors (we used GAP, CroW, lcGAP, GMP, and SCDA) leads to further improvement in OoD detection. We also found that the proposed method is efficient in recognizing adversarial examples. Moreover, our method more reliably identifies adversarial examples as OoD than individual SoTA OoD detectors for a wide range of adversarial attacks.

Finally, our method can be used as a generic framework that combines different outlierness scores (ensemble over OoD detectors of different natures) and different representations (ensemble over feature extractors). The essential contribution is (a) to show that OoD detection in DNNs consistently improves if based on ensembled information and (b) to propose the practical technique to ensemble over OoD detectors and feature extractors. Incorporating other OoD detectors and feature extractors into our framework may lead to further improvement - we leave this as future work.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_41

Our method can facilitate the application of state-of-the-art, pre-trained DNN models to real-world, safety-critical image and text recognition systems where efficient OoD detection is mandatory.

References

1. Andriushchenko, M., Croce, F., Flammarion, N., Hein, M.: Square attack: a query-efficient black-box adversarial attack via random search. In: European Conference on Computer Vision. pp. 484–501. Springer (2020)
2. Bendale, A., Boult, T.E.: Towards open set deep networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1563–1572 (2016)
3. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: Identifying density-based local outliers. SIGMOD Rec. **29**(2), 93–104 (may 2000). https://doi.org/10.1145/335191.335388
4. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp). pp. 39–57. IEEE (2017)
5. Du, X., Wang, Z., Cai, M., Li, S.: Towards unknown-aware learning with virtual outlier synthesis. In: Proceedings of the International Conference on Learning Representations (2022)
6. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
8. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. arXiv preprint arXiv:1610.02136 (2016)
9. Hendrycks, D., Mazeika, M., Dietterich, T.: Deep anomaly detection with outlier exposure. Proceedings of the International Conference on Learning Representations (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 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10951–10960 (2020)
11. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
12. Kalantidis, Y., Mellina, C., Osindero, S.: Cross-dimensional weighting for aggregated deep convolutional features. In: European conference on computer vision. pp. 685–701. Springer (2016)
13. Kim, H.: Torchattacks: A pytorch repository for adversarial attacks. arXiv preprint arXiv:2010.01950 (2020)
14. Lee, K., Lee, K., Lee, H., Shin, J.: A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. p. 7167–7177. NIPS'18, Curran Associates Inc., Red Hook, NY, USA (2018)
15. Li, Y., Xu, Y., Wang, J., Miao, Z., Zhang, Y.: Ms-rmac: Multiscale regional maximum activation of convolutions for image retrieval. IEEE Signal Processing Letters **24**(5), 609–613 (2017)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_41

16. 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)
17. Lin, M., Chen, Q., Yan, S.: Network in network. arXiv preprint arXiv:1312.4400 (2013)
18. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European conference on computer vision (ECCV). pp. 116–131 (2018)
19. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
20. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2574–2582 (2016)
21. Ren, J., Fort, S., Liu, J., Roy, A.G., Padhy, S., Lakshminarayanan, B.: A simple fix to mahalanobis distance for improving near-ood detection. arXiv preprint arXiv:2106.09022 (2021)
22. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., 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)
23. Sehwag, V., Chiang, M., Mittal, P.: SSD: A unified framework for self-supervised outlier detection. In: International Conference on Learning Representations (2021)
24. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
25. Su, J., Vargas, D.V., Sakurai, K.: One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation **23**(5), 828–841 (2019)
26. Sun, Y., Ming, Y., Zhu, X., Li, Y.: Out-of-distribution detection with deep nearest neighbors. arXiv preprint arXiv:2204.06507 (2022)
27. Tack, J., Mo, S., Jeong, J., Shin, J.: CSI: Novelty detection via contrastive learning on distributionally shifted instances. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 11839–11852. Curran Associates, Inc. (2020)
28. Tajwar, F., Kumar, A., Xie, S.M., Liang, P.: No true state-of-the-art? ood detection methods are inconsistent across datasets. arXiv preprint arXiv:2109.05554 (2021)
29. Ting, K.M., Witten, I.H.: Issues in stacked generalization. J. Artif. Int. Res. **10**(1), 271–289 (may 1999)
30. Walkowiak, T., Datko, S., Maciejewski, H.: Utilizing local outlier factor for open-set classification in high-dimensional data-case study applied for text documents. In: Proceedings of SAI Intelligent Systems Conference. pp. 408–418. Springer (2019)
31. Wei, X.S., Luo, J.H., Wu, J., Zhou, Z.H.: Selective convolutional descriptor aggregation for fine-grained image retrieval. IEEE Transactions on Image Processing **26**(6), 2868–2881 (2017)
32. Winkens, J., Bunel, R., Roy, A.G., Stanforth, R., Natarajan, V., Ledsam, J.R., MacWilliams, P., Kohli, P., Karthikesalingam, A., Kohl, S., et al.: Contrastive training for improved out-of-distribution detection. arXiv preprint arXiv:2007.05566 (2020)
33. Xingjun, M., Bo, L., Yisen, W., Sarah, M.E., Sudanthi, N.R.W., Michael, E.H., Grant, S., Dawn, S., James, B.: Characterizing adversarial subspaces using local intrinsic dimensionality. In: ICLR (2018)
34. Yuan, X., He, P., Zhu, Q., Li, X.: Adversarial examples: Attacks and defenses for deep learning. IEEE transactions on neural networks and learning systems **30**(9), 2805–2824 (2019)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_41