CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering

SEMESTRAL PROJECT

Pavel Linder

Detecting Out-Of-Distribution Samples with Object Detectors

Department of Computer Science

Supervisor: **Ing. Bc. Radim Špetlík**

May, 2023

Contents

1 Introduction

Artificial Intelligence (AI) has become a topic of great interest among the general public especially in recent times. Machine Learning (ML) models are being widely adopted across various domains to handle a wide range of tasks, and novel applications are being discovered every day. When deploying machine learning models in real-world scenarios, our primary concern is typically centered around the ultimate precision of the predictive outcomes. However, it is equally crucial to take into account the reliability and validity of these predictions. One must assess whether the model's high response is a result of its exposure to comparable data during training or if it is yielding unreliable predictions for unexplored data that was not previously encountered during the training process.

Modern deep learning models can easily produce these overconfident predictions. This issue not only decreases a model's robustness but also raises significant concerns in areas such as medical care, where incorrect diagnoses can result in severe outcomes. Further, it can also question the safety in AI [1]. A new area of research called Out-Of-Distribution (OOD) detection aims to get rid of this vulnerability by determining whether an input is in-distribution (ID) or OOD. [2] [3] [4] [5] [6] [7] By identifying OOD samples, models can decrease the risk of inaccurate predictions, speed up human intervention when necessary, and establish a dependable and secure incorporation of machine learning technologies across a range of domains.

Over the years, extensive research has been carried out on multi-class classification; however, the multi-label task remains an area that has been largely underexplored. The goal of this work is to provide a comprehensive analysis of the different techniques and trends utilized in multi-label OOD detectors, while also categorizing and discussing the various methods employed. Additionally, we offer a concise overview of the multi-class context to show the fundamental concepts. We would see that multi-class and multi-label problems are connected and it makes sense to study both.

We assess state-of-the-art methods for multi-label OOD detection using pre-trained networks and identical methodology. We select MSP [2], JointEnergy[3] and ODIN[8]. All methods have the same evaluation pipeline. Evaluation is performed by two different classification networks on three public datasets. A new comparison of methods is made by selecting datasets where benchmarking was not done yet for all the methods.

The evaluation used two different models trained on MS-COCO[9] and Pascal[10] dataset. For OOD detection, three datasets were used: a subset of ImageNet22k[11], TACO[12], and Textures[13]. JointEnergy performed the best across all three OOD datasets.

2 Background

Multi-label classification Multi-label classification is an instance of supervised learning problem there each input sample can be associated with multiple labels simultaneously. This approach differs from the traditional multi-class classification, which assigns only a single label to each sample. Multi-label classification allows the presence of multiple or even zero labels for a given input. This makes it a more adaptable and comprehensive approach, making it suitable for several real-world scenarios.

Formally, let $\mathcal{X}$ (respectively, $\mathcal{Y}$) denote the input (respectively, output) space of classifier $f : \mathcal{X} \rightarrow \mathbb{R}^{|\mathcal{Y}|}$ trained on samples drawn from distribution $\mathcal{P}$, where $\mathcal{P}$ is a distribution over $\mathcal{X} \times \mathcal{Y}$. Each input sample $x \in \mathcal{X}$ is associated with a set of labels $Y = 1, 2, ..., K$ represented as a binary vector $y = [y_1, y_2, ..., y_K]$, where $y_i = 1$ if input sample x is associated with class i, and 0 otherwise.

To illustrate, consider the example of image classification. In a multi-label setting, an image can have multiple objects or attributes of interest. For instance, an image may contain a cat, a dog, and a tree simultaneously. Instead of assigning a single label to the image, a multi-label classifier would predict a binary vector where the elements corresponding to "cat," "dog," and "tree" are all set to 1.

For OOD detection we will utilize neural networks with a shared feature space to obtain a multi-label output prediction. As opposite to the approach of training disjoint classifiers as proposed in literature [14], implementing the end-to-end training method with a shared feature space is more computationally efficient than training K completely independent models. Multi-label classification models are now mostly trained using this technique, which has been shown to work effectively in various domains [15] [16] [17].

Multi-class classification The difference between multi-label and multi-class classification [18] is determined by the flexibility and granularity of the label assignments. Multi-label classification acknowledges the possibility of multiple labels being equivalently relevant simultaneously, accommodating more complex and diverse scenarios.

Object detection The goal of an object detector extends the task of classification and except classifying the objects present in an image it also provides their precise spatial localization by drawing bounding boxes around them. This allows for accurate identification and tracking of objects of interest. There are several well-known object detectors such as SSD [19], YOLO [20], R-CNN [21] and Mask R-CNN [22].

Out-Of-Distribution detection Out-Of-Distribution (OOD) detection can be formulated as an instance of binary classification. Let $\mathcal{D}_{in}$ denote the marginal distribution of $\mathcal{P}$ over $\mathcal{X}$ which represents the distribution of in-distribution (ID) data. In practise, OOD is frequently characterized by a distribution that emulates the uncertainties that arise during deployment, such as samples from an irrelevant distribution whose label set does not intersect with $\mathcal{Y}$ and thus should not be predicted by the model. We will denote this distribution over $\mathcal{X}$ as $\mathbb{D}_{out}$. For OOD detector G in multi-label setting with classifier f we define a decision function for the binary classification as follows:

$$G(\mathbf{x}, f) = \begin{cases} 0 & \text{if } \mathbf{x} \sim \mathbb{D}_{out} \\ 1 & \text{if } \mathbf{x} \sim \mathbb{D}_{in} \end{cases} \tag{1}$$

Energy function An energy function is a mathematical function that maps a scalar value to a given input. In the context of the paper, the energy function is used to represent the cost or likelihood of a particular label. Formally, the energy function $E(x) : \mathcal{X} \to \mathbb{R}$ maps each point x of an input space to a scalar value called the energy. It assigns a high energy score to OOD samples and a low energy score to in-distribution samples.

Kullback-Leibler (KL) divergence Kullback-Leibler (KL) divergence [23] is a measure of the difference between two probability distributions. KL divergence is defined as the expectation of the logarithmic difference between the model-predicted distribution $q = q_i$ and the reference distribution $p = p_i$. Formally:

$$D_{KL} = (p \parallel q) = \sum_i p_i \log \frac{p_i}{q_i} = \sum_i p_i \log q_i + \sum_i p_i \log p_i = H(p, q) - H(p) \tag{2}$$

3 OOD detection in multi-class setting

In 2015, Nguyen et al. [24] exposed the vulnerability of deep networks. They found that one cause of overconfidence for OOD data is due to the nature of the fast-growing exponential function used in computing softmax probabilities. Small changes to the logits can result in significant changes to the output distribution. The direct correlation between prediction probability and confidence in a softmax distribution is poor.

3.1 MSP detector

Maximum Softmax Probability (MSP) score In 2016, Hendrycks and colleagues [2] took a significant step towards addressing the vulnerabilities of neural networks. The main findings can be concluded as follows. The probability of correctly classified examples is higher than that of misclassified and out-of-distribution examples. Thus, by capturing the statistics of prediction probabilities of correct examples, we can use them to detect whether an example is in error or abnormal. It is important to note that relying solely on softmax probabilities may lead to unreliable outcomes. However, analyzing the statistics of these probabilities is yet effective across various domains, including computer vision. This work has become a baseline for OOD detection in multi-class settings and is still used for benchmarking state-of-the-art methods.

3.2 Out-of-DIstribution detector for Neural networks (ODIN)

Temperature scaling and input pre-processing In 2017, Liang et al. presented another output-based approach known as ODIN [8] (Out-of-DIstribution detector for Neural networks). ODIN is a technique that uses temperature scaling and input pre-processing (adding small perturbations). It aims to effectively separate the softmax probability distributions of in-and-out-of-distribution images. This separation enables more accurate and efficient OOD detection. Temperature scaling involves scaling the logits (inputs to the softmax function) by a temperature parameter T before computing the softmax function. This has the effect of sharpening the softmax output, making it easier to distinguish between in-distribution and out-of-distribution images.

3.3 Mahalanobis distance detector

Mahalanobis distance In 2018, Lee et al. [25] introduced a new feature-based approach for detecting out-of-distribution (OOD) samples, which can be applied to any pre-trained softmax neural classifier without requiring to re-train. The method uses a 'generative' (distance-based) classifier to measure the probability density of test samples on feature spaces of Deep Neural Networks (DNNs). The authors assume that pre-trained low and upper level features can be fitted well by a class-conditional Gaussian distribution. Confidence score is defined using the Mahalanobis distance with respect to the closest class-conditional distribution.

3.4 Energy-based detector

Energy score In 2021, Liu, Wang, et al. [26] presented a new method that uses outputs of the DNNs. Instead of employing softmax scores, as with the MSP approach, they employ energy scores, which better distinguish between in- and out-of-distribution samples. Unlike softmax confidence scores, energy scores are theoretically aligned with the probability density of the inputs and are less susceptible to the overconfidence issue. Energy can be flexibly used as a scoring function for any pre-trained neural classifier, as well as a trainable cost function to explicitly shape the energy surface for OOD detection. Energy score function was introduced in section 2 and for a given input $(\mathbf{x}, y)$ is defined as $E(\mathbf{x}, y) = -f_y(\mathbf{x})$. Without altering the parameterization of the neural network $f(\mathbf{x})$, we can define the **free energy function** $E(\mathbf{x}; f)$ over $\mathbf{x} \in \mathbb{R}^D$ in terms of the denominator of the softmax activation:

$$E(\mathbf{x}, f) = -T * \log \sum_{i}^{K} e^{f_i(\mathbf{x})/T} \tag{3}$$

where K is the number of classes and T is the temperature parameter.

3.5 Fast Out-Of-Distribution Detector (FOOD)

Statistical testing and additional output neuron In 2021, Amit and Levy [27] proposed a Fast Out-Of-Distribution Detector (FOOD) that efficiently detects out-of-distribution (OOD) samples with minimal inference time overhead. The paper presents a DNN model with a final Gaussian layer that models a density function for each class and a rapid OOD detector that does not require OOD samples for training or hyperparameter tuning. The paper uses a log likelihood ratio statistical test and an additional output neuron for OOD detection.

3.6 GradNorm

Categorizing OOD detectors We can categorize the OOD detection methods that have been implemented into two main groups: **output-based** and **feature-based**. Output-based detectors employ the output of neural networks which is then computed using an aggregation function (such as max or sum) and a scoring function to detect OOD instances. E xamples of these methods include ODIN[8], Energy-based[26], and MSP[2]. On the other hand, feature-based detectors utilize the feature space to distinguish between OOD and ID samples. Mahalanobis distance based detection[28] is an example of this method.

Using KL divergence In 2021, Huang, Rui and Geng [29] introduced a novel approach for OOD detection called GradNorm. Their method utilizes information extracted from the **gradient space**. The main idea proposed in this paper is to use the vector norm of gradients, which are backpropagated from the **KL divergence** between the softmax output and a uniform probability distribution, as a means of detecting OOD inputs.

Definition for KL divergence can be find in Equation 2. For this method we set the reference distribution to be uniform $\mathbf{u} = [1/C, 1/C, ..., 1/C] \in \mathbb{R}^C$. The predictive probability distribution is represented by the softmax output. KL divergence then becomes:

$$D_{KL} = (\mathbf{u} \parallel \text{softmax}(f(\mathbf{x}))) = \frac{1}{C} \sum_{c=1}^{K} \log \frac{e^{f_c(\mathbf{x})/T}}{\sum_{j=1}^{C} e^{f_j(\mathbf{x})/T}} - H(\mathbf{u}) \tag{4}$$

where the first term is the cross-entropy loss with a uniform vector u, and the second term is a constant H(u). KL divergence measures how far the predictive distribution is from the uniform distribution. ID data is expected to have a larger KL divergence because the prediction tends to concentrate on one of the ground-truth classes and is therefore less uniformly distributed.

GradNorm score This technique does not directly employ KL divergence, but instead utilizes the gradient vector norm, which is propagated backwards from the KL divergence. The OOD score is then defined as:

$$S(\mathbf{x}) = \|\frac{\partial D_{KL}(\mathbf{u} \parallel \mathrm{softmax}(f(\mathbf{x})))}{\partial \mathbf{w}}\| \tag{5}$$

where $\mathbf{w}$ is is the set of parameters in vector form and $\|.\|$ denotes L_1 norm...

At the end exploring gradient space proved to be useful and this become a new state-of-the-art method.

4 OOD detection in multi-label setting

As mentioned previously, research on detecting Out-Of-Distribution in multi-label classifiers has quite recently just begun, resulting in a limited number of approaches. Most of the methods adopt concepts and ideas from multi-class OOD detection. In this section, we will provide a more detailed overview description of state-of-the-art OOD detectors in multi-label setting.

4.1 JointEnergy detector

Using energy function and summation In 2021, Wang et al. [3] introduced JointEnergy, a simple and effective method for estimating OOD indicator scores by aggregating label-wise energy scores from multiple labels. In multi-label settings, MaxLogit[30] served as the baseline for OOD detection. MaxLogit obtains logits from the neural network and takes the maximum of these logits as a score. JointEnergy introduces two fundamental changes. Firstly, it uses energy scores instead of logits, and secondly, it take the sum across the classes instead of maximum. Liu, Weitang and Wang [26] demonstrated that the energy-based approach can enhance OOD uncertainty estimation for multi-class settings and in this paper, it is proven the efficiency for multi-label settings as well.

Multi-label classifier can be interpreted from energy-based perspective by viewing the logit $f_{y_i}(\mathbf{x})$ of class y_i as an energy function $E_{y_i}(\mathbf{x}, y_i) = -f_{y_i}(\mathbf{x})$. Without changing the parameterization of the neural network f (x), we can express the free energy function $E_{\text{joint}}(\mathbf{x})$ over $\mathbf{x} \in \mathbb{R}^D$: The free energy function is computed by aggregating the label-wise energy scores from multiple labels.

$$E_{y_i}(\mathbf{x}, f) = \log\left(1 + e^{f_{y_i}(\mathbf{x})}\right) \tag{6}$$

$$E_{\text{joint}}(\mathbf{x}) = \sum_{i=1}^{K} -E_{y_i}(\mathbf{x}, f) \tag{7}$$

Label-wise energy $E_{y_i}(\mathbf{x})$ by definition is a negative value, and the aggregation methods output a positive value by negation. This aligns with the convention that a larger score indicates in-distribution and vice versa.

The authors emphasize the importance of both the **scoring function**, **aggregation function** as well as **their compatibility** to achieve an accurate OOD detection. They proved that using summation over the labels makes the estimation much stronger than just taking the maximum. This can be seen in Figure 1

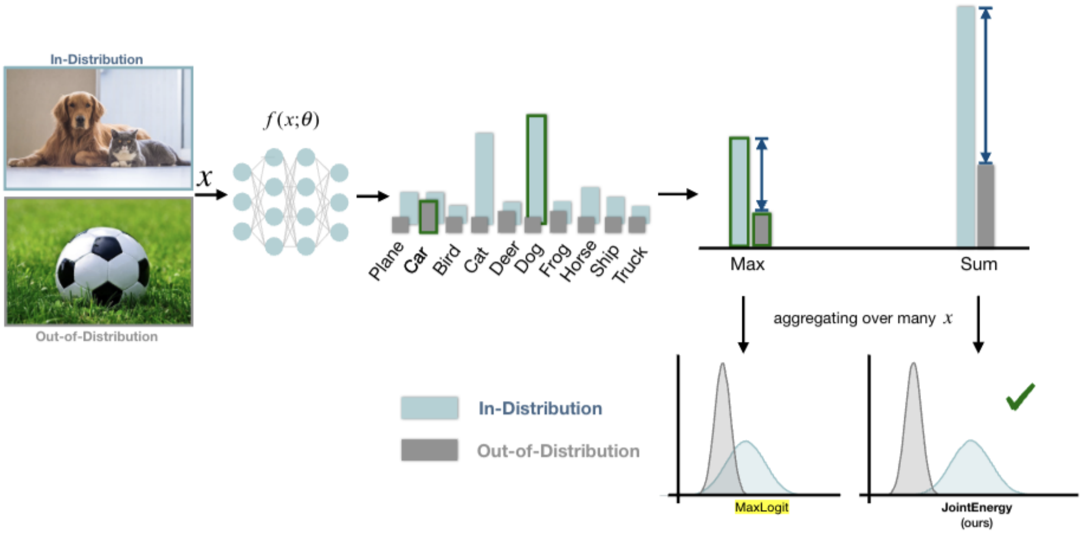

Figure 1: OOD scores are either the maximum-valued score (denoted by green outlines) or the sum of all scores. Taking the sum results in a larger difference in scores and more separation between in-distribution and OOD inputs (denoted by red lines), resulting in better OOD detection. Plots in the bottom right depict the probability densities of MaxLogit[30] versus JointEnergy[3]

4.2 YolOOD

Using object detector In 2022, Zolfi et al. presented YolOOD [5], a novel method for out-of-distribution (OOD) detection in the multi-label classification task. The main contribution of this paper is the utilization of concepts from object detection to perform OOD detection. The authors convert a regular object detection model Yolo [20] into an image classifier with inherent OOD detection capabilities with just minor changes. Object detectors have an inherent ability to distinguish between objects of interest (in-distribution data) and irrelevant objects (OOD data) making them perfect candidates for fulfilling the goals of the OOD detection task. Yolo object detector predicts objectness score and probability class scores for every cell in the grid.

Network architecture Yolo has a backbone network and three detection heads that process features at different scales. YolOOD has an adapted last layer of the detector to produce the image-wise scores and objectness scores. An overview of the network's pipeline can be seen in Figure 2.

Yolo Objectness score The training of object detectors involves passive negative learning, where unlabeled data in training images helps the model discard irrelevant objects. YOLO uses objectness scores to achieve this. The YOLO objectness score is a single value that represents the model's confidence that the bounding box contains an object. In YolOOD, the objectness score is used as one of the components of the custom loss function devised to train the model. The objectness score for a candidate (cell in the grid) c is:

$$c_{\text{obj}}(i,j) = \begin{cases} 1 & \text{if } \varphi_u \wedge \varphi_s \wedge \varphi_l \wedge \varphi_r \\ 0 & \text{if otherwise} \end{cases} \tag{8}$$

$$\begin{aligned} \varphi_u = i \geq x_{\text{center}} - p * \frac{W'}{2}, \varphi_l = j \geq y_{\text{center}} - p * \frac{H'}{2} \\ \varphi_d = i \leq x_{\text{center}} + p * \frac{W'}{2}, \varphi_r = j \leq y_{\text{center}} + p * \frac{H'}{2} \end{aligned} \tag{9}$$

where $W(H)$ is the width (height) of the image respectively and p is the portion of grid cells relative to the grid's size, with regard to the object's center.

Class score Class score for a class $n \in 1, \dots, N_c$ in a specific candidate (cell in the grid) c is:

$$c_{\text{cls n}} = \begin{cases} 1 & \text{if class n is in cell} \\ 0 & \text{if otherwise} \end{cases} \tag{10}$$

YolOOD score Finally the OOD detection score is:

$$\text{YolOOD}(\mathbf{x}) = \max_{c' \in f(\mathbf{x})} (\sigma(c'_{obj} * \max_{i \in 1,\dots,N_c} (\sigma(c'_{\text{cls i}})))) \tag{11}$$

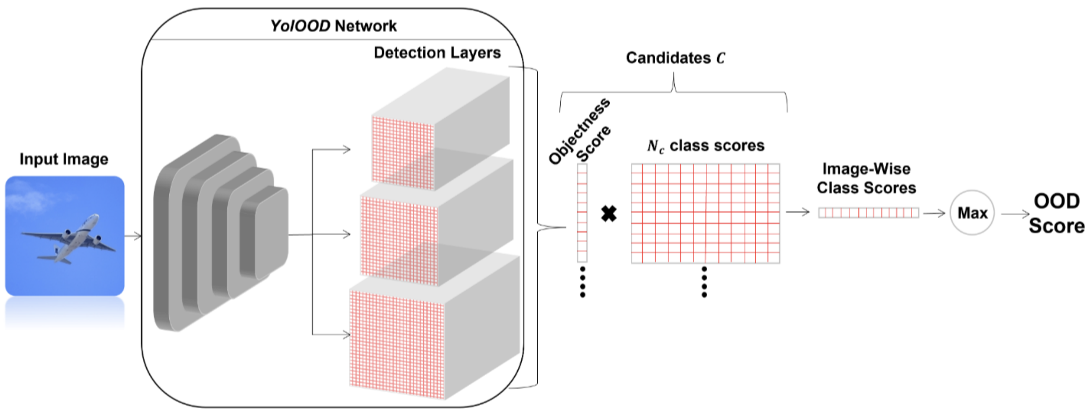

Figure 2: YolOOD's pipeline.

5 Datasets

MS-COCO [9] The MS-COCO dataset comprises images of everyday scenes with common objects to advance object recognition within the context of scene understanding. There are 82,783 training, 40,504 validation, and 40,775 testing images, all of which contain 80 common object categories. This dataset is commonly used for state-of-the-art benchmarking in object recognition and classification including multi-label classification.

Figure 3: Examples of images from dataset MS-COCO and Pascal-VOC. MS-COCO: Picture containing objects from categories ‘person’, ‘tie’, ‘bowl’, ‘chair’, ‘dining table’ and ‘clock’.

VOC-Pascal [10] It is good for multi-label classification because it includes a variety of object categories, allowing for the evaluation of algorithms that can recognize and detect multiple objects in an image. PASCAL-VOC has 22,531 images, which are divided into 20 classes.

ImageNet [11] ImageNet is a large-scale hierarchical image database built upon the WordNet structure. Compared to small image datasets like Caltech101/256, MSRC, and PASCAL, ImageNet is much larger in scale and diversity, offering 20× the number of categories and 100× the number of total images. ImageNet is often use not only for benchmarking for the variety of image but also for pre-trained models.

Figure 4: Examples of images with label annotations from dataset TACO and Textures.

TACO [12] The TACO dataset is designed for litter detection and segmentation with 1500 high-resolution images and 4784 annotations. Litter detection is a challenging problem for multiple reasons so this dataset can be use to show the robustness of classifiers.

6 Experiments

In this section, we evaluate some of the existing state-of-the-art methods for multi-label OOD detection on public datasets in order to assess how they compare on identical settings. To compare state-of-the-art methods, we will be using pre-trained networks with identical weights and experimentation methodology.

To provide a nice overview of the methods we select **MSP** as a baseline of multi-label OOD detection and **JointEnergy** with **ODIN** for state-of-the-art methods. All three methods were implemented to have the same evaluation pipeline. As a bonus, **GradNorm** will be converted for multi-label classifiers.

The evaluation was performed on datasets that have been utilized across multiple papers about multi-label OOD detection. As an outcome a new comparison of the methods will be made. The labels of the ID datasets are different from the labels of the OOD datasets in this evaluation to reduce bias.

6.1 Experimental setup

ID datasets We will use two multi-label datasets: MS-COCO[9] and PASCAL-VOC[10]. These datasets are used to train multi-label classifier and for producing ID reliability scores.

OOD datasets To assess the models trained on the ID datasets, we adopt the same approach as described in [3]. We select a subset of ImageNet[11] dataset, the whole Textures[13] dataset and the whole TACO[12] dataset. For ImageNet, we use the same set of 20 classes as in [3] to have different labels than ID datasets. Textures dataset was used both in [3] and [29]. Finally, TACO dataset was used in [5] and other papers. By this selection we should cover different class domains and provide new insights.

Classification model training We deploy two classifiers sourced from the study by Wang et al.[3] featuring DenseNet-121 architecture. These classifiers were initially trained on ImageNet-1K and subsequently fine-tuned through the utilization of sigmoid function. Data was augmented with random crops and flips. Achieved mAP is 87.51% for PASCAL-VOC, 73.83% for MS-COCO.

6.2 OOD detection

MSP, JointEnergy, ODIN We use the public implementation of original authors to produce OOD scores.

GradNorm Since GradNorm has yet to be adopted for multi-label classifiers, it is necessary to adjust the original implementation. For our implementation, we have set the parameters in the same fashion as the authors of GradNorm did, with temperature T being equal to 1 and using the gradients weights of the last layer only. Although binary cross-entropy loss with logits is used for the multi-label classifier, it is still noteworthy that we compute cross-entropy loss here. The multi-label classifier does not utilize softmax on the network's output, but rather employs the sigmoid function to generate a K-long vector of class probabilities. This vector is exactly what we need to calculate GradNorm, and hence, we can utilize it in a way displayed below.

To generate GradNorm scores, one must execute the following steps:

- use the classifier to produce the predicted probability distribution over classes $[K \times 1]$ vector
- divide the probability distribution by temperature T
- compute gradients of KL divergence as average the derivative of the cross-entropy loss for all labels
- perform backward pass of the network
- get the last layer's output and compute the first norm

6.3 Evaluation settings

Evaluation pipeline For every classifier, scores are computed from In-Distribution (ID) datasets as **inScores** and scores for Out-Of-Distribution (OOD) datasets as **outScores**. OOD detection is a binary classification so it requires the generation of ground truth labels. In the case of *inScores*, the ground truth is set to 0 since we know that they contain labels which they were trained for. Alternatively, for *outScores*, the ground truth is set to 1. Metrics are then computed from both *inScores* and *outScores* along with their corresponding ground truth labels.

Metrics The evaluation of performance is executed by means of metrics that are widely utilized in the field of Out-of-Distribution (OOD) detection. These metrics include:

(a) FPR95 - the false positive rate of OOD samples when the true positive rate is at 95%

(b) AuROC - the area beneath the receiver operating characteristic curve

(c) AuPR - the area beneath the precision-recall curve

			MS-COCO			PASCAL	
			FPR95	**AUROC**	**AUPR**		
	OOD score		↓	↑	↑		
ImageNet	MSP	79.69	74.74	85.40	74.04	79.34	72.58
	ODIN	42.21	87.72	91.43	46.23	85.23	90.23
	JointEnergy	**33.34**	**92.73**	**96.26**	40.98	91.12	86.36
Textures	MSP	47.11	87.55	97.57	39.82	92.41	95.91
	ODIN	15.33	96.10	98.54	16.02	95.32	98.14
	JointEnergy	**9.7**	97.28	**99.56**	10.83	**97.39**	98.62
TACO	MSP	55.93	86.05	99.35	70.40	82.81	99.16
	ODIN	29.23	92.34	97.23	27.54	93.43	97.80
	JointEnergy	20.20	95.82	99.82	**11.80**	**97.51**	**99.90**

Table 1: Quantitative comparison of MSP, ODIN and JointEnergy.

6.4 Results

The evaluation was done with two different models (trained on MS-COCO and Pascal dataset). Also for the OOD detection we use three datasets (subset of ImageNet22k, TACO and Textures). In Table 1 you can see the results over all three OOD datasets for JointEnergy, MSP and ODIN. We see that JointEnergy has the best performance across all OOD datasets.

The results from GradNorm implementation with the same setting as describe above were opposing the expectations. ID image samples produces lesser scores than OOD image sample which is in contrary to the study by [29]. There must be more experimantation done with this but we can state that the direct translation of the method does not work.

7 Conclusion

In this report, we provided a comprehensive overview of Out-Of-Distribution (OOD) detectors along with their timelines. We thoroughly discussed the main concepts behind each of the methods and formalized the scoring functions. Although there are numerous approaches for OOD detection in multi-class classifiers, the multi-label setting has not yet been fully explored. We have observed the implementation of existing methods in this domain and have also noted the emergence of new approaches such as YolOOD, which have produced interesting results as well. Some of the methods were implemented and quantitative results were produced. This could be used for benchmarking on new set of datasets. We attempted to convert GradNorm directly to the multi-label setting, but discovered that it does not produce a valid classifier.

8 Future Work

There is a lot of potential areas to explore. Existing approaches could be explored more and adapted for multi-label settings. As suggested by [5], using a different dataset that contains pixel-level annotations for training object detector will allow to learn accurate representations without irrelevant areas. Another direction is to evaluate on large-scale dataset that mimic real-world settings. [30, 7]

References

[1] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. June 2016.

[2] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. October 2016.

[3] Haoran Wang, Weitang Liu, Alex Bocchieri, and Yixuan Li. Can multi-label classification networks know what they don't know? September 2021.

[4] Sina Mohseni, Arash Vahdat, and Jay Yadawa. Shifting transformation learning for out-of-distribution detection. June 2021.

[5] Alon Zolfi, Guy Amit, Amit Baras, Satoru Koda, Ikuya Morikawa, Yuval Elovici, and Asaf Shabtai. YolOOD: Utilizing object detection concepts for out-of-distribution detection. December 2022.

[6] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. October 2021.

[7] Rui Huang and Yixuan Li. MOS: Towards scaling out-of-distribution detection for large semantic space. May 2021.

[8] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, June 2017.

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context, February 2015.

[10] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision*, 111(1):98–136, January 2015.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009.

[12] Pedro F. Proença and Pedro Simões. TACO: Trash Annotations in Context for Litter Detection, March 2020.

[13] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing Textures in the Wild, November 2013.

[14] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification. *International Journal of Data Warehousing and Mining*, 3(3):1–13, July 2007.

[15] Wenjie Zhang, Junchi Yan, Xiangfeng Wang, and Hongyuan Zha. Deep extreme multi-label learning. April 2017.

[16] Konstantinos Trohidis, Grigorios Tsoumakas, George Kalliris, and Ioannis Vlahavas. Multi-label classification of music by emotion. *EURASIP Journal on Audio, Speech, and Music Processing*, 2011(1):4, December 2011.

[17] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. May 2017.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, June 2021.

[19] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single Shot MultiBox Detector. volume 9905, pages 21–37. 2016.

[20] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. YOLOv4: Optimal Speed and Accuracy of Object Detection, April 2020.

[21] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, October 2014.

[22] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN, January 2018.

[23] Richard Dykstra. Kullback–Leibler Information. July 2005.

[24] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images, April 2015.

[25] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks, October 2018.

[26] Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based Out-of-distribution Detection, April 2021.

[27] Guy Amit, Moshe Levy, Ishai Rosenberg, Asaf Shabtai, and Yuval Elovici. FOOD: Fast out-of-distribution detector. August 2020.

[28] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. July 2018.

[29] Rui Huang, Andrew Geng, and Yixuan Li. On the Importance of Gradients for Detecting Distributional Shifts in the Wild, October 2021.

[30] Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joe Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling Out-of-Distribution Detection for Real-World Settings, May 2022.