This CVPR Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Exploiting Adversarial Learning and Topology Augmentation for Open-Set Visual Recognition

Rosa Zuccarà
rosa.zuccara@phd.unict.it

Georgia Fargetta
georgia.fargetta@unict.it

Alessandro Ortis
alessandro.ortis@unict.it

Sebastiano Battiato
Department of Mathematics and Computer Science,
University of Catania
sebastiano.battiato@unict.it

Abstract

This paper proposes a novel approach that introduces a dedicated unknown class, inspired by recent studies on adversarial machine learning images, that are completely unrecognizable to humans yet are confidently misclassified by CNNs as objects. Our method seeks to improve the class cohesion in the feature space, guiding the model to manage the vector space better. We demonstrate that using unknown and unseen images leads to more effective management of the feature space and significantly improves recognition performance in the presence of unknown inputs. Initially, we evaluate the distribution of a C-class subset from the `MNIST` and `ImageNet` dataset, investigating the single accuracy of each class, and using t-SNE visualizations to assess how these classes are arranged in the feature space, for a closed-set classification. Then, we extend our approach to a $C+1$ classification scenario. Indeed, the unknown class is constructed using a customized method that integrates NEAT, a technique for neural network evolution, and defines a novel fitness function for this problem, called `f_NEAT`. Then, the synthetic unknown class is added to the test set of images belonging to unseen classes, within an open-set recognition context. The results are especially relevant for real-world applications, such as open-set biometric authentication in an adversarial environment, an area we intend to explore in our future work.

1. Introduction

Recognition in the real world is inherently open-set [1], requiring a model not only to classify known categories but also identify and manage previously unseen instances. However, conventional classification models struggle in open-set scenarios, where the classifier is unaware of the existence of unknown inputs. Traditional training methodologies impose several simplifications, such as assuming a fixed, closed set of categories and adopting predefined network topologies. As a result, deep models learn a feature space optimized solely for projecting known training samples, often focusing on maximizing average accuracy rather than maximizing the accuracy of individual known classes. This limitation prevents the learned decision boundaries from fully representing the real data distribution, leading to several issues, including inefficient feature space utilization and the emergence of adversarial regions. Indeed, prior research has shown that the geometric structure of deep feature representations, and consequently, the geometric understanding of the decision boundaries learned by deep networks, plays a critical role in both adversarial attacks and defense strategies [9]. Since the model's decision boundary is shaped by the training data, even a small number of adversarial samples can significantly alter its behavior, lead-

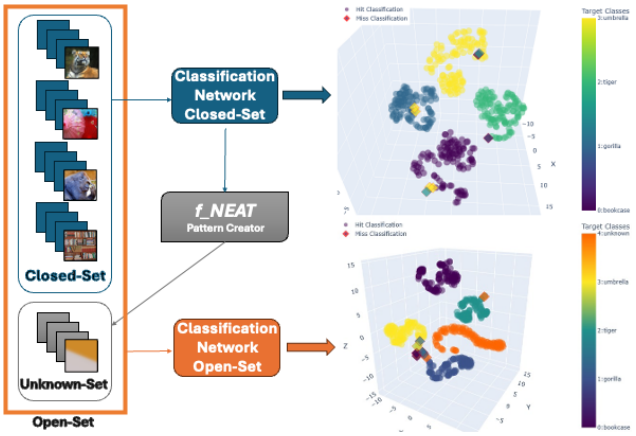

Figure 1. Pipeline of different classification network with closed and open-set, using `f_NEAT` to create patterns as the unknown set.

ing to misclassifications on genuine training instances [14]. This occurs because the hypothesis function is optimized to generalize across the entire training set, forcing it to accommodate adversarial perturbations at the expense of accurate classification. We leverage this property to strategically generate adversarial examples that simultaneously carve out dedicated regions for unknown samples while compacting the feature space allocated to genuine class instances, ultimately enhancing the model's ability to distinguish between known and unknown categories. The lack of explicit control over known and unknown regions in the feature space often results in inefficient separation between them, limiting the model's robustness in real-world applications. To address these challenges, we propose a novel framework that explicitly incorporates a dedicated unknown class during training. Inspired by adversarial learning techniques, we use a topology evolution method to generate synthetic unknown samples that are structurally distinct from known categories yet challenging for deep models to classify. The rest of this paper is structured as follows: Section 2 introduces the motivation and related work, while Section 3 presents the proposed method and explains how we generate the unknown set of images. Section 4 evaluates the experimental results for closed-set training and testing, open-set training and testing, and open-set only testing. Section 5 concludes with a discussion on future directions for evaluating known and unknown sets.

2. Motivation and Related Work

Classical supervised machine learning methodologies rely on training the model using a thoroughly labeled dataset. This ensures that once the model is trained, it can correctly classify new examples in the test set, a principle known as the closed-set assumption. Indeed, in classical approaches, even methods designed for real-world scenarios exploit the closed-set simplification [5, 18, 22]. Applications of open-world machine learning have been widely investigated by researchers in image processing, computer vision [7, 8, 13], and open-set recognition [11]. In the field of open-world classification and unknown class discovery, researchers aim to address the challenge of identifying and handling unknown classes. However, most existing approaches rely on rejecting unknown samples during classification or on incremental learning from unlabeled samples, without actively introducing controlled unknown classes into the training phase [13, 20]. Furthermore, the authors in [17] developed an open-world machine learning framework that identifies unknown instances by detecting unknown data and subsequently labeling them into novel categories. In [15], the authors investigate counterfactual image generation based on generative adversarial networks, generating examples that are close to training set examples yet do not belong to any training category. In [16], images are generated through evolutionary algorithms (EAs) that, although unrecognizable to a human observer, are classified with high confidence by a neural network as belonging to a specific class. See also [6], which evolves images to match a single `ImageNet` class. This suggests that the model relies on internal features for classification rather than actual visual similarity. Our idea for creating the unknown class stems precisely from the fact that the model does not rely on human perception but rather on the features learned during training, an interesting for an adversarial machine learning point of view [2]. In fact, the concept of creating an unknown class arises from the idea of constructing a category that does not visually represent a specific image, but it is built based on the knowledge acquired during the training, of well-defined image classes. Indeed, a key distinction in this work lies in the generation of the unknown class. [3] does not utilize real unknown class samples but instead generates latent representations called Reciprocal Points to model open space and enhance the separation from known classes, and [16] generates images with the aim of optimizing a deep neural network model to determine whether an image belongs to a specific class, without studying the feature space or the behavioral of this type of images. In contrast, our method explicitly introduces an unknown class created through an evolutionary network, enabling controlled manipulation of its distribution and characteristics. Our approach, in contrast to [3], examines how the synthetic unknown class naturally distributes within the feature space rather than enforcing a predefined structure. This results in a more exploratory approach, allowing for an in-depth understanding of how unknowns integrate with known classes without imposing rigid separability constraints, incrementing their accuracy intra-class. To make this evaluation we consider the `MNIST` and the `ImageNet` datasets, see also [21], where the authors generate adversarial images using the same datasets and analyze the pixel space of these images with noise of varying intensity and distribution. The approach presented in this work differentiates itself by introducing an unknown class that is synthetically generated using a custom-designed system called `Pattern_Creator`. This system integrates a specially configured evolutionary neural network, employs a custom fitness function called `f_NEAT`, and ultimately utilizes the model of the supervised classification network trained on a closed-set, see Figure 1. This novel aspect allows for a more controlled and systematic evaluation of the classifier's ability to distinguish between known and unknown classes. Moreover, our approach extends beyond classification and rejection by analyzing the cohesion of known classes and the spatial distribution of the unknown class within the feature space. This provides deeper insights into how unknown samples interact with established categories, offering a more comprehensive eval-

uation of model robustness. Compared to previous works that focus on open-set recognition, semi-supervised classification [19], or clustering-based discovery, our pipeline directly examines the structural integrity of feature representations in the presence of unknown data, which are seen or not during training, enabling an evaluation of their impact on feature space distribution and class accuracy.

3. Proposed Method

This section presents the methodology and the components used in the proposed work.

Definition 1 *Let D_k be the known dataset, such that:*

$$D_k = \{D_k^{c_i} \mid i = 0, \ldots, |C| - 1\}, \tag{1}$$

where $D_k^{c_i}$ represents the subset of samples belonging to class $c_i \in C$, and $|C|$ is the cardinality of set C of classes.

Definition 2 *Let $D_k^{c_i}$ be the subset of the balanced D_k containing r samples belonging to class c_i, such that:*

$$D_k^{c_i} = \left\{(x_{k_j}^{c_i}, y_{k_j}^{c_i}) \mid x_{k_j}^{c_i} \in X_k, y_{k_j}^{c_i} \in Y_k, y_{k_j}^{c_i} = c_i\right\}, \tag{2}$$

where X_k is the set of all input samples in D_k, and Y_k is the set of all labels associated with samples in D_k and, $_{j=1,\ldots,r}$.

Definition 3 *Let D_u be a unknown synthetic dataset of s synthetic samples that represent unknown instances, such that:*

$$D_u = \{(x_{u_\ell}, y_{u_\ell}) \mid \ell = 1, \ldots, s\}, \tag{3}$$

where $(x_{u_\ell}, y_{u_\ell}) \in D_u$ is the generic pair, with $x_{u_\ell} \in X_u$ as the input sample, and $y_{u_\ell} \in Y_u$ as the associated but unknown label in the given D_u.

Definition 4 *Let $D_{k,u}$ be the dataset resulting from the union of the known and unknown synthetic datasets, such that:*

$$D_{k,u} = D_k \cup D_u \tag{4}$$

Definition 5 *Let $\bar{D}_u$ be the open-world set that includes t samples from the unseen classes $\bar{c}_i$ of the considered dataset, such that:*

$$\bar{D}_u = \{(x_{\bar{u}_m}^{\bar{c}_i}, y_{\bar{u}_m}^{\bar{c}_i}) \mid x_{\bar{u}_m}^{\bar{c}_i} \in X_{\bar{u}},\ y_{\bar{u}_m}^{\bar{c}_i} \in Y_{\bar{u}},\ m = 1, \ldots, t\}, \tag{5}$$

where $X_{\bar{u}}$ is the set of all aggregated real-unknown inputs in $\bar{D}_u$, and $Y_{\bar{u}}$ is the set of all labels associated with sample in $\bar{D}_u$.

We denote by $\mathcal{X} := X_k \cup X_u \cup X_{\bar{u}}$ and $\mathcal{Y} := Y_k \cup Y_u \cup Y_{\bar{u}}$.

Table 1. Configuration of some parameters of the custom `NEAT` evolutionary network.

Parameter	Value
fitness_criterion	max
fitness_threshold	0.99 || 0.98
pop_size	200
num_inputs	2
num_hidden	2
num_outputs	1 || 3
initial_connection	full_nodirect
activation_function	sigmoid

3.1. Pattern Creator and Fitness Function

With the aim of making a classification system capable of recognizing, during the inference phase, when a sample is not effectively assignable to any of the known classes and therefore assigning it the *unknown* label, we simulated the scenario in which the model is unable to determine with high confidence the class of belonging among the known ones, using other datasets and more classes. To this end, we have defined a pattern generation method that integrates the NEAT (NeuroEvolution of Augmenting Topologies) technique. The patterns of interest are those for which, when the model, previously trained only on known classes, takes them as input, it returns an output probability distribution that is as uniform as possible. Therefore, the ideal expected output of the model is a uniform discrete distribution over $|C|$ classes (6). Consequently, the pattern should not favor any particular class.

$$\mathbb{E}(p_i) = \left[\frac{1}{|C|}, \frac{1}{|C|}, \ldots, \frac{1}{|C|}\right], \text{ s.t. } \sum_{i=0}^{|C|-1} p_i = 1. \tag{6}$$

A customized method called `Pattern_Creator` was designed to generate patterns using NEAT, an evolutionary algorithm for creating artificial neural networks. To run NEAT and adapt it to our problem, we have carefully configured its parameters via a configuration file (see Table 1) and defined a custom *fitness function*. The following sections describe the pattern generation process and the implementation of the fitness function.

3.1.1 Synthetic-pattern generation method

To generate grayscale (GS) or color (RGB) patterns (Figures 2 and 3, we use an evolutionary NEAT network in an iterative process. We configure the NEAT network with two input neurons, which receive the normalized coordinates $(x, y) \in ([-1, 1] \times [-1, 1])$ of the pixel to be generated. If the pattern to be generated is gray-scale, the network has a single output neuron that provides the pixel intensity. If the pattern is in color, the network has three output neurons,

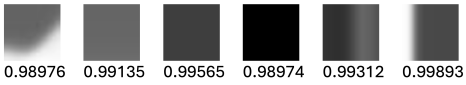

Figure 2. Example and relative fitness value of 2D gray generated pattern for MNIST dataset.

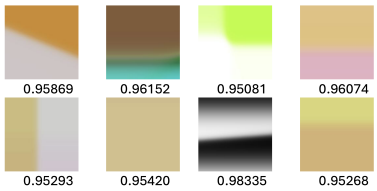

Figure 3. Example and relative fitness value of 3D color generated pattern for `ImageNet` dataset.

each associated with an RGB channel. The activation function chosen for the output nodes is the sigmoid function, ensuring that each neuron produces a value in the range $[0, 1]$, which can be interpreted as intensity levels for the colors. Subsequently, the generated pattern is processed through a pipeline of operations. **Data augmentation**: application of transformations such as Random Flip, Gaussian Blur, and Color Jitter (the latter only for RGB patterns). **Standardization**: values are normalized using the mean and standard deviation calculated on the `MNIST` or `ImageNet` dataset. **Inference**: the pattern is processed by the trained model to obtain the probability distribution P. **Performance evaluation**: computation of the *fitness function* value using P to calculate the fitness of the neural network (genome) generated in the current iteration by the NEAT method. If the fitness score meets or exceeds the established threshold, the pattern is saved.

3.1.2 Implementation of the fitness function

In the NEAT framework, *fitness* is used to evaluate how well the generated neural network performs a specific task. We have implemented a custom *fitness function*, called `f_NEAT`, that guides the evolution of networks to produce patterns increasingly closer to the desired goal. With the aim of obtaining a uniform probability distribution U, we adopted the *Jensen-Shannon divergence* (JSD) metric [12], a method for measuring the similarity between two probability distributions. The JSD is based on the *Kullback-Leibler divergence* (D_{KL}) [10], and introduces symmetry and stability. The JSD between the distributions P and Q is defined as follows:

$$\text{JSD}(P \parallel Q) = \frac{1}{2} D_{\text{KL}}(P \parallel M) + \frac{1}{2} D_{\text{KL}}(Q \parallel M), \quad (7)$$

Table 2. Description of probability distributions.

Probability distribution	Description
U	Ideal uniform probability distribution
P	Probability distribution obtained from the model
M	Average between the P and U distributions
$S_{c_i} = [0, \ldots, 0, 1, 0, \ldots 0]$	Probability distribution concentrated on a single class i

where

$$M = \frac{1}{2}(P+Q) \ , \quad D_{\text{KL}}(P \parallel Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \quad (8)$$

We propose the following definition of `f_NEAT`:

Definition 6 *Let the probability distributions given in Table 2 be considered. We compute the maximum divergence that is obtained between P and U as:*

$$JSD(P \parallel U)_{max} = JSD(S \parallel U) \quad (9)$$

Then, we compute normalized JSD in the range [0,1] as:

$$JSD_{norm} = \frac{JSD(P \parallel U)}{JSD(P \parallel U)_{max}} \quad (10)$$

Finally, our fitness function is `f_NEAT` $= 1 - JSD_{norm}$

- ***Optimal case*** $: if \ \ P \equiv U, \ \ then$

$$JSD_{norm} = 0, \quad hence \quad \texttt{f_NEAT} = 1$$

- ***Worst case*** $: if \ \ P \equiv S, \ \ then$

$$JSD_{norm} = 1, \quad hence \quad \texttt{f_NEAT} = 0$$

Condition 1 *A pattern is considered suitable if and only if the following condition holds:* `f_NEAT` $\geq$ *Threshold, where:*

- *Threshold* $= 0.98$ *for grayscale (GS) patterns generated for the* `MNIST` *dataset.*
- *Threshold* $= 0.95$ *for RGB patterns generated for the* `ImageNet` *dataset.*

To generate synthetic patterns for the unknown class, the `Pattern_Creator` system was run for multiple rounds, each executing up to 200 generations. At each generation, the patterns satisfying Condition 1 were saved. Figures 2 and 3 show some example patterns, respectively for `MNIST` and `ImageNet`, along with their fitness values.

4. Experimental evaluation

In this subsection, we evaluate the proposed technique to analyze the distribution of classes in the features space for both known and unknown classes. In particular, as observed in [11], the experiments are structured in three steps defined as follows:

Table 3. Training parameters for `Net` on `MNIST` and for ResNet-18 on `ImageNet` for closed-set classification.

Parameter	Net Values	ResNet-18 Values
Optimizer	SGD	SGD
Learning Rate (lr)	0.01	0.01
Momentum	0.5	0.9
Weight Decay	0	0.0005
Scheduler	/	StepLR
Step Size	/	30
Gamma	/	0.1
Loss Function	CrossEntropy	CrossEntropy

1. *closed-set classification*, in which we made a supervised classification; this step is referred to as Experiment 1;

2. *open-set training and testing unknown*, where we performed a supervised classification using the same classes as in the *closed-set classification*, plus one unknown class consisting of synthetic images; this step is referred to as Experiment 2;

3. *open-set testing with an additional unknown set*, in which we evaluate the model trained in *open-set training*. The unknown set includes both real and synthetic images. The real images come from different classes than those considered in the *closed-set classification* but belong to the same dataset; this step is referred to as Experiment 3a-3b.

4.1. Closed-set classification

In correspondence of [21], to evaluate the proposed method, we conducted experiments on two types of datasets: one containing grayscale images and the other containing RGB images; `MNIST` and `ImageNet`, respectively. For each dataset, we chose four classes: from the `MNIST` dataset, the digits `2, 4, 6, 8`; while from the `ImageNet` dataset, the classes `bookcase, gorilla, tiger, umbrella`. From `MNIST`, we created a dataset consisting of four balanced classes, each containing 6800 images. The dataset was split into three subsets: training (≈69%), validation (≈17%), and test (≈14%) using a stratified split to maintain class proportions across sets. Since `MNIST` images are simple ($1 \times 28 \times 28$), the model used for training, referred to as `Net`, consists of two convolutional layers and two fully connected layers. The architecture employs max pooling for dimensionality reduction and employs the ReLU activation function to model complex, non-linear behaviors. Training was performed using Stochastic Gradient Descent (SGD) with a learning rate of 0.01 and CrossEntropyLoss as the loss function. Table 3 presents the training parameters, while Table 4 reports the performance obtained after 10 epochs. Figure 4 displays the confusion matrices for the training, validation, and test sets, showing class-wise accuracy distributions. From

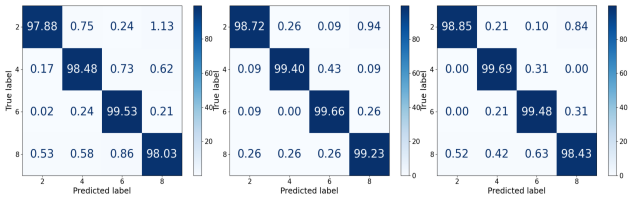

Figure 4. Confusion matrices for `MNIST` training, validation and test in closed-set classification task, Experiment 1.

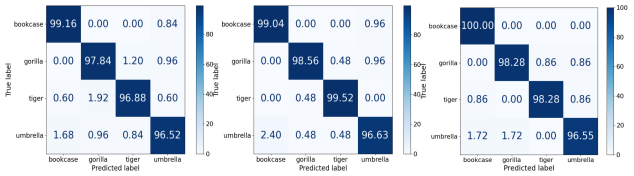

Figure 5. Confusion matrices for `ImageNet` training, validation and test in closed-set classification task, Experiment 1.

`ImageNet`, we created a dataset consists of 4632 images of size ($3 \times 224 \times 224$), with 1158 samples per class. The dataset was split into training (834 images per class), validation (209 images per class), and test (116 images per class). For classification, we employed a ResNet-18 model pretrained on `ImageNet`. Table 3 details the model parameters, while Tables 4 and 5 present the obtained performance of loss, total accuracy and classes' accuracy, respectively. Figure 5 present the confusion matrices for the training, validation, and test sets for Experiment 1, i.e. the *closed-set classification*.

4.2. Open-set training and testing unknown

In this second part of the experiments we analyze the *open-set training, validation*, and *testing unknown*, such as a supervised classification, where we use the same 4 classes used in the *closed-set classification* plus an unknown class created by the synthetic images. Indeed, the set of synthetical images belonging to D_u was concatenated with the dataset $\bigcup_{i=0}^{3} D_k^{c_i}$ of known classes and labeled as the *unknown* class. The amount of samples of this class was set to match that of each class in the closed-set experiments' dataset. The extended dataset $D_{k,u}$ was then used to train the network, `Net` for `MNIST` or `ResNet18` for `ImageNet`. with an additional output node. The model hyper-parameters remained unchanged, as did the number of epochs (see Table 3). In Figures 8 and in Table 4, the performance of the models, loss and total accuracy, for the classification task on the extended dataset is reported. In fact, in Table 5, we observe that the `ImageNet` dataset improves the training, the validation and the test accuracy of the `tiger` and `umbrella` classes. The `MNIST` dataset improves classes `2`, `4` in the training and validation, classes `4`, `8` in the test phase, and reaches the equal value for the

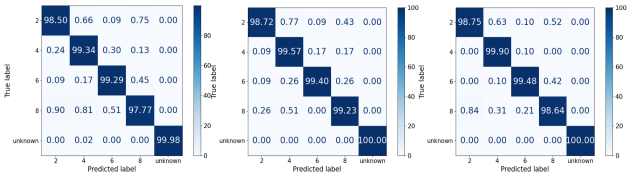

Figure 6. Confusion matrices for `MNIST` training, validation and test in open-set classification task, Experiment 2.

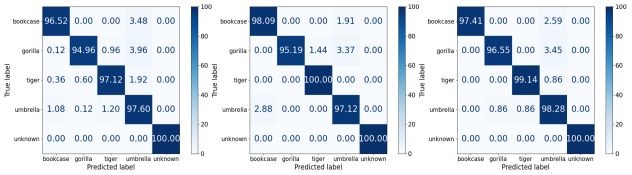

Figure 7. Confusion matrices for `ImageNet` training, validation and test in open-set classification task, Experiment 2.

first and the second experiments in validation phase for classes `8` and in test phase for class `6`. For the results in Table 4, `MNIST` achieves the highest accuracy across training, validation, and testing, and `ImageNet` reaches the same testing accuracy, comparing Experiment 1 and 2.

4.2.1 Feature Space Visualization

To evaluate the effectiveness of the features (extracted from the samples through the convolutional layers) to guarantee a good separability between the classes, we considered the probability distributions P, which are obtained by applying the Softmax function to the output of the model, as *characteristic descriptors* of the samples. We define the condition for assigning a sample to the predicted class $c_h \in C$, as:

Definition 7 *Let $P = [p_0, p_1, \ldots, p_{|C|-1}]$ be the probability vector for j-th sample $x_{k_j}^{c_i} \in X_k$. The sample is said to belong to class $c_h \in C$ if and only if:*

$$p_h = \max_{i=0}^{|C|-1}\{p_i\}, \quad where \;\; h \in \{0, \ldots, |C|-1\} \qquad (11)$$

By mapping the $|C|$-dimensional *characteristic descriptors* into an embedding space, we expect that samples of the same class cluster closely together, while those from different classes remain distant and augment their accuracy intra-class. To visualize the $|C|$-dimensional vectors in a 3D space, we employed the t-SNE (t-Distributed Stochastic Neighbor Embedding) dimensionality reduction technique. The t-SNE defines a mapping function $\Psi : \mathbb{R}^d \rightarrow \mathbb{R}^3$. In our context, $P \in [0,1]^d \subset \mathbb{R}^d$, where $d = |C|$. t-SNE preserves the local structure of the data by reducing dimensionality in such a way that similar points remain close to each other, while dissimilar points are pushed apart. Figures 8 depicts the 3D mappings of test set descriptors for

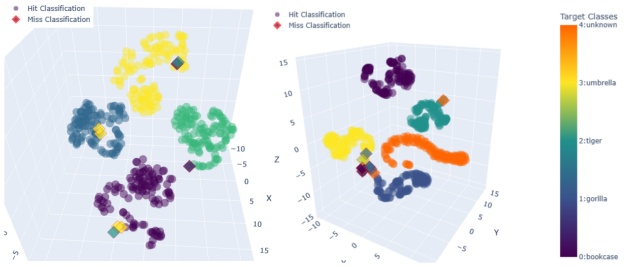

Figure 8. t-SNE visualization of known and unknown classes features in `ImageNet` in closed and open-set classification training and test task, Experiment 1 and 2.

`ImageNet`, for Experiment 1 and 2, respectively. Although the overall accuracy is slightly lower at the second decimal place, there is a significant increase in the accuracy of individual classes, highlighting the effectiveness of incorporating undefined elements, not only visually, such as unknowns, which lead to greater class cohesion. In fact, from Table 5 and from Figures 5 and 7, we observe that in `ImageNet` dataset, we improve the training, the validation and the test accuracy of the `tiger` and `umbrella` classes, in Experiment 2. For lack of space, we do not introduce t-SNE for the experiment with the `MNIST` dataset.

4.3. Open-set testing on real-unknown set

The models trained on datasets composed of the known classes and pattern class (called synthetic unknown) were used as an inference tool in the purely open-set environment. To this end, the same samples of the known classes of the test set employed in the previous experiment were used, to which samples belonging to never seen classes were added, in addition to a section of the patterns. We call augmented class the real-unknown class $\bar{c}_i$, i.e. an augmented class is a class that is unknown during the training stage, but appears in the test stage, [4]. In particular, for the experiment referred to `MNIST`, the samples of the classes `1`, `3` and `5` were chosen, while for the experiment on `ImageNet` those of the classes `lemon`, `socks` and `hamster`. The real-unknown class has the same number of elements as each known class (116 for `ImageNet`, 958 for `MNIST`) with a uniform distribution of the elements of the 4 sub-classes (i.e for `ImageNet`: 29 sock, 29 lemon, 29 hamster, and 29 patterns). Figure 9 shows the confusion matrices of the open-set testing experiment, respectively for `MNIST` and `ImageNet`. Between the two types of experiments (testing with synthetic-unknown and open-set testing with real-unknown), the probability distribution of correctly classified samples was compared for each of the known classes. It was observed that the minimum and maximum values of the probabilities relating to the individual classes are identical across both experiments. This allowed

Table 4. Performance of Loss and Total Accuracy of the three experiments for closed and open-set classification using `Net` and ResNet-18 on `MNIST` and `ImageNet`, respectively.

Experiment	Dataset	Set	Loss	Accuracy (%)
Experiment 1	`MNIST`	Training	0.1099	98.48
		Validation	0.0303	99.25
		Test	-	99.11
	`ImageNet`	Training	0.0958	97.60
		Validation	0.0497	98.44
		Test	-	**98.28**
Experiment 2	`MNIST`	Training	0.0927	**98.98**
		Validation	0.0210	**99.38**
		Test	-	**99.31**
	`ImageNet`	Training	0.1184	97.24
		Validation	0.0522	98.08
		Test	-	**98.28**
Experiment 3a	`MNIST`	Test	-	84.41
	`ImageNet`	Test	-	83.45
Experiment 3b	`MNIST`	Test	-	85.15
	`ImageNet`	Test	-	83.79

us to determine a criterion to be applied in the open-set testing environment to accept or reject a sample predicted as belonging to one of the known classes and, consequently, to establish the correct label to assign $\tilde{y}$. Specifically, for each of the known classes, the minimum value was considered. The following describes this criterion:

Criterion 1 *Let $y_{k_j}^{c_i}$ be the true label corresponding to class $c_i \in C$ of the j-th sample belonging to the known dataset D_k, which contains r samples; $\hat{y}_{k_j}^{c_h}$ the corresponding predicted label, and $p_{k_j}^{c_h}$ the corresponding probability (computed as 11), we defined $min_k^{c_i}$ be minimum value, as:*

$$min_k^{c_i} = \min \left\{ p_{k_j}^{c_h} \mid \hat{y}_{k_j}^{c_h} = y_{k_j}^{c_i}, \ \forall j = 1, \ldots, r \right\} \quad (12)$$

Let $\hat{y}_{k,u_j}^{c_h}$ be the predicted label for j-th sample from dataset known-unknown $D_{k,u}$, $p_{k,u_j}^{c_h}$ be the corresponding probability (computed as 11), $y_k^{c_i}$ be label indicating class c_i, $\tilde{y}_j$ be the correct label to assign, established as:

$$\tilde{y}_j = \begin{cases} y_k^{c_i}, & \text{if } \hat{y}_{k,u_j}^{c_h} = y_k^{c_i} \text{ and } p_{k,u_j}^{c_h} \geq min_k^{c_i} \\ unknown, & otherwise \end{cases} \quad (13)$$

The open-set testing experiment was repeated by applying Criterion 1 (13). From the obtained confusion matrices 10, it is evident that, for both datasets, the percentage of samples correctly classified as belonging to the *unknown* classes increased by approximately 5.65% for `MNIST` and 4.31% for `ImageNet`, while the percentage of samples correctly classified as known remained unchanged (within a fraction of a percentage point). This allows us to conclude that applying Criterion 1 (13) made the classifiers more robust

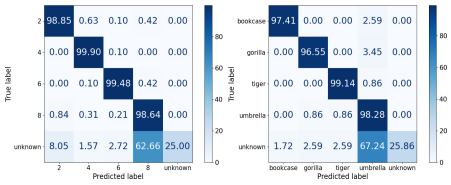

Figure 9. Confusion matrices for `MNIST` and `ImageNet` in open-set testing, Experiment 3a.

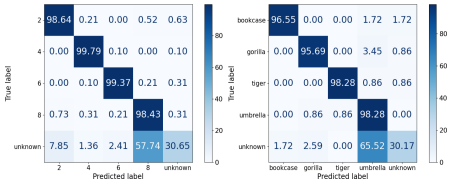

Figure 10. Confusion matrices obtained by the Crit.1 in open-set testing referring to `MNIST` and `ImageNet`, Experiment 3b.

against potential impostors that may appear in the open-set testing environment with real-unknown classes. The advantage of this experiment is that the criterion was designed by considering the statistics obtained from testing with synthetic-unknown samples on the model that will later be used in the real environment. In conclusion, as shown in Table 4, the `MNIST` dataset achieves the highest overall accuracy across training, validation, and testing, even when introducing unknown classes from Experiment 1 to Experiment 2. In addition, in Table 5, we highlight in bold, the highest accuracy achieved for each class across the first three experiments, while bold and underlined values indicate cases where the highest accuracy is reached with the same value in two or more experiments. From this analysis, several key observations emerge. While global accuracy increases across all phases for the `MNIST` dataset and only in testing for `ImageNet`, Experiment 2 shows a significant improvement in individual class accuracy compared to the closed-set scenario, across training, validation, and testing, where no unknown classes are introduced. In Experiment 3a, introducing previously unseen class images during testing does not significantly impact individual known class accuracies, although it results in a decrease in global accuracy. Specifically, for `MNIST`, the highest accuracy obtained in other experiments is matched in three out of four classes, while for `ImageNet`, the accuracy values that outperformed the closed-set scenario, i.e. Experiment 1, are also matched. This finding underscores the robustness of the model, demonstrating its ability to maintain performance even in the presence of unseen classes.

5. Conclusion

This work presents a novel approach for open-set recognition by explicitly incorporating the concept of unknown

Table 5. Performance of individual class accuracy of the three experiments for closed and open-set classification using Net and ResNet-18 on MNIST and ImageNet, respectively.

Experiment	Dataset	Known Class	Accuracy (%)		
			Training	Validation	Test
Experiment 1	MNIST	2	97.88	**98.72**	**98.85**
		4	98.48	99.40	99.69
		6	**99.53**	**99.66**	**99.48**
		8	**98.03**	**99.23**	98.43
	ImageNet	bookcase	**99.16**	**99.04**	**100.0**
		gorilla	**97.84**	**98.56**	**98.28**
		tiger	96.88	99.52	98.28
		umbrella	96.52	96.63	96.55
Experiment 2	MNIST	2	**98.50**	**98.72**	98.75
		4	**99.34**	**99.57**	**99.90**
		6	99.26	99.40	**99.48**
		8	97.77	**99.23**	**98.68**
	ImageNet	bookcase	95.52	98.09	97.41
		gorilla	94.96	95.19	96.55
		tiger	**97.12**	**100.0**	**99.14**
		umbrella	**97.60**	**97.12**	**98.28**
Experiment 3a	MNIST	2	-	-	**98.85**
		4	-	-	**99.90**
		6	-	-	**99.48**
		8	-	-	98.64
	ImageNet	bookcase	-	-	97.41
		gorilla	-	-	96.55
		tiger	-	-	**99.14**
		umbrella	-	-	**98.28**

class during training, and assessing the behavior of the model with respect unseen sample categories during test, with a specific focus on the deep embedding space. In this context, we aimed to influence the model's decision boundary by inserting specific samples during training. In particular, inspired by adversarial learning, we leveraged an evolution method to generate synthetic unknown samples that are structurally distinct from known categories. Our approach allows for a controlled study of class cohesion in an open-set scenario than in a closed one, improving the classifier's ability to distinguish between known and unknown instances while optimizing feature space utilization. Through extensive experiments on the MNIST and ImageNet datasets, we demonstrated that integrating an explicit unknown class leads to a more effective management of class distribution into the feature space, significantly improving recognition performance. The introduction of synthetic unknown samples resulted in increased intra-class cohesion and enhanced robustness against unseen categories. Notably, we also proposed, in the last experiment, a preliminary method to reduce the mis-classification rate of unknown instances into known classes while maintaining high classification accuracy for known samples. Our findings suggest that the model learns to allocate a dedicated region in the feature space for uncertain samples rather than forcefully assigning them to known categories, excepted for vulnerable class. This contributes to a more structured representation, mitigating the effects of adversarial examples and improving the interpretability of classification decisions. By incorporating the unknown class, we improved the structural organization of the feature space. Indeed, as a result of our approach, the classifier allocated a distinct region for uncertain samples instead of forcefully assigning them to known categories. This adjustment significantly enhances classification performance, especially in open-set recognition scenarios. To validate our method, we conducted various experiments, considering all possible scenarios for the unknown class. Specifically, we evaluate its presence across different phases, training, validation, and testing, as well as restricting it to the test phase only. This approach highlights the potential for exploring and refining classification systems to enhance their robustness, by identifying vulnerable classes, we aim to develop strategies to protect them and mitigate mis-classification issues. Our results demonstrate that integrating an explicit unknown category leads to a more effective management of out-of-distribution inputs, improving the model's robustness in real-world applications.

5.1. Future Works

In this work, we primarily focused on evaluating the classification accuracy of individual classes through the closed-set and open-set recognition scenarios and investigating the classes behavior into the feature space. However, in future works we will investigate a more in-depth analysis of the feature space from a topological perspective to better understand how the unknown class is distributed relative to the known ones. Future research will deeper focus on this aspect by leveraging advanced techniques such as Reciprocal Point Learning, which can help enforce better separation between classes and improve the robustness of the model, among others. Additionally, we will extend our study to specific application domains. A pivotal example is represented by face recognition in the context of person authentication, where the ability to correctly handle unknown instances is crucial for security and biometric authentication and recognition systems in adversarial and real-world environments. This will allow us to assess the effectiveness of our method in real-world settings and refine the feature space representation to better distinguish between known and unknown categories. Ultimately, this work provides a theoretical foundation for further advancements in open-set recognition.

Acknowledgments. Authors are supported by PNRR MUR project PE0000013-FAIR, by Spoke 1 Future HPC & Big Data of the Italian Research Center on High-Performance Computing, by SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union NextGenerationEU, and by Research Program PIAno di inCEntivi per la Ricerca di Ateneo 2020/2022, Linea di Intervento 3 "Starting Grant", University of Catania.

References

[1] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572, 2016. 1

[2] Nino Cauli, Alessandro Ortis, and Sebastiano Battiato. Fooling a face recognition system with a marker-free label-consistent backdoor attack. In *International Conference on Image Analysis and Processing*, pages 176–185. Springer, 2022. 2

[3] Guangyao Chen, Peixi Peng, Xiangqian Wang, and Yonghong Tian. Adversarial reciprocal points learning for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8065–8081, 2021. 2

[4] Qing Da, Yang Yu, and Zhi-Hua Zhou. Learning with augmented class by exploiting unlabeled data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28, 2014. 6

[5] Georgia Fargetta, Alessandro Ortis, Stefano Anile, and Sebastiano Battiato. Evaluation of cnns for wildcats classification in real world scenario. In *International Conference on Advanced Engineering, Technology and Applications*, pages 15–25. Springer, 2024. 2

[6] Dario Floreano and Claudio Mattiussi. *Bio-inspired artificial intelligence: theories, methods, and technologies*. MIT press, 2008. 2

[7] Yang Gao, Yi-Fan Li, Bo Dong, Yu Lin, and Latifur Khan. Sim: Open-world multi-task stream classifier with integral similarity metrics. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 751–760. IEEE, 2019. 2

[8] Xiaojie Guo, Amir Alipour-Fanid, Lingfei Wu, Hemant Purohit, Xiang Chen, Kai Zeng, and Liang Zhao. Multi-stage deep classifier cascades for open world recognition. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 179–188, 2019. 2

[9] Marc Khoury and Dylan Hadfield-Menell. On the geometry of adversarial examples. *arXiv preprint arXiv:1811.00525*, 2018. 1

[10] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951. 4

[11] Jinsol Lee and Ghassan AlRegib. Open-set recognition with gradient-based representations. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 469–473. IEEE, 2021. 2, 4

[12] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991. 4

[13] Ting-En Lin and Hua Xu. A post-processing method for detecting unknown intent of dialogue system via pre-trained deep neural network classifier. *Knowledge-Based Systems*, 186:104979, 2019. 2

[14] Jasmita Malik, Raja Muthalagu, and Pranav M Pawar. A systematic review of adversarial machine learning attacks, defensive controls and technologies. *IEEE Access*, 2024. 2

[15] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 613–628, 2018. 2

[16] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015. 2

[17] Jitendra Parmar, Satyendra Singh Chouhan, and Vaskar Raychoudhury. A machine learning based framework to identify unseen classes in open-world text classification. *Information Processing & Management*, 60(2):103214, 2023. 2

[18] Francesco Ragusa, Valeria Tomaselli, Antonino Furnari, Sebastiano Battiato, and Giovanni M Farinella. Food vs non-food classification. In *Proceedings of the 2nd International workshop on multimedia assisted dietary management*, pages 77–81, 2016. 2

[19] Stefan Schrunner, Bernhard C Geiger, Anja Zernig, and Roman Kern. A generative semi-supervised classifier for datasets with unknown classes. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, pages 1066–1074, 2020. 3

[20] Lei Shu, Hu Xu, and Bing Liu. Unseen class discovery in open-world classification. *arXiv preprint arXiv:1801.05609*, 2018. 2

[21] Pedro Tabacof and Eduardo Valle. Exploring the space of adversarial images. In *2016 international joint conference on neural networks (IJCNN)*, pages 426–433. IEEE, 2016. 2, 5

[22] Qingguo Xiao, Guangyao Li, Li Xie, and Qiaochuan Chen. Real-world plant species identification based on deep convolutional neural networks and visual attention. *Ecological Informatics*, 48:117–124, 2018. 2