Proceedings of the International Conference on
Formal Methods and Foundations of Artificial Intelligence
Eszterházy Károly Catholic University
Eger, Hungary, June 5–7, 2025
pp. 129–139
DOI: 10.17048/fmfai.2025.129

Efficiency testing of openset learning methods in image classification

László Kovács, Enikő Palencsár, Péter Bán

University of Miskolc, Institute of Computer Science
{laszlo.kovacs,palencsar.eniko}@student.uni-miskolc.hu
bn.peter.hun@gmail.com

Abstract. The problem of detecting untrained categories may cause efficiency degradation in many application areas, because the real-word domains are usually dynamic, or the available data set may be incomplete. Despite the relatively high cost of related misclassification errors, the field of openset learning is an underinvestigated domain in machine learning. The main goal of this paper is to investigate the efficiency of current technologies for the openset learning problem on a standard benchmark image dataset. As the results of the performed comparison tests show that the widely proposed standard methods do not provide good results, in many cases the hybrid methods can dominate the usual approaches. In the paper, we present a novel extended threshold method that provides better accuracies than the usual benchmark methods.

Keywords: image classification, CNN neural networks, openset learning problem

AMS Subject Classification: 68T07

1. Introduction

Neural networks are now the dominant technologies in complex classification and regression tasks. The neural network as a universal approximator applies a complex network of elementary functions to predict the function values at arbitrary positions. According to the General Approximation Theorem, a feedforward neural network with a single hidden layer containing a finite number of neurons and using a continuous activation function increasing monotonically can approximate any continuous function [6]. The model construction process to adjust the weight values of the neural network is optimized with a training process. The usual backprop-

agation optimization method adapts the weight values to the available supervised training dataset.

In the case of classification problems, the neurons in the output layer correspond to the different categories in the dataset and the neuron with the highest output value determines the winner category [1]. For example, in the case of image classification, the set of image categories is fixed and the training set should cover all categories in a uniform way. The problem of unbalanced class distribution is a widely investigated problem [7] as it can cause efficiency degeneration due to difficulty learning the limits of the decision or to misleading performance metrics. Thus, one of the main goals of the data preparation phase is to build a well-balanced training data set for the predefined categories.

Usually, it is hard work to meet this kind of requirement, or sometimes it is an impossible task. Our investigation focuses on the domain of related open-set learning problems [2]. Unlike the traditional situation, the test set may contain cases that do not belong to any of the categories presented in the training set. The test data set may contain instances of previously unseen classes. The key challenge here is to detect these unseen cases; the neural network should recognize that the input differs significantly from any trained categories.

The problem of detecting untrained categories may occur in many areas of application, because the real word domains are usually dynamic, or the available data set may be incomplete [5]. We can highlight the following application domains where the risk of incomplete training set is relatively high:

- image object classification, where the image contains un-trained objects widely used technology in medical diagnosis or autonomous driving,
- intruder detection,
- fault detection in industrial monitoring,
- sentiment analysis.

Despite the relatively high cost of related misclassification errors, the field of openset learning is an underinvestigated domain in machine learning. The main goal of this paper is to investigate the efficiency of current technologies for the openset learning problem on a standard benchmark image dataset.

The results of the performed comparison tests show that the widely proposed standard methods do not provide good results, in many cases the hybrid methods can dominate the usual approaches. In the paper, we also present a novel extended threshold method that provides accuracies that are better than the usual benchmark methods.

2. Related methods

The problem domain of openset learning is similar in many aspects to some other problem domains related to incomplete training datasets. One of these fields is

the one-class classification problem [15]. In the case of one-class classification, the training set contains only a single class (only positive cases), and the main goal of the prediction is to determine whether the test object belongs to this class or not [5].

The one-shot learning domain [14] refers to the case when the training set contains only a single example for each existing class. Here, the generated model should provide an optimal generalization based on a single element per class. The training process cannot memorize the common features found in the different instances of the class, it should discover the characteristic features which can be used to distinguish the different classes.

In zero-shot learning, the generated model is not based on instances, but on some available metadata, semantic information on the different classes [16]. The main challenge in this problem domain is the efficient integration of the different multi-model metadata information items.

In the investigated openset learning task, the problem domain can be characterized by the following properties [12]:

- The applied training set does not cover all classes possibly found in the production data.
- The generated model should correctly recognize all classes found in the training set.
- The method should identify the classes not contained in the training set as an outlier or an 'unknown' class.
- No additional semantic information is provided on the classes; the model is inferred only from the available instances.

In the literature, we can find several approaches to deal with this domain of open-set learning problem. The main methods are summarized in the following table.

- Threshold-based category acceptance. We apply the usual multicategory classifier neural network built on the training set. Using the softmax activation function, the output values represent the probability distribution across the different classes. If the maximum output value is below a certain threshold (no clear winner category), the test image is assigned as an outlier [4].
- The OpenMax method is a special variant of the threshold approach. It applies a Weibull distribution to involve the probability of the 'unknown' class. [18]
- Distance-based approach where the method is based on the concept of locality. If the new item is far from any training items, the tested item can be considered an outlier, unknown class [12].

- Development of a MLP neural network for similarity regression, where the similarity shows the distance of the test image to the trained categories. If the similarity is below a given threshold, the test image belongs to the new category.
- Application of generative models, autoencoder neural networks to predict the membership similarity to the known categories. Separate autoencoder neural networks are constructed for the different categories. Taking the test image, the engine generates the autoencoder output for every class. If the maximum similarity between the input and output images is below a threshold, the image is classified as an outlier.
- Outlier exposure method, where the initial training set is extended with noisy outlier data that are labeled 'unknown' class. Although this method improves the robustness of the model, noisy extension usually does not cover all possible external cases [17].

The most widely used approach for the openset problem is the OpenMax method introduced in [2] from 2016. The method first takes the activation outputs of all known classes for all input in the training set. Then it calculates the mean activation vector for each class. In the next step, the distances between the mean vector and the single activation vectors are calculated, then the top k elements with largest distances are selected, and using these values a Weibull distribution is fit for the distinct classes.

$$P(d \mid \lambda_c, k_c) = 1 - \exp\left(-\left(\frac{d}{\lambda_c}\right)^{k_c}\right)$$

For an input test item, we first calculate the standard activation vector and distance values. In the next step, using the Weibull distribution weights, a revised activation vector is calculated. The formula for the Weibull weights:

$$w_{s(i)}(x) = 1 - a(i) \exp\left(-\left(\frac{d}{\lambda_{s(i)}}\right)^{k_{s(i)}}\right)$$

Next, we take an additional activation function for the unknown class. In the last step, we apply the softmax layer for this extended revised activation vector, which also contains a probability value for the unknown class.

The Isolation Forest method [3] applies a space segmentation algorithm to isolate outlier elements in the data space. The Isolation Forest architecture is built up from more random isolation trees. For each node in the isolation tree, a random dimension (attribute) is assigned, and a binary split procedure is performed. The split procedure ends if the size of the corresponding subtree is below a given threshold. Using this methodology for the entire forest, we can calculate an average depth for each item in the data set. Those items are considered outliers or unknown cases, where this average depth is a small value.

Isolation Forests were introduced by Liu, Ting, and Zhou in 2008 [10] as an unsupervised method for fast anomaly detection. The algorithm is based on two key assumptions:

- anomalies represent a minority in the dataset, and
- their feature values differ significantly from those of the majority [11].

For each node in the tree:

- Randomly select a feature $f \in \{1, \ldots, d\}$
- Randomly select a split value $s \in [\min(X_f), \max(X_f)]$
- Partition data:
  - Left child: $X_f < s$
  - Right child: $X_f \geq s$
- Recursively apply until:
  - Node has only one instance, or
  - Tree reaches **maximum depth** $\lceil \log_2(n) \rceil$

Figure 1. Algorithm of building the isolation tree.

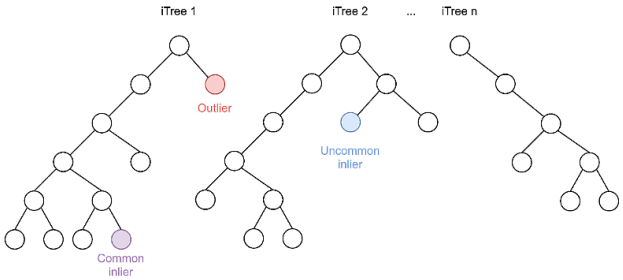

Figure 2. The structure of the isolation tree.

3. Proposed approaches

In our investigation, we propose an extended variant of the threshold-based approach. Usually, the existing methods perform a similarity check in the input space or in the output space. Our assumption is that both spaces can provide additional information about outlier items. Thus, the proposed method applies two class similarity measures for outlier detection. The first component calculates the similarity of the softmax output probability values. The method calculates the differences between the probability p_1 of the winner class and the probability p_2 of the second-best class.

$$w_1(x) = p_1(x) - p_2(x)$$

In the formula, the symbol x denotes the current element to be tested. If the model is not sure which class is the winner, this measure $w_1(c)$ is small.

The second measure is used to locate outliers in the feature space of the objects. Usually, the input space contains the required feature vectors, but in the case of images, the input matrix provides only a pixel-level description, which representation is far from the semantic-level feature vector content. To provide a better representation of semantical features, the first method version utilizes the input vectors of the last Dense layer classifier module of the CNN neural network. Thus, the output of the flattening layer is used to describe the image features.

For a given element x, we define

$$w_{2a}(x, c) = \sum_{i \in c} \exp(-d(x, x_i))$$

where the summation runs over the objects of class c. The symbol $d()$ denotes the distance between the tested item and a single object in class c. The large $w_{2a}(x, c)$ values mean that the item is near the elements of class c, and the small $w_{2a}(x, c)$ values indicate an outlier element. The proposed method calculates the weight values both for the winner class and the complement classes. In the case of outliers, where the class prediction is incorrect, there is no significant difference between the values for the winner and rejected classes.

In addition to the flattening layer method, we also applied the perceptual hashing approach. The perceptual hashing method is used to generate content-based fingerprints of images. The calculation of the description vector consists of the following steps:

- Standardization of the image, conversion to predefined size, and grayscale colormap.
- Application of a discrete cosine transform to detect the internal description using components of different frequency.
- Extraction of the most important components
- Calculation of the hash value for the selected components.

The final decision on the outlier status is calculated with the following method.

1. Generating the predicted class ($c1$) and the second-best class $c2$ for the input object x using the trained CNN neural network.
2. Calculation of $w_1(x)$.
3. If $w_1(x) < \alpha_1$ then x is an unknown outlier class.
4. Calculation of $w_2(x, c1)$.
5. If the value of w_2 is above a threshold $alpha_2$, then x is an unknown outlier class.

In addition to the extended threshold method, we have also tested an ensemble CNN version classifier, where we built up a separate classifier neural network for all classes in the data set. Each of the classifiers works as a binary classifier related to one of the classes. If the predicted class probability is below a threshold in each class, the test item will be predicted as an unknown class.

Considering the members of the ensemble, we investigated more variants of the output vector representation forms, all related to some type of layer of the CNN network. Our tests involved the following feature vector variants:

- Flattened output of convolutional layers, where a PCA reduction concept was applied to have a moderate vector size.
- Output of the fully connected layer in the MLP module.
- Logits of the output layer in the MLP unit.
- Softmax of logits.

As this architecture presented the weakest accuracy in the preliminary tests (more than 22% lower accuracy than the extended threshold method), we decided to eliminate it from the group of final candidate methods.

The third proposed variant was the application of an isolation forest architecture. This structure is a special variant of the random forest architecture, where the main goal is to efficiently locate outlier nodes. The method will partition the item into disjoint leaf nodes. If the size of the container nodes is below a threshold, the element is considered as an outlier.

In our investigation, we adapted the Isolation Tree method to the outlier detection of images, and we proposed two variants:

1. The output of the convolution module was used as the vector of attributes of the object.
2. The softmax layer output was used as the feature vector.

4. Experimental evaluation

4.1. Test environment

The main goal of the tests performed was to evaluate the proposed outlier detection methods and compare the accuracy levels achieved with a benchmark method.

In the tests, the following methods were involved:

- OpenMax algorithm as benchmark method
- Isolation forest
- Extended threshold method

The test system was implemented in Python Keras-Tensorflow framework using the Colab development environment.

For the tests, we applied the following three benchmark image classification datasets:

- CIFAR-10 [8]
- MNIST-10 [9]
- COIL-20 [13].

The CIFAR-10 benchmark dataset contains 60000 images (50000 for training and 10000 for test) of small resolution (32*32). The images belong to 10 categories. In the tests, we created reduced training sets to exclude instances of some selected categories. In the test dataset, we assign these instances to a new common category. From the point of view of category prediction in the test phase, the training dataset contains only positive examples, having only "known known" and "unknown unknown" classes [5].

The original CIFAR-10 dataset includes items from 10 classes, but we selected 3 as unknown classes, thus the CNN neural network models involved in the experiments were trained only on 7 known classes.Thus the training dataset contained 35000 images, Image resolution is (32, 32, 3). For the tests, we used all classes, the test set contained 10000 images Thus nearly 30% of the test items belonged to the unknown category.

The MNIST-10 dataset is a key benchmark in machine learning and computer vision, specifically for image classification tasks. The dataset consists of a large collection of 70,000 grayscale images of handwritten digits. The images are encoded into a pixel matrix of size (28 , 28, 1). Its primary role is to provide a standardized, simple dataset for training and testing various image classification algorithms, from classic machine learning models to complex deep learning architectures. .

The Columbia Object Image Library (COIL-20) is a well-known dataset in the field of computer vision. It consists of 1,440 grayscale images of 20 different objects. In our tests, we included only 10 categories. The images were created by placing each object on a turntable and capturing 72 images at 5-degree intervals as it was rotated through 360 degrees. This setup provides a comprehensive set of images for each object, showing it from a wide range of poses and angles.

4.2. Test results

In the efficiency tests, we measured the following accuracy values:

- T1: Validation accuracy in training of the baseline CNN model with 7 classes;
- T2: Test accuracy with 7 classes on the trained baseline CNN model;
- T3: Test accuracy with 10 classes on the trained baseline CNN model;
- T4: Test accuracy with 10 classes using the baseline OpenMax NN model;

- T5: Test accuracy with 10 classes on the proposed Extended threshold NN model;
- T6: Test accuracy with 10 classes on the proposed Isolation Forest NN model using convolution layer output;
- T7: Test accuracy with 10 classes on the proposed Isolation Forest NN model using softmax layer output.

In the tests, we performed five measurements and calculated the average and standard deviation aggregations. The resulting aggregation values are summarized in Table 1. The aggregated values are based on 5 measurements. In the first row, we see the validation accuracy at the end of the training process. We involved only 7 classes into the training, the members from 3 classes were removed from the training set. The output layer of the constructed neural network was able to recognize only 7 categories. The second row shows the accuracy of the test data set with 7 classes. In the third row, the values show the test accuracy when the test dataset contained all 10 classes. The fourth row shows the test accuracy achieved using the OpenMax method using 10 classes in the test. The fifth row relates to the Extended threshold method, while the sixth row is for the results of the Isolation tree method. All values in the table are given in percentage units.

Table 1. Results of the comparison tests.

Method measure	CIFAR avg	CIFAR stdev	MNIST avg	MNIST stdev	COIL avg	COIL stdev
T1	91.1	0.94	99.5	0.07	99.1	0.68
T2	71.5	0.54	98.8	0.41	98.7	1.56
T3	49.8	0.35	68.9	0.39	82.7	0.89
T4	48.7	0.56	69.5	0.04	82.8	0.34
T5	50.8	0.43	74.9	0.89	82.9	3.2
T6	49.0	1.94	47.8	3.12	82.3	0.24
T7	64.1	1.92	78.6	1.71	81.9	0.74

As we can see in the result table, we experienced an unexpected weak result in the case of the baseline OpenMax method. The OpenMax method achieved very similar accuracies as the baseline neural network, there was no significant improvement. We remark that we did not perform a hyperparameter tuning for the OpenMax method in our tests.

In the tests on the Extended threshold method, the proposed method consistently yielded better results than the baseline CNN neural network or the OpenMax method. The improvement is 1% for the CIFAR and 6% for the MNIST datasets. In each individual test sample, the extended threshold dominated both methods mentioned above.

The tests on the Isolation Forest method has clearly shown, that from the variants of the methods tested, the isolation tree with softmax output provided

the best result. Similarly to the Extended threshold approach, the isolation tree method dominated the OpenMax engine as well. In comparison of the different feature vector variants for the isolation tree, only the softmax variant dominated the Extended threshold method.

5. Conclusion

The main goal of the presented work was to investigate the efficiency of the detection of unknown classes in the image classification problem. The situation, when the test dataset contains instances of such classes which were not present in the training set, may cause significant efficiency degradation. The methods to cope with this kind of problem are investigated under the umbrella term openset learning.

In the paper, we present the key solution approaches and also introduce two proposed method variants: Extended threshold method and the Isolation Tree method. For the tests, we used three widely popular datasets: CIFAR-10, MNIST-10, and COIL-20. The CIFAR-10 dataset contains a large amount of low-quality, small images. As can be expected, this dataset is a hard target for the open-set learning problem.

Based on the test results,we can summarize our experiences in the following points:

- The proposed Isolation Forest with softmax feature representation yielded the highest accuracy.
- The other proposed method, Extended Threshold approach secured second place in the competition.
- The baseline OpenMax outlier detection method produced results comparable to the baseline ANN neural network.
- The test results clearly demonstrate that the presence of unseen categories significantly degrades accuracy.

These experiences and test results with low accuracy values show that the detection of unknown classes is a hard problem, the known methods can provide only a slightly improvement. The further optimization of the proposed method is the next key step in our investigation.

References

[1] G. Asadollahfardi: *Water Quality Management: Assessment and Interpretation*, in: Artificial neural network, Netherlands: Springer, 2014, pp. 77–91.

[2] A. Bendale, T. E. Boult.: *Towards open set deep networks*, in: Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE, 2016.

[3] M. M. Diniz, A. Rocha: *Open-set deepfake detection to fight the unknown*, in: International Conference on Acoustics, Speech and Signal Processing, IEEE, 2024.

[4] e. a. Fang Zhen: *Learning bounds for open-set learning*, in: International conference on machine learning, IEEE, 2021.

[5] C. Geng, S.-j. Huang, S. Chen: *Recent advances in open set recognition: A survey*, IEEE transactions on pattern analysis and machine intelligence 43.10 (2020), pp. 3614–3631.

[6] K. Hornik, M. Stinchcombe, H. White: *Multilayer feedforward networks are universal approximators*, Neural networks 2.5 (1989), pp. 359–366.

[7] J. M. Johnson, T. M. Khoshgoftaar: *Survey on deep learning with class imbalance*, Journal of big data 6.1 (2019), pp. 1–54.

[8] A. Krizhevsky, V. Nair, G. Hinton.: *The CIFAR-10 dataset*, https://www.cs.toronto.edu/~kriz/cifar.html, Accessed: 2025-07-15, 2009.

[9] Y. LeCun, C. Cortes, C. Burges: *MNIST handwritten digit database*, https://www.kaggle.com/datasets/hojjatk/mnist-dataset, Accessed: 2025-09-23, 2010.

[10] F. Liu, K. M. T. Tony, Z.-H. Zhou: *Isolation forestv*, in: International Conference on data mining, IEEE, 2008.

[11] F. T. Liua, K. M. Ting, Z. H. Zhou: *Isolation-Based Anomaly Detection*, ACM Trans. Knowl. Discov. Data 6.1 (2012), Article 3.

[12] A. Mahdavi, M. Carvalho: *A survey on open set recognition*, in: Fourth International Conference on Artificial Intelligence and Knowledge Engineering, IEEE, 2021.

[13] S. A. Nene, S. K. Nayar, H. Murase: *Columbia Object Image Library (COIL-20)*, https://www.kaggle.com/datasets/yupanliu999/coil-20, Accessed: 2025-09-23, 1996.

[14] M. H. Saad, A. E. Salman.: *A plant disease classification using one-shot learning technique with field images*, Multimedia Tools and Applications 83.20 (2024), pp. 58935–58960.

[15] e. a. Strani Lorenzo: *One class classification (class modelling): State of the art and perspectives*, TrAC Trends in Analytical Chemistry 183 (2020), p. 118117.

[16] e. a. Wang Wei: *A survey of zero-shot learning: Settings, methods, and applications*, ACM Transactions on Intelligent Systems and Technology 10.2 (2019), pp. 1–37.

[17] J. Xu, M. Kovatsch, S. Lucia: *Learning placeholders for open-set recognition*, in: International Conference on Industrial Informatics (INDIN), IEEE, 2021.

[18] D.-W. Zhou, H.-J. Ye, D.-C. Zhan: *Learning placeholders for open-set recognition*, in: IEEE/CVF conference on computer vision and pattern recognition, IEEE, 2021.