Acta Cybernetica **26** (2024) 297–322.

Single and Combined Algorithms for Open Set Classification on Image Datasets

Modafar Al-Shouha[ab] and Gábor Szűcs[ac]

Abstract

Generally, classification models have closed nature, and they are constrained by the number of classes in the training data. Hence, classifying ”unknown” – OOD (out-of-distribution) – samples is challenging, especially in the so called ”open set” problem. We propose and investigate different solutions – single and combined algorithms – to tackle this task, where we use and expand a K-classifier to be able to identify $K + 1$ classes. They do not require any retraining or modification on the K-classifier architecture. We show their strengths when avoiding type I or type II errors is fundamental. We also present a mathematical representation for the task to estimate the $K + 1$ classification accuracy, and an inequality that defines its boundaries. Additionally, we introduce a formula to calculate the exact $K+1$ classification accuracy.

Keywords: binary classification, multi-class classification, GAN, out-of-distribution, open set classification

1 Introduction

In the field of computer vision, classification is one of the earliest and most common tasks that are challenged by deep neural networks [38]. With the availability of large, well maintained training datasets, and the advancement of convolutional neural networks (CNNs) [21, 22], neural networks could achieve remarkable results in performing this task. However, their classification ability is bounded by the training data features and attributes [3].

Majority of these neural networks apply SoftMax [14] function on the last layer, that outputs the probability of each of the K training classes, and as a result the most likely class is chosen accordingly. One main limitation is the inability of classifying an instance correctly in case it is not presented during training, i.e. OOD (out-of-distribution) or ”unknown” class. The task to overcome this limitation is

[a] Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Műegyetem rkp. 3., H-1111, Budapest, Hungary

[b] E-mail: modafar.alshouha@tmit.bme.hu, ORCID: 0000-0003-2051-4036

[c] E-mail: szucs@tmit.bme.hu, ORCID: 0000-0002-5781-1088

DOI: 10.14232/actacyb.298356

called open set recognition, open set classification, or as we call it $K+1$ classification. As a solution for such challenge, data can be produced or gathered for the $K+1$ class, and a classifier could be trained on $K+1$ classes instead of K. However, this solution remains insufficient and constrained by the ambiguity of defining the ”unknown” class while covering its wide features and possibilities.

Another way to address $K+1$ classification is by adjusting K-classifier to be able to solve the task. Most of the available approaches require retraining for the original K-classifier or altering its architecture [43, 46]. In this paper, we propose and study several solutions, that avoid the need for defining the ”unknown” data explicitly or retraining the original K-classifier. The first group of solutions consists of two single algorithms. One of them relies on the K classes confidence when classifying an instance. The other takes advantage of GANs [15] to learn the representation of the training data. A GAN consists of two parts, a generator and a discriminator, and there is competition between them. The generative network generates candidates while the discriminative network evaluates them. We use the discriminator block as a binary classifier to distinguish between ”known” and ”unknown” instances, before performing K classification. As for the second group we propose more robust solutions, by joining the strengths of various individual algorithms; namely, the discriminator-based algorithm from the first group with a threshold-based algorithm.

Moreover, we suggest a formula that represents mathematically the $K+1$ algorithm classification accuracy when following our approach. Based on this formula, we define an inequality that sets the boundaries for the $K+1$ algorithm classification accuracy. We validate those formulas empirically, and show that the test results confirm their applicability.

In the next chapter we present some solutions that try to tackle the $K+1$ classification task. Then we detail the proposed algorithms in two groups; single and combined ones. In the same chapter, we introduce and prove the constructed formula and inequality. After detailing the examination approach and presenting the used models, datasets and metrics, we show and discuss the experimental results. Lastly, we conclude the paper and review the limitations and future work possibilities.

2 Related work

Supervised learning methods hold an assumption about the excessive similarity between training and testing data. With the presence of ”open set” data, the performance of such models might degrade hugely, and it could be worse than random guessing [9]. Many solutions were proposed to address this challenge focusing on enhancing the supervised learning pipeline [33].

In computer vision related tasks, learning the feature representation is the first component of the pipeline, where the aim is to achieve a proper generalization on unseen target domain instances (images), i.e. images under different circumstances. Some methods try to learn the disentangled and casual feature representation of the data [4, 27], in order to assess the model generalization ability over OOD data during the learning process [17, 20, 42]. DANN (domain-adversarial neural network) [12, 13], CIAN (conditional invariant adversarial network) [24], and some others follow domain adversarial learning approach to catch the domain invariant features during training and inference. Another approach for representation learning is to increase and decrease the distances between different and similar domain instances, i.e. domain alignment [23, 36, 39].

Other works focus on the training strategy. Under this category, Finn et al. [11] and later improvements aim to achieve model domain generalization, this is done with the help of meta-learning [19]. Works [32, 40, 45] follow ensemble learning approach, by combining group of models from different domains' knowledge. Papers [26, 44] adopt semi-supervised and unsupervised approaches. Zhang et al. [43] combine the original classifier with a discriminative classifier. Later, they train the model (end-to-end) based on the latent feature space of the train data. They propose a flow-based model (OpenHybrid), without facing a common issue of assigning larger likelihood to the OOD data.

Closer to our work, ODIN (Out-of-DIstribution detector for Neural networks) [25] does not require model retraining, but it involves temperature scaling and input preprocessing inspired by other papers [16, 18]. Additionally, they introduce a detector which catches the OOD data after combining the preprocessing components. On the other hand, paper [46] integrates a GAN network from an AC-GAN [28], where the discriminator is used as a K+1 classifier for HSIs (hyper-spectral images).

In this work, Double Probability Model (DPM) [30] is used in constructing some of the combined algorithms. DPM relies on the likelihoods of a classifier with the assumption that the training data is accessible. The K classifier cumulative distribution function (CDF) and its inverse (inverse-CDF) are calculated. After obtaining the K classifier output for an instance, and using CDF and inverse-CDF, the probability of the i^{th} and $K+1$ classes are calculated, P_{C_i} and $P_{C_{K+1}}$ respectively. Lastly, the condition described by Formula 1 is checked, and if it is true, then $K+1$ class is assigned to the instance, otherwise, the K classifier predicted label. Thus, the K classifier is extended by the OOD class; $K+1$.

$$P_{C_{K+1}} > max_i\{P_{C_i}\} \tag{1}$$

3 Proposed method

In this paper, we propose multiple algorithms with various combinations to tackle $K+1$ classification task. In contrast to prior work, these algorithms are model agnostic (where the model is a K-classifier), and they do not require K-classifier retraining or any modification on its architecture. The task becomes more difficult in scenarios where the training data is not accessible. Our aim is to tackle these challenges while maintaining the immunity against several uncertainties, including but not limited to: OOD data characteristic and amount. We rely on two assumptions; (1) models tend to assign lower likelihoods to OOD (out-of-distribution) than

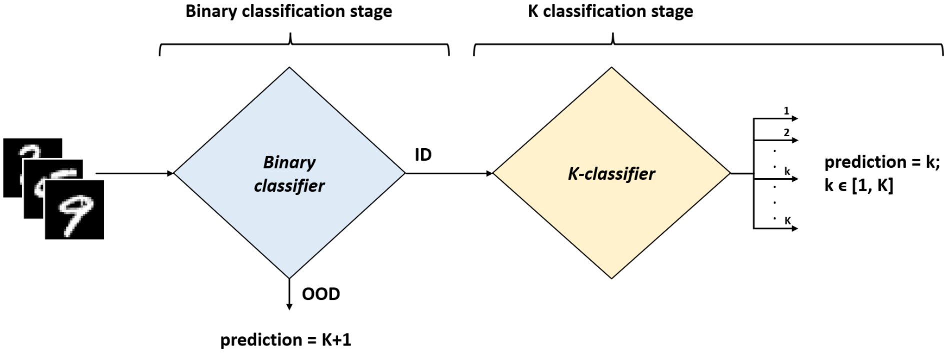


Figure 1: $K + 1$ classification task stages; (1) binary classification, and (2) K classification.

ID (in-distribution) instances [5], and (2) the two distributions (ID and OOD) are different [1].

Moreover, $K + 1$ classification task can be divided into two steps, (1) binary classification, and (2) K classification (Figure 1). Firstly, in the binary classification stage, the instance is categorized as either ID or OOD. If it is determined as OOD instance, $K + 1$ label is assigned to it. Otherwise, it is directed to the K-classifier to get one of the K labels. Therefore, our approach is to construct the binary classification component, and let the original K-classifier to handle the other task. To do so, we design two groups of algorithms; i.e. single [2] and combined. In the combined algorithms, the binary classification task is performed jointly by two methods; while in the single algorithm, it is an individual decision. Furthermore, we formulate the overall accuracy of the $K + 1$ classification task, by connecting the test accuracy scores of the individual components (binary and K classification), and the ratio of the OOD data in the test set. Consequently, we define an inequality for the $K + 1$ classification accuracy based on those factors.

3.1 Single algorithms

We propose two single algorithms, one of them is the Threshold algorithm "Thr" (Algorithm 1) that relies on the prediction of the K-classifier. The prediction output of the K-classifier is a vector that contains values that can be considered as probabilities of an instance belonging to the K classes, i.e. higher probability means more confidence. The Threshold algorithm checks the highest confidence level for a prediction, and if it is lower than the threshold value, the instance is assigned an $K + 1$ label, otherwise, the label is one of the K classes (K-classifier prediction). The threshold value β ($\beta \in [0, 1]$) represents the aggressiveness of the algorithm, higher threshold means that the algorithm is more strict in considering the ID decision from the classifier. Despite that a similar idea was presented earlier, e.g. by [43] and [30], those approaches require either an end-to-end retraining or access to the training data, to obtain a proper threshold value. In contrast, our proposed ”Thr” algorithm does not assume that the training ID data is accessible, and does not require any data preprocessing or additional model training.

Algorithm 1 Threshold algorithm

```
Obtain $Preds$, $\beta\,(Threshold)$
$y_i = ArgMax(Preds)$
$P_i = Max(Preds)$                          Prediction confidence
if $P_i < \beta$ then
    $Y_i = not\,in\,class$                  K+1 label
else
    $Y_i = y_i$                             $K$-classifier label
end if
```

The other proposed algorithm is the Discriminator algorithm ”Disc”; which, unlike ”Thr”, requires access to ID training data. We construct a $K+1$-classifier by using a discriminator block from a GAN as a binary classifier, then cascade it by the pre-trained K-classifier. We built a simple GAN using two convolution and two deconvolution layers for its discriminator and generator, respectively. In contrast to some of the papers which are listed in the Related Work [11, 19, 24, 28], neither the GAN nor the Discriminator component has a special architecture. Additionally, unlike [25, 40, 45] it does not require knowledge or assumption about the OOD data distribution, and it does not need any preprocessing. Furthermore, in contrast to the mentioned papers [12, 13, 17, 20, 23, 32, 36, 39, 43], all the proposed methods do not need any access, modification or retraining for the K-classifier. Later, we trained the GAN on the available ID data only. The discriminator's job is to distinguish between ”real” and ”fake” instances based on the knowledge it gains about the data during the GAN training process. We use this discriminator to catch the OOD instances before deciding if the K-classifier prediction is considered or not. If the discriminator defines an instance as ”fake”, $K+1$ label is assigned to it, otherwise, its label is the one that is offered by the K-classifier. While this algorithm requires an extra training step with access to the ID training data, there is no need to retrain the original K-classifier. Additionally, it is considered as a generic method, and there are no specific characteristics defined for the GAN network or any of its components (Algorithm 2).

3.2 Combined algorithm

Although the simplicity of the single algorithms is a big advantage, stability issues in the performance might appear. For instance, the decision about β value is crucial and influences ”Thr” algorithm performance. Also, relying on the K-classifier Softmax confidence might be misleading in our task [31]. In order to utilize their strengths, we combine them together in different variations. The general framework stays the same; at first, if the instance is OOD, $K+1$ label is assigned to

Algorithm 2 Discriminator algorithm

```
Obtain $Preds$, $Disc\_Pred$
$y_i = ArgMax(Preds)$
if $Disc\_Pred = fake$ then
    $Y_i = not\,in\,class$                          ▷ K+1 label
else
    $Y_i = y_i$                                     ▷ K-classifier label
end if
```

it, otherwise, K-classifier decision is considered. The combination happens in the binary classification level; at the stage where an instance is allocated either to ID or OOD. Unlike in the single algorithms, the decision is jointly made by two individual algorithms.

The combination has two main aspects; (1) selecting the individual methods to combine, and (2) defining the logical relation between their decisions. We join our proposed ”Disc” method with a threshold based method. For the latter, we use either our proposed ”Thr” algorithm or ”DPM” (Double Probability Model) [30]. ”DPM” [30] is a threshold based approach, which unlike ”Thr” algorithm demands access on training data, and its threshold value is dynamically set. Regarding the other aspect, logical ”OR” and ”AND” are used to combine the decisions of the individual methods (Table 1). As a result, the combined algorithm has four different variations: ”ThrAndDisc”, ”ThrOrDisc”, ”DpmAndDisc” and ”DpmOrDisc”.

Table 1: Truth table for combined algorithms. The first main column shows two individual methods (”A” and ”B”) decisions, while the other represents their decisions combined by logical ”OR” and ”AND”.

individual decision		combined decision	
method A	method B	OR	AND
ID	ID	ID	ID
ID	OOD	OOD	ID
OOD	ID	OOD	ID
OOD	OOD	OOD	OOD

3.3 Formula for estimating the open set classification accuracy

Our approach to solve the $K+1$ classification task goes through two stages; (1) binary classification to opt out the OOD instances, and (2) K classification for the instances which are ruled to be ID (Figure 1). Accordingly, the accuracy of the algorithm can be estimated by combining high level information about the binary and K classification tasks. This can be accomplished by incorporating (1) the ratio of OOD data in the test set, (2) the original model K classification accuracy, and (3) the algorithm binary classification accuracy. This estimation does not require more details about the task, such as: access to the confusion matrix, or the true positive and negative ratios. In Formula 2, the first and second terms represent the binary and the K classification tasks, respectively. Another form for Formula 2 is Formula 3.

$$\hat{A}_{K+1} \; = \; A_{bin} \cdot OOD \; + \; A_{bin} \cdot \hat{A}_K \cdot ID \tag{2}$$

$$\hat{A}_{K+1} \; = \; A_{bin} \; + \; A_{bin} \cdot ID \cdot (\hat{A}_K - 1) \tag{3}$$

Where:

$\hat{A}_{K+1}$: estimated $K + 1$ classification accuracy

$\hat{A}_K$: K classification accuracy of the original model

A_{bin}: binary classification accuracy

OOD: OOD data ratio

ID: ID data ratio

Proof.

1. Let us denote the number of instances by N. The number of ID instances and OOD instances are $N \cdot ID$ and $N \cdot OOD$ respectively. The decision between ID and OOD leads to a binary classification task. The approximate number of correct decisions among the OOD (TP_{OOD} or TN) and ID instances (TP_{ID} or TP) is expressed by Equations 4 and 5, receptively.

$$TP_{OOD} \; \approx \; N \cdot OOD \cdot A_{bin} \tag{4}$$

$$TP_{ID} \; \approx \; N \cdot ID \cdot A_{bin} \tag{5}$$

2. The correctly classified ID instances are passed to the K-classifier. The original accuracy of the K-classifier ($\hat{A}_K$) is not more than the original model test accuracy (test set consists of ID data only). The number of the true positives in the K-classification task is approximated by multiplying $\hat{A}_K$ by the all number of the instances in the K-classification task (TP_{ID}).

$$TP_K \; \approx \; N \cdot ID \cdot A_{bin} \cdot \hat{A}_K \tag{6}$$

3. The estimated $K+1$ classification accuracy ($\hat{A}_{K+1}$) is the ratio of the correct decisions.

$$\hat{A}_{K+1} = \frac{N \cdot ID \cdot A_{bin} \cdot \hat{A}_K \ + \ N \cdot OOD \cdot A_{bin}}{N} \tag{7}$$

   After using $OOD = 1 - ID$ and simplifying Equation 7, we get Formula 3.

□

Additionally, we prove that the estimated $K+1$ classification accuracy is necessarily larger than w and cannot exceed the algorithm binary classification accuracy (A_{bin}); where $w = A_{bin} \cdot \hat{A}_K$ (Formula 9) and the conditions (Formulas 10, 11 and 12) hold. To do so, we rewrite Formula 2 as Equation 8 by using $OOD = 1-ID$.

$$OOD = \frac{\hat{A}_{K+1} \ - \ w}{A_{bin} \ - \ w} \tag{8}$$

Where:

$$w = A_{bin} \cdot \hat{A}_K \tag{9}$$

$$0 < A_{bin} \leq 1 \tag{10}$$

$$0 \leq \hat{A}_K < 1 \tag{11}$$

$$0 \leq OOD \leq 1 \tag{12}$$

We use Formula 8 to derive the $K+1$ classification accuracy inequality defined in Formula 13.

$$w \leq \hat{A}_{K+1} \leq A_{bin} \tag{13}$$

Proof.

1. Using proof by contradiction, we will prove that the denominator in Formula 8 is always positive ($A_{bin} - w > 0$). Let us suppose the opposite of this statement.

$$A_{bin} - w \leq 0 \tag{14}$$

   a) if $A_{bin} - w = 0$, then OOD is $undefined$, which contradicts Formula 12. Thus,

$$A_{bin} - w \neq 0 \tag{15}$$

b) Let us suppose the following

$$A_{bin} - w < 0 \tag{16}$$

$$A_{bin} < w \tag{17}$$

$$A_{bin} < A_{bin} \cdot \hat{A}_K \tag{18}$$

$$1 < \hat{A}_K \tag{19}$$

but this contradicts Formula 11.
Therefore,

$$A_{bin} - w > 0 \tag{20}$$

2. Using the left side of the inequality (Formula 12) and Formula 20, the numerator cannot be negative ($\hat{A}_{K+1} - w \geq 0$).

$$OOD \geq 0 \tag{21}$$

$$\frac{\hat{A}_{K+1} \ - \ w}{A_{bin} \ - \ w} \geq 0 \tag{22}$$

but $A_{bin} - w > 0$, then,

$$\hat{A}_{K+1} - w \geq 0 \tag{23}$$

$$\hat{A}_{K+1} \geq w \tag{24}$$

3. Using the right side of the inequality (Formula 12) and Equation 8

$$OOD \leq 1 \tag{25}$$

$$\frac{\hat{A}_{K+1} \ - \ w}{A_{bin} \ - \ w} \leq 1 \tag{26}$$

$$\hat{A}_{K+1} \ - \ w \leq A_{bin} \ - \ w \tag{27}$$

$$\hat{A}_{K+1} \leq A_{bin} \tag{28}$$

4. Finally, combining Formulas 24 and 28

$$w \leq \hat{A}_{K+1} \leq A_{bin} \tag{29}$$

□

3.4 Formula for calculating the exact open set classification accuracy

The exact accuracy of the algorithm can be calculated by using Formula 30. It is important to highlight that A_K is the actual K-classification accuracy. It is calculated for the ID instances in the test set and it depends on the actual scenario. In general, it can be assumed that A_K is close to $\hat{A}_K$, in scenarios where the ID data has similar distribution as the original K-classifier test set. Additionally, if the data has only OOD instances ($ID = 0$), Formula 30 cannot be applied, since R_{ID} is undefined ($R_{ID} = \frac{0}{0}$).

$$A_{K+1} \; = \; A_{bin} + R_{ID} \cdot ID \cdot (A_K - 1) \tag{30}$$

Where:

A_{K+1}: exact $K+1$ classification accuracy

A_K: K classification accuracy of the actual model

A_{bin}: binary classification accuracy

R_{ID}: binary True Positive Ratio ($TPR = \frac{TP_{ID}}{TP_{ID}+FN_{ID}}$)

OOD: OOD data ratio

ID: ID data ratio

Proof.

1. Let us denote the number of instances by N. The exact number of correct decisions among the OOD (TP_{OOD}) and ID instances (TP_{ID}) is expressed by Equations 31 and 32, receptively.

$$TP_{OOD} \; = \; N \cdot OOD \cdot R_{OOD} \tag{31}$$

$$TP_{ID} \; = \; N \cdot ID \cdot R_{ID} \tag{32}$$

2. The accuracy of binary classification task A_{bin} comes from the sum of correct decisions divided by the all decisions (Equation 33).

$$A_{bin} \; = \; \frac{N \cdot ID \cdot R_{ID} \; + \; N \cdot (1 - ID) \cdot R_{OOD}}{N} \tag{33}$$

Using Equation 33, R_{OOD} can be expressed by Equation 34.

$$R_{OOD} \; = \; \frac{A_{bin} \; - \; ID \cdot R_{ID}}{(1 - ID)} \tag{34}$$

3. The diagonal entries of the confusion matrix contains the true positive instances in the K-classification task, and the sum of them gives the number of correct decisions, which is equal to accuracy A_K multiplied with all instances in the K classification task (Equation 35).

$$TP_K \; = \; N \cdot ID \cdot R_{ID} \cdot A_K \tag{35}$$

4. The exact $K+1$ classification accuracy (A_{K+1}) is the ratio of the correct decisions.

$$A_{K+1} = \frac{N \cdot ID \cdot R_{ID} \cdot A_K \, + \, N \cdot (1 - ID) \cdot R_{OOD}}{N} \tag{36}$$

5. After substituting the R_{OOD} in Equation 36, and simplification, Equation 36 can be written as Formula 30

□

To determine A_{K+1}, R_{ID} and A_K have to be known. Whereas the estimation ($\hat{A}_{K+1}$) is calculated using $\hat{A}_K$ without the need for R_{ID}. The difference between the exact and the estimated algorithm accuracy is the correction factor (ΔA_{K+1}). It is expressed by Formula 38, which is derived by plugging Formulas 3 and 30 in Equation 37. If $A_K = \hat{A}_K$ and $R_{ID} = A_{bin}$, then $\Delta A_{K+1} = 0$.

$$\Delta A_{K+1} = \Big| A_{K+1} \, - \, \hat{A}_{K+1} \Big| \tag{37}$$

$$\Delta A_{K+1} = \Big| ID \cdot \Big[R_{ID} \cdot (A_K - 1) \, - \, A_{bin} \cdot (\hat{A}_K - 1) \Big] \Big| \tag{38}$$

4 Experiments

In the experiments, we examined and evaluated the four variations of the combined algorithm; ”ThrAndDisc”, ”ThrOrDisc”, ”DpmAndDisc” and ”DpmOrDisc”. Also, we studied the three single algorithms – ”Thr”, ”Disc” and ”DPM” – individually to show their drawbacks and set them as a baseline. One main variable is the threshold value (β) for the ”Thr” algorithm. Hence, we picked and tested different threshold values (with 0.01 step) out of the infinitely many possibilities; $\beta \in [0,1]$. The other variable is the OOD percentage in the test set, since it has a direct relation with the overall algorithm performance (Formulas 2 and 30). Those two variables; i.e. β and OOD ratio, affect the algorithms robustness when dealing with different scenarios. For proper generalization, we used multiple datasets, models (classifiers) and metrics.

4.1 Datasets

We defined ID and OOD datasets; ID data includes instances of K classes, while OOD data is the test set from other datasets. For ID data we used two sets separately, creating two main experiment groups. The first is Extended-mnist (E-MNIST - by merge) dataset [7], where the numbers are excluded, hence, it contains 37 letter categories ($K = 37$). The other is Arabic handwritten characters set (Arab-L) [10], that contains 13440 and 3360 grey-scaled 32x32 pixel images for train and test splits respectively. Those images are distributed evenly over 28 classes, hence, $K = 28$. We augmented the data in order to expand its size. After resizing it into 28x28 pixel, random scaling and rotation was applied resulting in 40 000 train and 10 000 test images. The train sets were used to train two different K-classifiers, and the experiments were conducted with the test sets (Table 2).

Table 2: Datasets size details. E-MNIST and Arab-L datasets are the ID data. K is the number of classes in each dataset.

Dataset	train	test	K
E-MNIST	410 000	10 000	37
Arab-L	40 000	10 000	28

In order to provide better generalization and represent diverse levels of similarity with respect to the ID data, we used six different types for OOD data (Figure 2). We chose the test set of four classical datasets: MNIST dataset [8], Fashion-mnist (F-MNIST) [41], Kuzushiji-mnist (Ku-MNIST) [6] and B-MNIST (we performed binarization for MNIST test set). Additionally, we generated two datasets with Gaussian distribution. R-gauss (28x28 pixel random Gaussian data with 128 mean and 12.8 standard deviation), and I-gauss (ID-based Gaussian data). For the I-gauss we calculated the training data (ID) mean and standard deviation and used them as the distribution parameters. Since we have two ID sets, separate I-gauss data was generated for each of them.

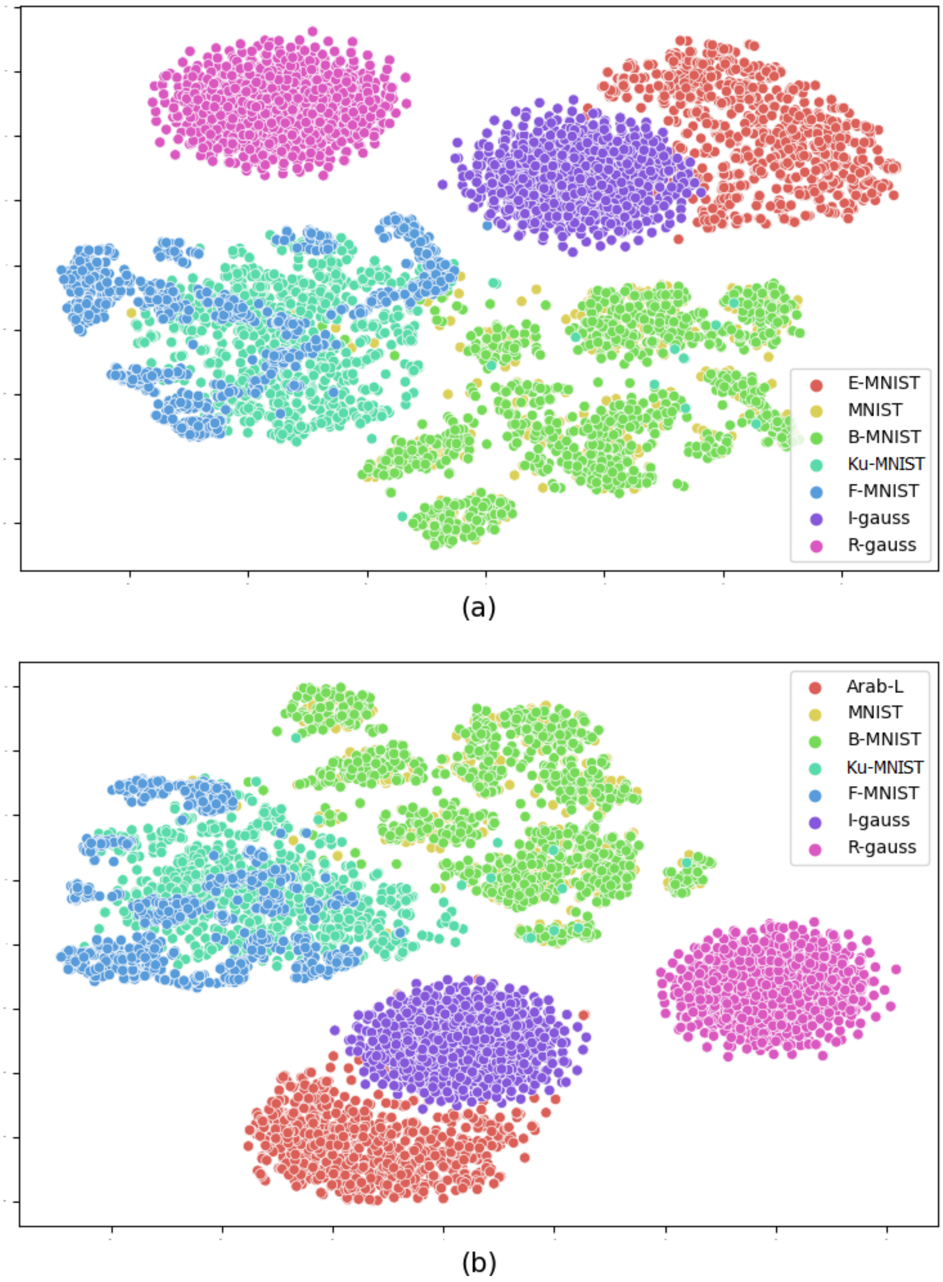
We executed the experiments on mixed test sets, each containing in total 10000 ID and OOD instances. For the test set of each experiment, we selected one ID set and one OOD set. Hence, we chose either E-MNIST or Arab-L data as ID, and combined it with one of the six OOD sets. The percentage of OOD instances was defined by setting the OOD ratio, which varies from 0% to 100% with 5% step.

4.2 Classifiers

For K classification we trained two different classifiers on the two different ID sets. For the first classifier, we used VGG16 [34] architecture with dropout layers. The classifier was trained on E-MNIST data (without numbers data). Since the training data has 37 categories ($K = 37$), it is called "Classifier-37". Similarly, we trained AlexNet [21] on Arab-L dataset. Following the same fashion, this classifier is called "Classifier-28"; $K = 28$. Additionally, we trained a simple GAN on the two ID training sets separately, then we extracted the discriminators to be used as a binary classifier in the proposed "Disc" algorithm accordingly. As mentioned earlier, any arbitrary classifier can fit this purpose.

4.3 Metrics

The first component of our approach is the binary classification, where the suspected OOD instances are filtered out. In case the instance is from ID, it is either correctly classified (TP) or results in type II error (FN). While if it is from OOD,



(a)

(b)

Figure 2: OOD with (a) E-MNIST and (b) Arab-L datasets after dimension reduction using t-SNE [37]. The number of components for t-SNE is 2, and it uses PCA initialization with 50 components.

it is either TN or FP (type I error). Therefore, to evaluate the first step we used sensitivity (TP rate) and specificity (TN rate). Moreover, we used the accuracy (Equation 39) and the balanced accuracy[1] (Equation 40) to measure the performance of the overall classification task [35], because unbalanced data might mislead the conclusion [29]. Lastly, we used mean squared error for evaluating the goodness of Formula 2 using Formula 37 (ΔA_{K+1}), and aggregating these results of all the experiments.

$$Accuracy = \frac{1}{n_{samples}} \sum_{i=1}^{n_{samples}} 1\left(\hat{y}_i = y_i\right) \tag{39}$$

$$Balanced\ Accuracy = \frac{1}{\sum \hat{w}_i} \sum_{i=1}^{n_{samples}} \hat{w}_i\left(\hat{y}_i = y_i\right) \tag{40}$$

5 Results and discussion

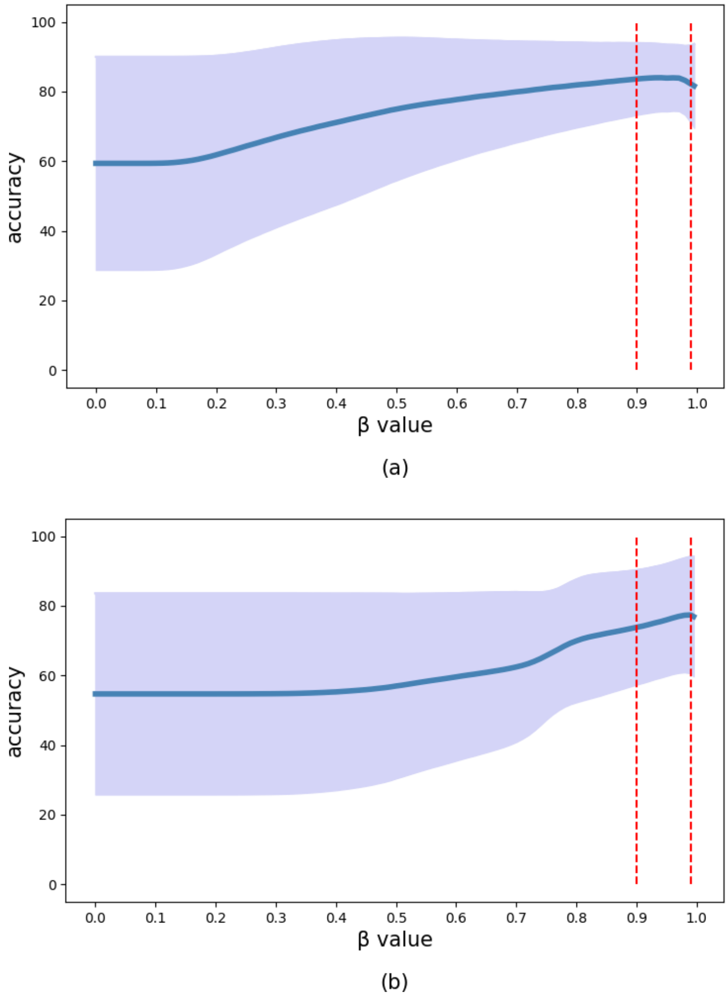
When performing the experiments, the two main variables to deal with are the OOD ratio in the test set, and the β value. Keeping in mind that β value is only relevant for the methods that include ”Thr” algorithm, and it is not arbitrarily chosen. Instead, we defined a range $[0.90, 0.99]$, where the resulted $K+1$ accuracy is the highest with the smallest variance as shown in Figure 3. The aim is to avoid any misleading intuitions that might be caused by relying on K-classifier Softmax confidence [31]. Additionally, we created three groups. In the first group we fixed β value to 0.99, and checked the results over the OOD ratio between 5% and 95%. The other two groups simulate two extreme scenarios; we fixed OOD ratio to 5% and 95% over a range of β values (Table 3).

Table 3: Groups parameters

Parameter	Group 1	Group 2	Group 3
β	0.99	$\in [0.90, 0.99]$	$\in [0.90, 0.99]$
OOD ratio	$\in [0.05, 0.95]$	0.05	0.95

First, we evaluated the proposed algorithms’ ability to execute the binary classification task. In Table 4 it can be seen that ”And” methods, i.e. ”ThrAndDisc” and ”DpmAndDisc”, achieve the highest sensitivity score, alongside with the single ”Disc” algorithm. In other words, they can lead the ID instances to the next stage successfully and attain the highest TP rate. They are well suited in scenarios where avoiding type I error (FP) is vital. In contrast, Table 5 shows that ”OR” methods, i.e. ”ThrOrDisc” and ”DpmOrDisc”, are better fit in scenarios where committing type II error (FN) is more harmful. Sensitivity and specificity results

[1] $\hat{w}_i$ is the sample weight adjusted according to its true class inverse prevalence.



(a)

(b)

Figure 3: The average $K+1$ accuracy (in dark blue) and standard deviation range (in light blue) from all experiments that include "Thr" at β value in range [0, 1]. (a) and (b) plots are for the experiments that use Classifier-37 and Classifier-28, respectively. Red lines define the range of β value where the accuracy is the highest and the standard deviation is the lowest.

(Tables 4 and 5) also highlight some interesting points: (1) despite its simplicity, single "Disc" algorithm executes the binary classification task adequately; (2) the ability to catch OOD instances is in an acceptable range regardless of the algorithm; (3) the methods are more stable in terms of sensitivity (TP rate) compared to specificity (TN rate).

Next, we used the three groups to assess the algorithms capability in accomplishing the $K+1$ classification task, based on their average accuracy results. The experiment results are consistent regardless of the used classifier and the ID data type. Furthermore, second and third group results in Tables 6 and 7 confirm our previous findings w.r.t. sensitivity and specificity. For instance, when OOD ratio is low (Group 2) ”And” methods perform the best, whereas ”OR” methods excel with high OOD ratio (Group 3). Also, the first group demonstrates the overall superiority of ”DpmAndDisc” method in terms of achieved accuracy and stability.

Table 4: Average sensitivity of binary task among all OOD datasets (using Classifier-37 & Classifier-28). In Table 3, Group 1 shows the corresponding experiment parameters. The results are in the form of mean and standard deviation. The highest three sensitivity scores in every group are in bold.

Algorithm	Classifier-37		Classifier-28	
	mean	sd	mean	sd
Disc	**85.81**	0.27	**76.73**	2.42
DPM	65.14	0.45	37.46	2.28
DpmAndDisc	**95.59**	0.24	**84.97**	2.28
DpmOrDisc	55.36	0.48	29.22	2.39
Thr	66.44	0.41	69.98	2.35
ThrAndDisc	**95.54**	0.13	**91.90**	1.23
ThrOrDisc	56.72	0.57	54.81	3.33

Table 5: Average specificity of binary task among all OOD datasets (using Classifier-37 & Classifier-28). In Table 3, Group 1 shows the corresponding experiment parameters. The results are in the form of mean and standard deviation. The highest three specificity scores in every group are in bold.

Algorithm	Classifier-37		Classifier-28	
	mean	sd	mean	sd
Disc	**93.84**	13.70	**100**	0.00
DPM	80.61	12.13	93.16	13.25
DpmAndDisc	75.40	16.00	93.16	13.25
DpmOrDisc	**99.04**	2.12	**100**	0.00
Thr	91.63	9.89	82.53	27.48
ThrAndDisc	85.47	13.53	82.53	27.48
ThrOrDisc	**100**	0.00	**100**	0.00

Additionally, Tables 6 and 7 highlight the instability of single ”Thr” algorithm. It is very sensitive to β value, which is reflected in Group 2 and 3 standard deviation results. Thus, albeit having a high mean value in Group 2, it cannot be concluded that it outperforms the others. For instance, in this scenario single ”Disc” algorithm might be a better choice.

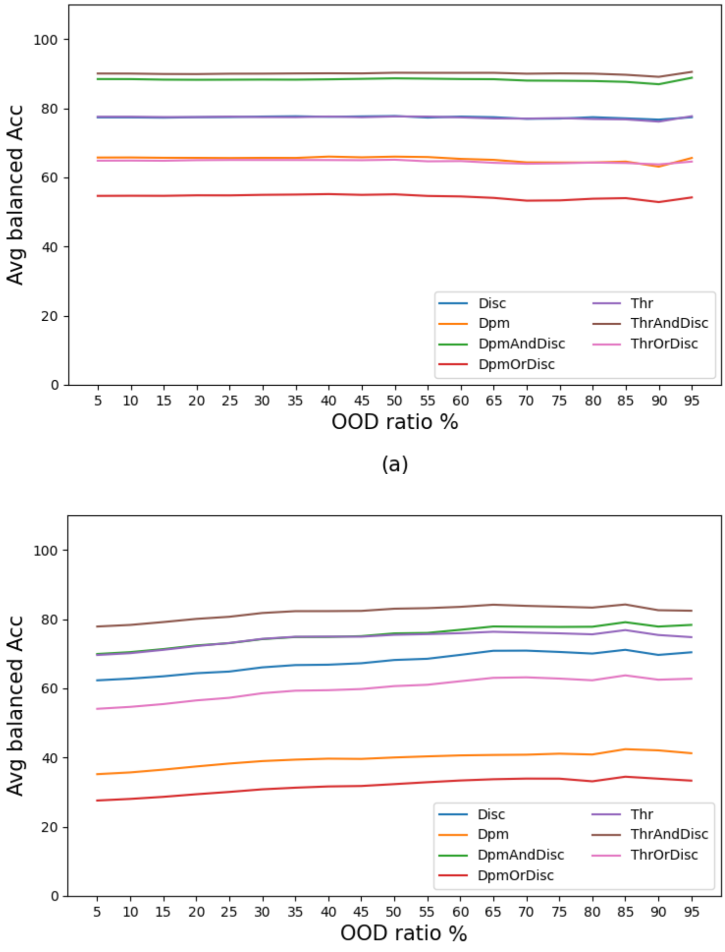
Table 6: Average A_{K+1} (using Classifier-37). The experiments’ parameters are detailed in Table 3. The results are in the form of mean and standard deviation. The highest three accuracy scores in every group are in bold.

Algorithm	Group 1		Group 2		Group 3	
	mean	sd	mean	sd	mean	sd
Disc	**87.39**	8.90	81.57	0.78	**93.15**	14.33
DPM	72.53	8.58	65.65	0.67	78.71	13.73
DpmAndDisc	**83.00**	10.80	**90.11**	0.86	75.09	17.34
DpmOrDisc	76.93	13.46	57.11	0.06	**96.77**	2.48
Thr	78.70	9.68	**86.46**	9.54	48.44	33.75
ThrAndDisc	**87.93**	8.29	**91.26**	1.60	46.06	32.42
ThrOrDisc	78.17	13.29	76.76	8.92	**95.53**	9.19

Table 7: Average A_{K+1} (using Classifier-28). The experiments’ parameters are detailed in Table 3. The results are in the form of mean standard deviation. The highest three accuracy scores in every group are in bold.

Algorithm	Group 1		Group 2		Group 3	
	mean	sd	mean	sd	mean	sd
Disc	**82.08**	12.17	62.87	0.01	**98.52**	0.00
DPM	64.39	19.94	36.12	0.64	90.38	13.94
DpmAndDisc	**82.81**	0.82	**70.28**	0.64	92.25	13.94
DpmOrDisc	63.66	23.06	28.72	0.02	**96.64**	0.00
Thr	73.83	17.34	**74.29**	7.96	28.68	31.52
ThrAndDisc	**80.71**	16.26	**77.52**	1.73	28.84	31.73
ThrOrDisc	75.20	16.49	59.64	6.59	**98.36**	0.36

We evaluated the algorithms further, by investigating a more general scenario. Figure 4 shows their average $K + 1$ balanced accuracy, given that $\beta \in [0.90, 0.99]$. Using balanced accuracy eliminates the effect of OOD ratio and provides broader insight about the algorithms’ performance. This figure gives another evidence of the ”And” methods general effectiveness. In this case, ”ThrAndDisc” algorithm outperforms the others, followed closely by ”DpmAndDisc”. Since the two algorithms include ”Disc” component, they both require access to the training data. Therefore, the main advantage of ”Thr” algorithm, i.e. train data independence, vanishes.



(a)

(b)

Figure 4: The average $K+1$ balanced accuracy for all OOD sets, at OOD ratio value in range [0.05, 0.95] for the experiments using (a) Classifier-37 and (b) Classifier-28.

Moreover, Figure 5 shows the algorithms' average $K + 1$ accuracy results with respect to the OOD set. All the algorithms, regardless of the used K-classifier, performed the best when OOD data was random (R-gauss). With other OOD sets, "Disc" algorithm performance was independent from the OOD data. This behaviour was reflected also on the "OR" methods.

Lastly, Table 8 demonstrates an empirical evidence of our proposed Formula 2. We executed an extensive amount of experiments (∼76 000 experiments[2]) and

[2] total number of experiments = 2 ID sets (classifiers) * 6 OOD sets * 20 OOD ratios * [4 algorithms without "Thr" + 3 algorithms with "Thr" * 100 β values].

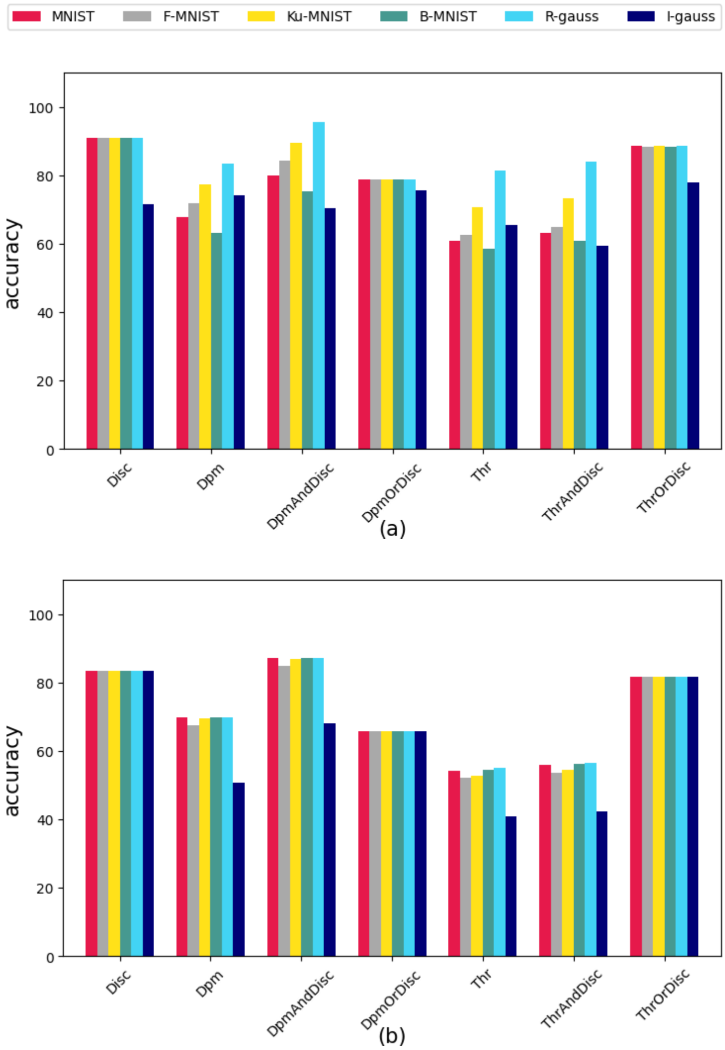
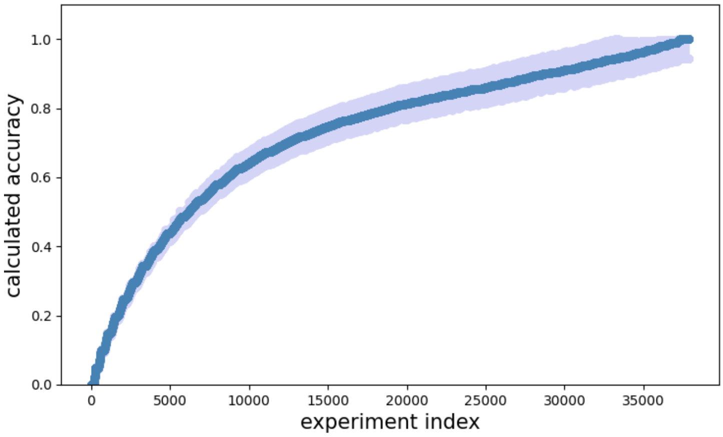


Figure 5: The average $K+1$ accuracy for all algorithms with respect to the selected OOD set for the experiments using (a) Classifier-37 and (b) Classifier-28.

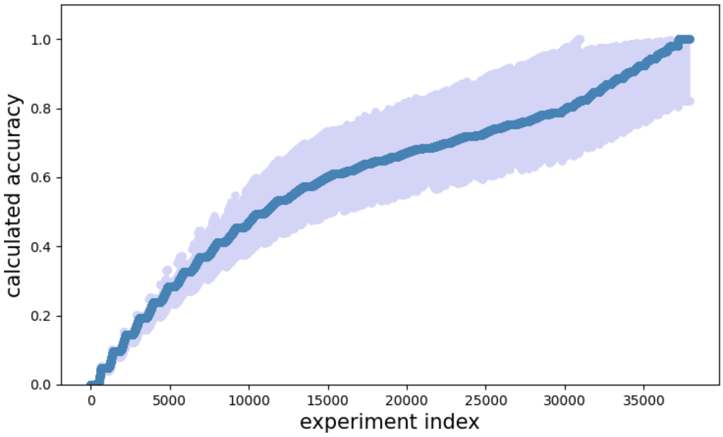
calculated the mean squared error between the actual and estimated $K+1$ accuracy. Additionally, we validated that the proposed inequality (Formula 13) holds in all cases (Figure 6). The estimated $K+1$ accuracy ($\hat{A}_{K+1}$) of the algorithm is higher than or equal to w ($\hat{A}_K \cdot A_{bin}$), but it can not exceed A_{bin}. The inequality highlights that the binary classifier is the vital segment in this architecture (Figure 1). Failing to distinguish ID from OOD data degrades the overall algorithm performance.

Table 8: MSE scores between the actual (by experiment) and the estimated (by formula) $K+1$ accuracy for the algorithms (using Classifier-37 & Classifier-28).

	Classifier-37	Classifier-28
MSE	$1.02 \cdot 10^{-4}$	$6.31 \cdot 10^{-4}$



(a)



(b)

Figure 6: The inequality empirical results sorted by the calculated accuracy value. The dark blue data is the calculated $K+1$ accuracy ($\hat{A}_{K+1}$), that lies between the lower (w) and upper (A_{bin}) bounds in light blue for all the experiments using (a) Classifier-37 and (b) Classifier-28. A plot was used instead of a table, because of the large number of experiments (more than 76 000).

6 Conclusion

In this paper we proposed various approaches to solve the open set classification task for image datasets. By proposing a flexible methodology, we overcome the need for retraining a pretrained K-classifier or altering its architecture. As a result, our proposed methods can adapt to any available classifier.

We interpreted $K+1$ classification task as two consecutive steps: (1) Binary classification; i.e. ID or OOD, followed by (2) K classification. Our proposal handles the first task and lets the original K-classifier to solve the other. We grouped our proposed algorithms based on the decision technique. The first is the single algorithms, where we proposed threshold-based ”Thr” and discriminator-based ”Disc” methods. The second is the combined algorithms, where we built the final judgment based on a collective decision between ”Disc” and a threshold-based method, i.e. ”Thr” or ”DPM”. Their outcomes are joined either by logical ”OR” or ”AND”. As a result, we proposed four variations; ”ThrAndDisc”, ”ThrOrDisc”, ”DpmAndDisc” and ”DpmOrDisc”. After evaluating all methods, the results show that ”DpmAndDisc” and ”ThrAndDisc” algorithms are an excellent general solutions. Additionally, ”And” algorithms are good fit when the priority is to avoid committing type I error (FP), while ”OR” algorithms are more suitable in dealing with higher percentage of OOD instances; avoiding type II error (FN).

Furthermore, we presented mathematical formulas to calculate the exact and estimated $K+1$ accuracy of the algorithm, and used the latter to define an inequality for $\hat{A}_{K+1}$. We proved mathematically and empirically that $\hat{A}_{K+1}$ is equal to or larger than w $(\hat{A}_K A_{bin})$, but it is lower than A_{bin}.

7 Future work

We evaluated our proposal to tackle open set classification task for image datasets from multiple aspects. However, the proposal ability to solve the task for other data types, e.g. text (document) classification, can be shown. Another direction is to investigate the influence of the ID and OOD data characteristic on the proposed solutions performance. For instance, the task is expected to be more challenging with higher similarity between ID and OOD data distribution. Additionally, more experiments can be conducted to analyze how the hyper-parameters (β) tunning is affected by multiple factors, such as the ID and OOD data characteristic and the K-classifier performance (A_K).

References

[1] Adila, D. and Kang, D. Understanding out-of-distribution: A perspective of data dynamics. In *I (Still) Can't Believe It's Not Better! Workshop at NeurIPS 2021*, pages 1–8. PMLR, 2022. DOI: 10.48550/arXiv.2111.14730.

[2] Al-Shouha, M. Two algorithms for not-in-class classification task on image datasets. *13th Conference of PhD Students in Computer Science (CSCS)*, pages 130–134, 2022. URL: https://www.inf.u-szeged.hu/~cscs/pdf/cscs2022.pdf.

[3] Bendale, A. and Boult, T. E. Towards open set deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1563–1572, 2016. DOI: 10.1109/cvpr.2016.173.

[4] Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. DOI: 10.1109/TPAMI.2013.50.

[5] Bishop, C. M. Novelty detection and neural network validation. *IEE Proceedings — Vision, Image and Signal Processing*, 141(4):217–222, 1994. DOI: 10.1049/ip-vis:19941330.

[6] Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., and Ha, D. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018. DOI: 10.48550/arXiv.1812.01718.

[7] Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. EMNIST: Extending MNIST to handwritten letters. In *International Joint Conference on Neural Networks*, pages 2921–2926. IEEE, 2017. DOI: 10.1109/IJCNN.2017.7966217.

[8] Deng, L. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012. DOI: 10.1109/MSP.2012.2211477.

[9] Duchi, J. and Namkoong, H. Learning models with uniform performance via distributionally robust optimization. *arXiv preprint arXiv:1810.08750*, 2018. DOI: 10.48550/arXiv.1810.08750.

[10] El-Sawy, A., El-Bakry, H., and Loey, M. CNN for handwritten Arabic digits recognition based on LeNet-5. In *International Conference on Advanced Intelligent Systems and Informatics*, pages 566–575. Springer, 2016. DOI: 10.1007/978-3-319-48308-5_54.

[11] Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, Volume 70, pages 1126–1135. Proceedings of Machine Learning Research, 2017. DOI: 10.48550/arXiv.1703.03400.

[12] Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189. Proceedings of Machine Learning Research, 2015. DOI: 10.48550/arXiv.1409.7495.

[13] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. DOI: 10.48550/arXiv.1505.07818.

[14] Goodfellow, I., Bengio, Y., and Courville, A. *Softmax Units for Multinoulli Output Distributions.* In *Deep Learning*, chapter 6.2.2.3. MIT Press Cambridge, MA, USA, 2016. URL: http://www.deeplearningbook.org.

[15] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Communications of the ACM*, 63(11):139–144, 2020. DOI: 10.1145/3422622.

[16] Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. DOI: 10.48550/arXiv.1412.6572.

[17] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL: https://openreview.net/forum?id=Sy2fzU9gl.

[18] Hinton, G., Vinyals, O., Dean, J., et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02530*, 2(7), 2015. DOI: 10.48550/arXiv.1503.02531.

[19] Hospedales, T., Antoniou, A., Micaelli, P., and Storkey, A. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020. DOI: 10.48550/arXiv.2004.05439.

[20] Kim, H. and Mnih, A. Disentangling by factorising. In *International Conference on Machine Learning*, Volume 80, pages 2649–2658. Proceedings of Machine Learning Research, 2018. DOI: 10.48550/arXiv.1802.05983, URL: https://proceedings.mlr.press/v80/kim18b.html.

[21] Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, Volume 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[22] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. DOI: 10.1109/5.726791.

[23] Li, H., Pan, S. J., Wang, S., and Kot, A. C. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2018. DOI: 10.1109/CVPR.2018.00566.

[24] Li, Y., Tian, X., Gong, M., Liu, Y., Liu, T., Zhang, K., and Tao, D. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision*, pages 624–639, 2018. DOI: 10.1007/978-3-030-01267-0_38.

[25] Liang, S., Li, Y., and Srikant, R. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017. DOI: 10.48550/arXiv.1706.02690.

[26] Liao, Y., Huang, R., Li, J., Chen, Z., and Li, W. Deep semisupervised domain generalization network for rotary machinery fault diagnosis under variable speed. *IEEE Transactions on Instrumentation and Measurement*, 69(10):8064–8075, 2020. DOI: 10.1109/TIM.2020.2992829.

[27] Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, pages 4114–4124. PMLR, 2019. DOI: 10.48550/arXiv.1811.12359.

[28] Odena, A. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016. DOI: 10.48550/arXiv.1606.01583.

[29] Papp, D. and Szűcs, G. Balanced active learning method for image classification. *Acta Cybernetica*, 23(2):645–658, 2017. DOI: 10.14232/actacyb.23.2.2017.13.

[30] Papp, D. and Szűcs, G. Double probability model for open set problem at image classification. *Informatica*, 29(2):353–369, 2018. DOI: 10.15388/Informatica.2018.171.

[31] Pearce, T., Brintrup, A., and Zhu, J. Understanding softmax confidence and uncertainty. *arXiv preprint arXiv:2106.04972*, 2021. DOI: 10.48550/arXiv.2106.04972.

[32] Segu, M., Tonioni, A., and Tombari, F. Batch normalization embeddings for deep domain generalization. *arXiv preprint arXiv:2011.12672*, 2020. DOI: 10.48550/arXiv.2011.12672.

[33] Shen, Z., Liu, J., He, Y., Zhang, X., Xu, R., Yu, H., and Cui, P. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021. DOI: 10.48550/arXiv.2108.13624.

[34] Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. DOI: 10.48550/arXiv.1409.1556.

[35] Sokolova, M. and Lapalme, G. A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4):427–437, 2009. DOI: 10.1016/j.ipm.2009.03.002.

[36] Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. DOI: 10.48550/arXiv.1412.3474.

[37] van der Maaten, L. and Hinton, G. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9(2579-2605), 2008. URL: https://jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf.

[38] Viola, P. and Jones, M. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Volume 1. IEEE, 2001. DOI: 10.1109/CVPR.2001.990517.

[39] Wang, J., Feng, W., Chen, Y., Yu, H., Huang, M., and Yu, P. S. Visual domain adaptation with manifold embedded distribution alignment. In *Proceedings of the 26th ACM International Conference on Multimedia*, pages 402–410, 2018. DOI: 10.1145/3240508.3240512.

[40] Wang, S., Yu, L., Li, K., Yang, X., Fu, C.-W., and Heng, P.-A. Dofe: Domain-oriented feature embedding for generalizable fundus image segmentation on unseen datasets. *IEEE Transactions on Medical Imaging*, 39(12):4237–4248, 2020. DOI: 10.1109/TMI.2020.3015224.

[41] Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. DOI: 10.48550/arXiv.1708.07747.

[42] Yang, M., Liu, F., Chen, Z., Shen, X., Hao, J., and Wang, J. CausalVAE: Disentangled representation learning via neural structural causal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9588–9597, 2021. DOI: 10.1109/CVPR46437.2021.00947.

[43] Zhang, H., Li, A., Guo, J., and Guo, Y. Hybrid models for open set recognition. In *European Conference on Computer Vision*, pages 102–117. Springer, 2020. DOI: 10.1109/JPROC.2021.3052449.

[44] Zhang, X., Zhou, L., Xu, R., Cui, P., Shen, Z., and Liu, H. Domain-irrelevant representation learning for unsupervised domain generalization. *arXiv preprint arXiv:2107.06219*, 2021. DOI: 10.48550/arXiv.2107.06219.

[45] Zhou, F., Jiang, Z., Shui, C., Wang, B., and Chaib-draa, B. Domain generalization with optimal transport and metric learning. *arXiv preprint arXiv:2007.10573*, 2020. DOI: 10.48550/arXiv.2007.10573.

[46] Zhu, L., Chen, Y., Ghamisi, P., and Benediktsson, J. A. Generative adversarial networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 56(9):5046–5063, 2018. DOI: 10.1109/TGRS.2018.2805286.