PREPRINT of:
K. Bjerge, P. Bodesheim, H. Karstoft: Few-Shot Learning with Novelty Detection.
International Conference on Deep Learning Theory and Applications (DeLTA), pages 340–363, 2024.

https://link.springer.com/chapter/10.1007/978-3-031-66694-0_21
https://dx.doi.org/10.1007/978-3-031-66694-0_21

Few-Shot Learning with Novelty Detection

Kim Bjerge[1 [0000−0001−6742−9504]], Paul Bodesheim[2 [0000−0002−3564−6528]], and Henrik Karstoft[1 [0000−0003−3739−8983]]

[1] Department of Electrical and Computer Engineering, Aarhus University, Finlandsgade 22, 8200 Aarhus N, Denmark
{kbe,hka}@ece.au.dk
[2] Computer Vision Group, Friedrich Schiller University, Ernst-Abbe-Platz 2, 07743 Jena, Germany
paul.bodesheim@uni-jena.de

Abstract. Machine learning has achieved considerable success in data-intensive applications, yet encounters challenges when confronted with small datasets. Recently, few-shot learning (FSL) has emerged as a promising solution to address this limitation. By leveraging prior knowledge, FSL exhibits the ability to swiftly generalize to new tasks, even when presented with only a handful of samples in an accompanied support set. This paper extends the scope of few-shot learning by incorporating novelty detection for samples of categories not present in the support set of FSL. This extension holds substantial promise for real-life applications where the availability of samples for each class is either sparse or absent. Our approach involves adapting existing FSL methods with a cosine similarity function, complemented by the learning of a probabilistic threshold to distinguish between known and outlier classes. During episodic training with domain generalization, we introduce a scatter loss function designed to disentangle the distribution of similarities between known and outlier classes, thereby enhancing the separation of novel and known classes. The efficacy of the proposed method is evaluated on commonly used FSL datasets and the EU Moths dataset characterized by few samples. Our experimental results showcase accuracy, ranging from 95.4% to 96.7%, as demonstrated on the Omniglot dataset through few-shot-novelty learning (FSNL). This high accuracy is observed across scenarios with 5 to 30 classes and the introduction of novel classes in each query set, underscoring the robustness and versatility of our proposed approach.

Keywords: Few-Shot Learning · Episodic training · Novelty detection · Out of distribution · Prototypical network

1 Introduction

In the ever-evolving landscape of machine learning, the quest for more efficient and adaptive models has led researchers to explore innovative paradigms that transcend traditional learning approaches. One such paradigm gaining substantial attention is few-shot learning (FSL), a subfield that addresses the challenge of training models with limited labeled data [41, 37]. FSL stands as a bridge between traditional machine learning, which often requires extensive labeled datasets, and the burgeoning need for systems capable of rapid adaptation to novel tasks or domains.

While FSL has proven to be a promising avenue for handling scenarios with scarce annotated examples, the integration of novelty or outlier detection mechanisms introduces a new layer of sophistication to these models [27]. Novelties, or outliers, represent instances that deviate significantly from the known data distribution, posing a unique set of challenges and opportunities [26]. This integration allows models not only to recognize patterns within familiar classes but also to discern and adapt to unforeseen and instances of novel classes.

The convergence of FSL and novelty detection addresses the critical need for models that can generalize effectively beyond the confines of their training data. As an example, the few-shot classification problem extended with novelty detection could be used in biology to identify unseen or rare animal species [9, 2]. In many real-world applications, the ability to detect and adapt to novel situations is paramount, such as anomaly detection in healthcare diagnostics [21] for recognizing and classifying images in categories of known and unknown symptoms. This paper explores the theoretical foundations, methodologies, and applications of FSL with a specific focus on incorporating novelty or outlier detection mechanisms.

Contribution. The paper endeavors to present a novel methodology designed specifically for identifying outlier samples in the realm of few-shot-novelty learning (FSNL). This latter objective is geared towards the detection of samples from novel classes not encompassed within the support set for FSL.

To further enhance the efficacy of FSL with novelty detection, we introduce a novel scatter loss function. This function is strategically applied during episodic training to proficiently disentangle and separate the distribution of known and novel classes within the similarity space. By doing so, our proposed approach aims to augment the discriminative capabilities of FSL, specifically in scenarios involving the identification of novel instances that lie outside the boundaries of the support set.

2 Related Work

2.1 Few-Shot Learning

The landscape of FSL methods is multifaceted, comprising diverse approaches tailored to address the challenges inherent in learning from limited labeled data. Among the pioneering methods, Matching Networks, as introduced by Vinyals et al. [38], represent an early foray into this domain. Relational networks is another simple and flexible FSL method by Sung et al. [35]. In accordance with the test setting, few-shot learning can be categorized into two distinct branches: *inductive FSL* and *transductive FSL*. The former involves the prediction of test samples individually, while the latter addresses the prediction of test samples collectively. As substantiated in earlier studies [16, 42], transductive inference consistently outperforms inductive inference, particularly when confronted with scenarios of limited training data [23, 14, 18, 28, 10, 5]. For a comprehensive overview of the evolving FSL landscape, we refer to surveys such as those presented in works by Wang et al. [41] and Song et al. [34].

A notable category of methods relevant to our proposed approach leverages cosine similarity as the fundamental measure. This includes Prototypical Networks [33],

Finetune [7], Transductive Information Maximization [5], and Prototypical Rectification [22]. The utilization of cosine similarity in these methods aligns with the approach adopted in our proposed methodology.

Meta-learning is a key paradigm in the FSL landscape, and a comprehensive survey on the subject is presented by Hospedales et al. [13]. Additionally, the integration of episodic training for domain generalization, as discussed by Li et al. [20], emerges as a crucial aspect in enhancing the adaptability and robustness of FSL models.

In the realm of recent advancements, ProFeat [17] and SAPENet [15] stands out as some of the newest and top-performing FSL methods. This continuous evolution and diversification of FSL methods underscore the dynamic nature of this field, emphasizing the ongoing quest for more effective approaches to tackle the challenges posed by FSL scenarios.

2.2 Novelty and Outlier Detection

Several research areas have contributed to addressing challenges related to rare events, anomalies, novelties, and outliers. The following studies explore various aspects of these issues, placing a particular focus on supervised and unsupervised classification [32, 12]. Out-of-distribution (OOD) detection, which separates in-distribution (ID) and OOD data, has gained attention in machine learning [8]. Recognizing instances of uncertainty or novelty holds significant importance in various deep learning applications, particularly within the domains of limited data for certain classes. Anomaly detection, interchangeably referred to as outlier detection or novelty detection [27, 26], serves as a crucial tool in achieving this objective by identifying patterns in data that deviate from the anticipated norms based on prior observations.

Carreno et al. [6] aims to clarify the distinctions among rare events, anomalies, novelties, and outliers, organizing these concepts within the framework of supervised classification. Novelty detection is crucial for a robust classification system. Markou and Singh [24] emphasizes the identification of new or unknown data during training, as test data may contain information about objects not present during the model's training.

Amarbayasgalan et al. [1] proposing deep autoencoders with density-based clustering (DAE-DBC). This approach involves calculating compressed data and error thresholds from a deep autoencoder model. Points not associated with any clusters are considered novelties. The review by Ruff et al. [30] identifies common underlying principles and assumptions implicit in various anomaly detection methods. It establishes connections between classic 'shallow' approaches and novel deep learning methods.

In many deep semi-supervised approaches, the initial step involves modeling normal behavior, followed by the subsequent identification of novelties [36]. The distribution of normal observations is acquired by leveraging the output scores provided by the trained model. Subsequently, a threshold rule is applied to designate samples as novelties with scores falling below a predefined threshold situated outside the learned distribution.

In situations where acquiring sufficient data for training models to classify all samples is impractical, we propose the incorporation of novelty detection within the FSL model to discern uncertain samples of novel classes. The approach employed in our work, referred to as threshold-based outlier tagging, leverages the concept of identifying outliers based on a predetermined learned threshold for cosine similarity scores.

3 Method

3.1 Few-Shot Learning

In few-shot classification, we are given a small support set of $(N \cdot K)$ labeled examples $S = \{(X_1^{(1)}, y_1), ..., (X_N^{(K)}, y_K)\}$ where each $X_n^{(k)} \in \mathbb{R}^D$ is the D-dimensional embedded feature vector of an example and $y_k \in \{1, ..., K\}$ are the corresponding labels. The set $S_k = \{(X_1^{(k)}, y_k), ..., (X_N^{(k)}, y_k)\}$ denotes the subset of examples labeled with class k and the number of class labels in the support set is denoted K-way where we have N-shots of examples in each S_k. A query set Q contains samples q_i that belong to classes in the support set where the goal is to match the query samples to the correct class label. When testing with a test dataset that comprises more classes than those included in the support set, a random subset of K classes is selected for each few-shot task, which encompasses both the support and query sets.

3.2 Few-Shot-Novelty Learning

In few-shot classification with novelty detection, the query set contains samples of K-way and M-novel classes. The classes denoted as M-novel do not pertain to any of the classes within the support set and, therefore, should be classified as outliers. A solution to this problem would benefit many real-life applications where the number of samples for each class is sparse or non-existent.

Given that novelty detection methods cannot differentiate between the M-novel classes and treats them collectively as a substantial outlier class, the M-novel classes in the support set will be classified as a single novel class. A task contains a labeled support set with K classes and a query of samples from those K classes and M novel classes selected randomly from the dataset. Each dataset is split into three sub-datasets with different classes for episodic training, validation and final testing. The query set only contains M novel classes during the final testing.

3.3 Prototypical Network with Outlier Detection

We propose to use the Prototypical Network [33] and modify it to use the cosine similarity function instead of the Euclidean distance when comparing the support center point with query embeddings. The cosine similarity is normalized between -1.0 and 1.0 in contrast to the Euclidean distance that lies between zero and infinite. We anticipate the cosine similarity to be more efficient in learning a threshold to discerning between similar and outlier samples.

The support center point $\overline{X}^{(k)}$ is computed as the average of N-shot support embeddings for each class

$$\overline{X}^{(k)} = \frac{1}{N} \sum_{n=1}^{N} X_n^{(k)} \tag{1}$$

The cosine similarity between the query sample q_i and support center point $\overline{X}^{(k)}$ is defined as

$$s_{q_i k} = \frac{q_i^T \cdot \overline{X}^{(k)}}{|q_i||\overline{X}^{(k)}|} \tag{2}$$

Table 1. Illustrates cosine similarities between support centers and queries, with two samples per class in the query. The predicted labels for the queries are correct, except for the second row, where the predicted label erroneously reads 2 instead of 0. For the correct predictions the support center similarities are grouped in true positive (known) $p(s|\omega_k) = [0.9, 0.9, 0.8, 0.7, 0.8]$ and true negative (outlier) $p(s|\omega_o) = [0.7, 0.5, 0.7, 0.4, 0.6, 0.5, 0.5, 0.4, 0.5, 0.3]$.

Support center similarities			Query	Predicted
Class 0	Class 1	Class 2	labels	labels
0.9	0.7	0.5	0	0
0.6	0.5	**0.8**	0	**2**
0.7	**0.9**	0.4	1	1
0.6	**0.8**	0.5	1	1
0.5	0.4	**0.7**	2	2
0.5	0.3	**0.8**	2	2

The predicted class $\tilde{y}_i$ is estimated as the maximum of cosine similarities for query embedding q_i and all support center points $\overline{X}^{(k)}$ for all K-way classes

$$\tilde{y}_i = \underset{k \in \{1,..,K\}}{\operatorname{argmax}} \left(s_{q_i k}\right) \tag{3}$$

A prediction is classified as a novelty or outlier if the cosine similarity is below a learned threshold th:

$$\tilde{y}_i = \begin{cases} \text{novelty} & \text{if} \quad s_{q_i k} < th \\ \tilde{y}_i & \text{otherwise} \end{cases} \tag{4}$$

3.4 Learning Threshold for Novelty Detection

All datasets are divided into three sets used for training, validation and testing each containing different classes. In this context, we have decided to utilize the validation dataset to learn a threshold for novelty detection, as it is also used during episodic training to determine the optimal model. Throughout the learning process, the support and query sets undergo changes, with random classes selected for each task from the validation dataset. Each task with a support and query set without any novel classes is used during the learning process. This setup is utilized to estimate a threshold that distinguishes between similarities of samples in the query and support set.

To automate the learning of a threshold (th) value to detect outliers as defined in Eq. (4) we use the validation dataset to learn the probability function for correct true positive (TP) and correct true negative (TN) as shown in Fig. 1. For every sample in the query that is correctly classified according to Eq. (3), the cosine similarities are computed with every support center point using Eq. (2). A query sample, not belonging to

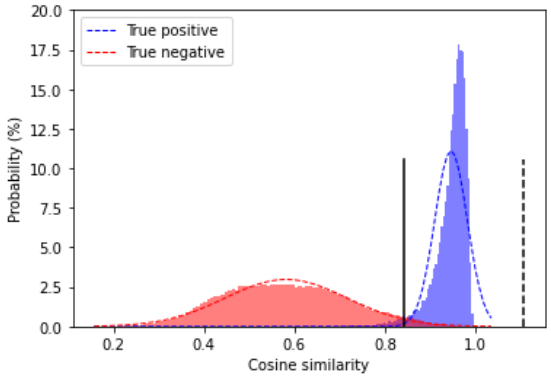


Fig. 1. Distribution of cosine similarities for true positive and true negative predictions learned from the validation dataset. True positive shows the probability function $p(s|\omega_k)$ of correct classified samples. True negatives show the probability function $p(s|\omega_o)$ of samples that are correctly classified as outliers. The black line shows the learned Bayes threshold to separate outliers from correctly classified samples. The dotted black line shows the second solution to the quadratic equation in Eq. (10).

the sample class in the support set, will be marked as an outlier similarity. The correctly classified (TP) cosine similarities (s) are grouped in the distribution $p(s|\omega_k)$ of known similarities. The correctly classified outliers (TN) are grouped in the distribution $p(s|\omega_o)$ of outlier similarities. An example with 3-way and two query samples is shown in Tab. 1. This entails that every sample in a query will undergo a cosine similarity comparison with all samples in the support set.

By using the Bayes decision function $d_j(s) = P(\omega_j|s)$ for classes of known (ω_n) and outlier (ω_o) similarities we have

$$d_j(s) = p(s|\omega_j)P(\omega_j) \tag{5}$$

where $p(s|\omega_j)$ is the probability given by the density function for the known or the outlier similarities and $P(\omega_j)$ is the prior probability for ω_n or ω_o.

The probability density functions $p(s|\omega_j)$ for the patterns in each class (TP, TN) are assumed to be Gaussian $\mathcal{N}(\mu, \sigma^2)$. The probability of occurrence $P(\omega_j)$ of each class, must be known. In the setting of FSNL with a query set of K known classes and M outlier/novel classes we have

$$P(\omega_k) = \frac{K}{K+M} \tag{6}$$

$$P(\omega_o) = \frac{M}{K+M} \tag{7}$$

here $P(\omega_k)$ is the probability for occurrence of a known class and $P(\omega_o)$ is the probability for occurrence of an outlier/novel class. Calculating these prior probabilities accurately is only feasible in our controlled experiments of FSNL; however, estimating them correctly in practice is likely to be challenging.

The Bayes decision boundary between two classes is defined by a single point s_d, such that

$$d_o(s_d) = d_k(s_d) \quad . \tag{8}$$

Here, d_o is the decision function for the outlier/novel class and d_k is the decision function for known classes. Here we have only one decision function for all known classes and one for all outlier/novel classes. The point s_d is the intersection of the two probability functions if the two classes were equally likely to occur, however for FSNL we use the probability of occurrence as defined in Eq. (6) and Eq. (7). Assuming a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ for $p(s|\omega_k)$ and $p(s|\omega_j)$ and combining Eq. (8) and Eq. (5), we get

$$\begin{aligned} &\frac{1}{\sqrt{2\pi\sigma_o}} \exp^{-\frac{(s_d-\mu_o)^2}{2\sigma_o^2}} P(\omega_o) \\ &= \frac{1}{\sqrt{2\pi\sigma_k}} \exp^{-\frac{(s_d-\mu_k)^2}{2\sigma_k^2}} P(\omega_k) \end{aligned} \tag{9}$$

here the variances (σ_o^2,σ_k^2) and means (μ_o,μ_k) are estimated on the validation dataset for the correct classified classes (TP) and correct classified outliers (TN) as illustrated in Fig. 1. The point s_d of the decision boundary is found by solving Eq. (9), where we have

$$\begin{aligned} &(\sigma_k^2 - \sigma_o^2)s_d^2 - 2(\mu_o\sigma_k^2 - \mu_k\sigma_o^2)s_d \\ &+\sigma_k^2\mu_o^2 - \sigma_o^2\mu_k^2 - 2\sigma_o^2\sigma_k^2 ln(\frac{M\sqrt{\sigma_k}}{K\sqrt{\sigma_o}}) = 0 \end{aligned} \tag{10}$$

The solution to the quadratic equation 10, ensuring that s_d falls within the valid range $s_d \in \mathbb{R} \cap [0; 1]$ of the cosine similarity function, is ultimately selected as the learned threshold ($th = s_d$). The proof of Eq. (10) can be found in Appendix A.

3.5 Episodic Training with Scatter Loss

In FSL episodic training with domain generalization (DG) [20] the goal is to train a model on a base domain containing data-label pairs that generalize well to a validation domain with different statistics to the base domain. That means the categories of classes in the training and validation datasets are different. Episodic training is also called meta-learning [13] where we have a set of tasks for each epoch of training. A task contains a number of episodes that each contain a labeled support set and a query set from the training dataset. After each epoch the validation dataset is used to evaluate accuracy on a different domain with other tasks of class categories than contained in the training dataset.

A Prototypical Network is using the Euclidean distance as a similarity function to predict query samples' relation to class labels in the support set. Experiments have shown that training with the Euclidean distance instead of the cosine similarity function gives the best results [20]. During training in our approach the distribution of outlier and known class similarities is controlled by a scatter loss. The scatter loss is designed to minimize within-class variance while simultaneously maximizing the mean separation between outlier and known classes. This method draws inspiration from the work of

Bodesheim et al. [3]. See Fig. 1 for an example of outlier and known class distribution. The univariate scatter loss is defined as

$$J_s(\theta) = \frac{\sigma_k + \sigma_o}{|\mu_k - \mu_o|} \tag{11}$$

here σ_k and σ_o are the standard deviation and μ_k and μ_o are the mean for the distribution of known and outlier similarities using Euclidean distances in the training episode. Only sample queries that are correctly classified according to training label are included in the distribution of known and outlier similarities. The cross-entropy loss function in Eq. (12) ensures that query samples are classified correct according to the support set during training:

$$J_c(\theta) = -\sum_{j=1}^{K} \hat{y}_j \log(\frac{exp(d_{q_j k})}{\sum_{i=1}^{K} exp(d_{q_i k})}) \tag{12}$$

Here, $d_{q_j k}$ is the euclidean distance between the support center point and the query sample j and $\hat{y}_j$ is the one-hot encoded vector for the correct label of the query sample. K is the number of classes (K-way) in the support set. Finally a combined loss function is defined to prioritize between the cross-entropy and scatter loss:

$$J(\theta) = \alpha J_s(\theta) + (1 - \alpha) J_c(\theta) \quad . \tag{13}$$

The goal is to increase the distance between the known and outlier distributions while increasing the correct classified query samples related to the support classes. The loss function $J(\theta)$ will prioritize between minimizing the scatter loss and the cross-entropy loss by adjusting $\alpha \in [0, 1]$

3.6 Fine-Tuning and Episodic Training

The convolutional neural network ResNet [11] is used as a backbone to extract feature embeddings. The output features from the last convolutional layer are flattened and used as embedding with a D-dimensional feature vector. ResNet12 and ResNet18 are commonly used in FSL for training [7, 5, 41] producing a 64 and 512-dimensional feature vector, respectively. In our work, we have trained models with and without fine-tuning. In fine-tuning classical pre-trained weights on the ImageNet dataset [31] are used. The pre-trained ResNet models were fine-tuned with episodic training on a new domain dataset. A 5-shot 5-way support set was used during all episodic training and validation sessions. However, the number of shots varied in the experiments conducted during the evaluation of the trained models.

The stochastic gradient descent optimizer (SGD) was used during training. The SGD was configured with the momentum of 0.9 and a weight decay of $5.0 \cdot 10^{-4}$ using a multi-step scheduler to lower the learning rate at two specified milestones specified by epochs. The first milestone was set to 120 epochs and the second to 190 or 250 epochs. SGD was tested with the initial learning rate of $1.0 \cdot 10^{-3}$ for pre-trained models and $1.0 \cdot 10^{-2}$ for trained models without fine-tuning.

A number of additional arguments were specified for each model to be trained covering: ResNet model, dataset, epochs, milestone 1, milestone 2, α, pre-trained weights, training tasks, validation tasks, query number and initial learning rate. The arguments for model trained and validated in our experiments are detailed in Appendix B.

3.7 Performance Metrics for Testing

The accuracy for different learning methods on the test dataset is calculated to compare few-shot and few-shot-novelty Learning with the commonly used 5-way (classes) with either a 5-shot or 1-shot support set. FSNL is evaluated with 1-novelty class in most experiments, however with a new novelty class in every query during testing. The experiments also explore the impact of varying the number of M-novel classes and the variation of K-way classes in both the support and query sets.

To evaluate model performance, the precision, recall, and $F1$-score metrics were chosen for the novelty class. These metrics are based on true positive (TP), false positive (FP), and false negative (FN) novelty detections. Recall and precision were used in conjunction to obtain a complete picture of the model's ability to find all novelties and detect them correctly. To balance precision and recall, we used the $F1$-score.

The metrics outlined in the tables of Sec. 5 include Acc. (FSL) representing the accuracy for few-shot learning, and Acc. (FSNL) denoting the accuracy for few-shot-novelty learning, along with precision, recall, and F1-score for the novelty class. Metrics in all tables are computed across 5 random runs, with average and standard deviations (SD).

4 Experimental Setup for Training, Validation and Testing

The proposed method undergoes training and evaluation using four distinct datasets, with three of them being widely employed in the realm of FSL. Each dataset is partitioned into three distinct class domains designated for training, validation, and ultimate testing. The validation dataset plays a pivotal role in episodic training, contributing to the selection of the optimal model and facilitating the determination of the novelty threshold during the learning process. Evaluation of performance metrics is exclusively conducted on the test datasets. Importantly, these test datasets encompass classes that were not part of the training or validation phases. During the final testing 500 tasks of support and query sets were randomly selected from the test datasets for final evaluation in each run.

This deliberate inclusion ensures a rigorous assessment of the proposed method's capability to generalize effectively to new and unseen classes, a critical aspect in the validation of its FSNL prowess.

4.1 Dataset

The Omniglot dataset [19] contains 1623 different handwritten characters from 50 different alphabets. Each of the 1623 characters was drawn online via Amazon's Mechanical Turk by 20 different people. The image size is 28x28 pixels. Images for evaluation are split into 40 classes for validation and 40 classes of handwritten characters for testing. ResNet12 models without pre-trained weights are trained on the Omniglot dataset.

MiniImageNet is a benchmark dataset and is a subset of the larger ILSVRC-12 dataset [31]. It has a total of 60,000 color images from 100 classes, where each class has 600 images of size 224x224 [38]. In alignment with established practices [29, 40],

we adopt a partitioning scheme consisting of 60 base classes, complemented by 20 validation classes and 20 test classes. ResNet18 models without pre-trained weights are trained on the miniImageNet dataset.

Caltech Birds-200-2011 (CUB) [39] is a fine-grained image classification dataset. We adopt Chen et al. [7] for few-shot classification on CUB, which splits into 120 base, 40 validation and 40 test classes for the experiments. To maintain consistency and comparability with miniImageNet, the images from CUB are uniformly resized to 224x224 pixels. ResNet18 models with pre-trained weights are trained on the CUB dataset.

The EU Moths dataset, as introduced by Böhlke et al. [4], encapsulates a collection of 200 moth species prevalent in Central Europe. Notably, each species is delineated by few samples, comprising a mere 11 images, resulting in a total of 2205 images within the dataset. The images have high resolution, surpassing 1000x1000 pixels, however resized to 224x224 pixels. The dataset is split into 100 base, 50 validation and 50 test classes for experiments. ResNet18 models with pre-trained weights are trained on the EU Moths dataset.

5 Experiment and Results

Several experiments were performed on the Omniglot dataset to evaluate FSL with novelty detection and episodic training with the scatter loss function. The presented method was also explored with and without fine-tuning. Here, results were obtained with fine-tuning on the CUB and EU Moths datasets. The dataset miniImageNet was evaluated with episodic training without pre-trained weights comparing the performance of few-shot and few-shot-novelty learning methods. All the experimental source code is available at Github[1].

5.1 Threshold Learning for Novelty Detection

The Omniglot dataset was used to evaluate the automated learning of a threshold to detect outliers. A ResNet12 model was trained on the Omniglot dataset with episodic training using the combined cross-entropy and scatter loss with $\alpha = 0.8$. This value was chosen since higher values of α give a slight increase in performance, which is evaluated in Sec. 5.2. The threshold was learned for 5-shot images with 10-way classes in the support set. The learned threshold was evaluated on the test dataset with a query of 10-way known classes and 1-novel class. The threshold was systematically adjusted to assess the proximity of the learned threshold at 100% compared to an optimal value, determined by the highest F1-score. Here, the threshold was varied with a percentage of 97% to 103% relative to the learned value as described in Sec. 3.4 assuming 100% would give the highest F1-score.

Figure 2 shows that the learned threshold is very close to the crossing of the precision and recall curve at 99.8%. However, the highest F1-score for the detection of the novelty class is at 101%. The learned threshold at 100% seems to be close to optimal even if it was learned on the validation dataset with other classes giving another

[1] https://github.com/kimbjerge/few-shot-novelty

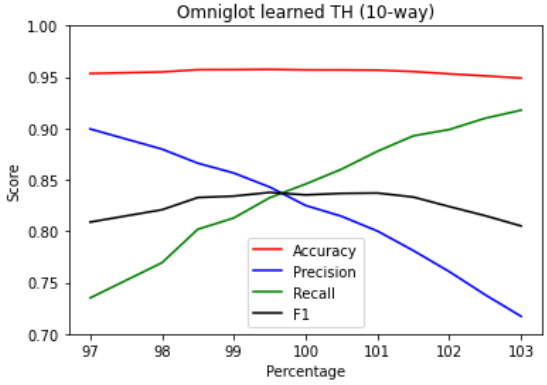


Fig. 2. Results for learning the Bayes threshold on the Omniglot dataset with 10-way and 5-shot images in the support set. The plot shows the few-shot-novelty accuracy on test episodes with precision, recall and F1-score for the novelty class. Average of metrics are computed across 5 random runs. The percentage variation of the learned Bayes threshold (TH) is shown on the x-axis. A percentage of 100 is the learned TH, which is very close to the crossing of precision and recall curves.

known/outlier distribution than contained in the test dataset. Adjusting the threshold above or below the learned threshold will prioritize either precision or recall to be the most important metrics to optimize. A threshold above the learned value increases the recall but lower the precision. To prioritize the precision the learned threshold must be decreased. The F1-score will however decrease by up to 0.05 compared to the optimal threshold at 100%. The same pattern for the learned threshold was observed for 5, 10, 20 and 30-way of classes in the support set. However, the overall F1-score decreases with an increasing number of classes as shown in Fig. 5b. With more classes increased from 5-way to 35-way in the support set, it seems that the efficiency of the outlier detector decreases with an F1-score from 0.9 to 0.7 with $\alpha = 0.8$.

Figure 3 depicts the variation in FSNL accuracy and F1-score as the number of novel classes (M-novel) in the query set increases. As the number of novel classes rises, the overall F1-score demonstrates improvement for the novelty classes. However, the overall accuracy exhibits a decline until the query set is predominantly dominated by the number of novel classes. The same tendency was observed on the EU Moths dataset as documented in Appendix C. These observations suggests that the learned threshold effectively detects and distinguishes between novel and known samples in the support set.

We also aimed to explore the sensitivity of the learned threshold concerning the prior probabilities ($P(\omega_o), P(\omega_k)$) in Eq. (10), which are determined by the ratio M/K. Initially, the threshold was learned assuming equal prior probabilities, specifically by setting the ratio $M/K = 1$. Subsequently, we compared the precision, recall, and F1-score for novel classes when learning the threshold with correct M/K ratio. Two tests were conducted: one with a single novel class ($M = 1$) and an increasing number of K-way classes, as illustrated in Fig. 4a, and another with 5-way and an increasing number of M-novel classes, as depicted in Fig. 4b. There was a notable performance difference

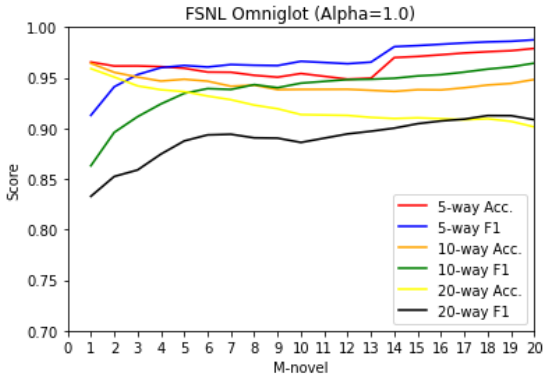
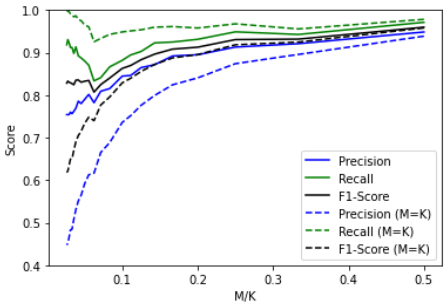
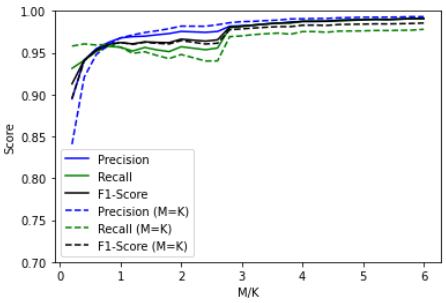


Fig. 3. Results of FSNL (trained with $\alpha = 1.0$) on the Omniglot dataset with varying numbers of novel classes (M-novel), fixed with either 5-way, 10-way, or 20-way, and with 5-shot images in the support set. The plot shows the few-shot-novelty accuracy on test episodes and F1-score for the novelty classes.

observed when there was only one novel class and a substantial number of K-way classes in the support set ($M/K < 0.1$). In the Appendix C, a parallel experiment was conducted on the EU Moth dataset, demonstrating a performance difference for many novel classes and 5-way ($M/K > 2$). This underscores the importance of utilizing an estimate of prior probabilities, particularly when the disparity between novel (M) and known classes (K) is high in the query set. In practice, we suggest starting with the ratio $M/K = 1$ and then evaluating and estimating the prior probabilities on a selected dataset of samples.



(a) Threshold sensitivity for one ($M = 1$) novel class and K-way.



(b) Threshold sensitivity for 5-way and M-novel classes.

Fig. 4. Shows the precision, recall and F1-scores for different relative probabilities of M/K on the Omniglot dataset. The dotted lines shows when threshold is learned assuming equal prior probabilities ($M = K$).

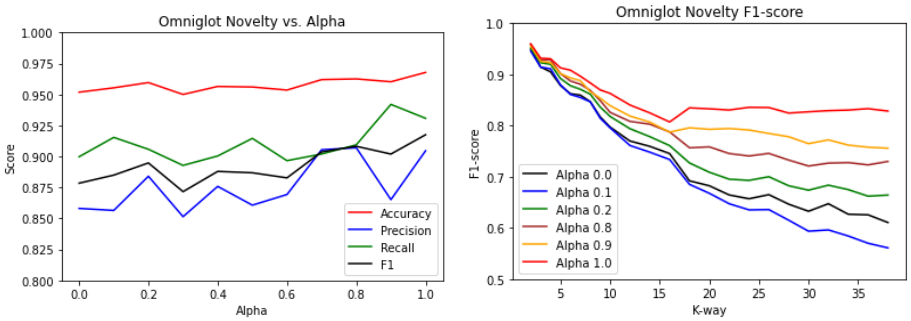
5.2 Episodic Training with Scatter Loss

In total 11 models were episodically trained on the Omniglot dataset with a variation of α with steps of 0.1 in the range of $\alpha \in [0, 1]$. The learned threshold on the validation

dataset was used to test the few-shot-novelty performance on a support set with 5-way and 5-shot images.

Figure 5a shows an improved performance for the overall accuracy of few-shot-novelty classification and F1-score of the novelty class with increased α values. However, the performance improvement with $\alpha = 0.0$ to $\alpha = 1.0$ was minimal with an F1-score of 0.878 increased to 0.917 and accuracy of 0.952 increased to 0.968. It is interesting that the training manages to succeed with $\alpha = 1.0$, since only the scatter loss function Eq. (11) will be contributing to optimizing the weights during backpropagation. It also shows that incorporating cross-entropy loss leads to worse performance on the Omniglot dataset. The scatter loss function tries to separate the distribution of known and novel classes and only indirectly ensures that query predictions are correctly classified to samples in the support set.

The threshold was learned for each trained model with different α values. It is noteworthy that the threshold is contingent upon the specific trained model with the employed α values. A threshold was learned on the 11 different trained models for each variation of K-way classes in the support set. The models trained with the three lowest and three highest α values are shown in Fig. 5b. Each curve represents a model trained with different α values and the F1-score was measured on few-shot-novelty classification with varying the K-way classes. It shows that the increase of α value improves the F1-score significantly when there are many classes (K-way) in the support set. However, for α values $(0.3 - 0.7)$ not shown in Fig. 5b the variation is high for values above 15-way. This indicates that there could be other factors that have an impact on the episodic training resulting in the variation between training sessions.



(a) FSNL accuracy with precision, recall and F1-score for the novelty class (5-way).

(b) F1-score for the novelty class with different numbers of classes (K-way) and values of α.

Fig. 5. Result plots for training models with different α values on the Omniglot dataset with 5-shot images in the support set.

Model performance is shown in Tab. 2 with cross-entropy loss ($\alpha = 0$) and scatter loss ($\alpha = 1.0$) trained on the Omniglot dataset. Average and standard deviations (SD) are computed across 5 runs with different random generated support and query set selected from the test dataset. The result shows that the accuracy improves for FSL with scatter loss, especially for 30-way 5-shot where the accuracy was increased from 0.950 to 0.962. The original prototypical network published in [33] has a bit higher 5-way accuracy with 0.988 for 1-shot and 0.997 for 5-shot FSL on the Omniglot dataset. This

higher accuracy is because they used more classes in the support set during episodic training. We get a similar accuracy of 0.981 for 1-shot and 0.994 for 5-shot when increasing the number of classes from 5-way to 20-way during training.

Few-shot-novelty has a 0.025 lower average accuracy compared to few-shot classification. This is due to a relatively lower precision and recall for the novelty class. The F1-score for the novelty class is significantly higher with scatter loss. For a support set comprising 30 classes (30-way, 5-shot), the F1-score decreases from 0.828 to 0.637, representing a reduction of 0.191. It is observed that the standard deviation (SD) varies and is higher for recall than precision and the SD recall is lowest with $\alpha = 1.0$.

Table 2. Shows the performance metrics for few-shot and few-shot-novelty classification with different α trained on the Omniglot dataset with 5-shot and 1-shot.

Metric	Shot (α)	5-way Avg (SD)	10-way Avg (SD)	20-way Avg (SD)	30-way Avg (SD)
Acc. (FSL)		0.988 (0.001)	0.979 (0.000)	0.963 (0.001)	0.950 (0.000)
Acc. (FSNL)	5	0.953 (0.001)	0.948 (0.001)	0.937 (0.001)	0.930 (0.000)
Precision	(0.0)	0.865 (0.004)	0.778 (0.007)	0.654 (0.004)	0.593 (0.004)
Recall		0.895 (0.010)	0.827 (0.011)	0.699 (0.009)	0.689 (0.008)
F1-score		0.880 (0.005)	0.802 (0.003)	0.676 (0.006)	0.637 (0.005)
Acc. (FSL)		0.992 (0.001)	0.985 (0.000)	0.972 (0.001)	0.962 (0.000)
Acc. (FSNL)	5	0.967 (0.002)	0.963 (0.001)	0.959 (0.001)	0.954 (0.000)
Precision	(1.0)	0.903 (0.005)	0.841 (0.008)	0.787 (0.004)	0.761 (0.004)
Recall		0.930 (0.008)	0.882 (0.006)	0.876 (0.003)	0.908 (0.005)
F1-score		0.916 (0.005)	0.861 (0.004)	0.830 (0.003)	0.828 (0.003)
Acc. (FSL)		0.971 (0.003)	0.947 (0.001)	0.911 (0.002)	0.883 (0.001)
Acc. (FSNL)	1	0.844 (0.004)	0.834 (0.002)	0.821 (0.002)	0.806 (0.002)
Precision	(1.0)	0.533 (0.008)	0.383 (0.003)	0.263 (0.004)	0.207 (0.002)
Recall		0.973 (0.006)	0.949 (0.009)	0.951 (0.007)	0.952 (0.006)
F1-score		0.689 (0.007)	0.546 (0.004)	0.412 (0.005)	0.340 (0.003)

5.3 Episodic Training on miniImageNet

Results of ResNet18 models, trained with varying values of α on the miniImageNet dataset, are shown in Appendix D. Notably, an empirical observation suggests that higher values of α adversely impact performance. This degradation is attributed to the inherent complexity of the miniImageNet dataset in comparison to Omniglot. The optimal model, trained with $\alpha = 0.1$, is illustrated in Appendix D and quantitatively presented in Tab. 3. Since there are only 20 classes in the miniImageNet test dataset the maximum of ways are 19 with one novel class. The FSL accuracy, specifically at 5-way 1-shot (0.614) and 5-shot (0.751), outperforms state-of-the-art ResNet18 models trained with Prototypical Networks [43, 33]. The standard evaluation in FSL research is 5-way classification with 5-shot and 1-shot scenarios, therefore 10-way to 19-way is not available in the state-of-the-art publications. While FSNL performance experiences a decline with an increase of K-way, it is noteworthy that the accuracy difference between FSL and FSNL at 19-way is merely 0.025. This observation suggests that FSL

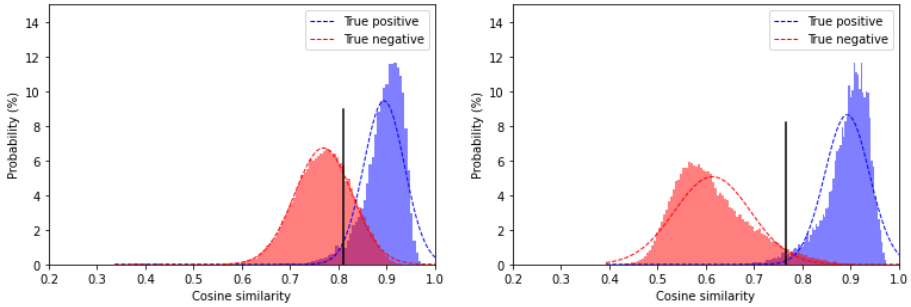
also encounters challenges when coping with a higher number of classes in the support set. Since the miniImageNet test dataset contains 20 classes the maximum test is 19-way with one novel class.

Table 3. Shows the performance metrics for few-shot and few-shot-novelty classification with $\alpha = 0.1$ trained on the miniImageNet dataset with ResNet18. The reference accuracy achived by Ziko et al. [43] for the Prototypical Networks trained on ResNet18 is shown.

Metric	5-way 1-shot Avg (SD)	5-way 5-shot Avg (SD)	10-way 5-shot Avg (SD)	15-way 5-shot Avg (SD)	19-way 5-shot Avg (SD)
Acc. [43]	0.542	0.737	-	-	-
Acc. (FSL)	0.614 (0.006)	0.751 (0.004)	0.626 (0.002)	0.557 (0.001)	0.517 (0.001)
Acc. (FSNL)	0.532 (0.002)	0.676 (0.004)	0.580 (0.002)	0.526 (0.002)	0.492 (0.001)
Precision	0.400 (0.004)	0.749 (0.008)	0.569 (0.014)	0.325 (0.007)	0.175 (0.006)
Recall	0.680 (0.008)	0.516 (0.013)	0.302 (0.012)	0.141 (0.003)	0.072 (0.003)
F1-score	0.504 (0.003)	0.611 (0.011)	0.394 (0.011)	0.196 (0.004)	0.102 (0.004)

5.4 Pre-Trained Models with Episodic Fine-Tuning

In this section, we present the results obtained from fine-tuning different ResNet models on the CUB and EU Moths datasets, utilizing pre-trained weights from ImageNet.



(a) Pre-trained on ImageNet. **(b)** Fine-tuned on EU Moths dataset.

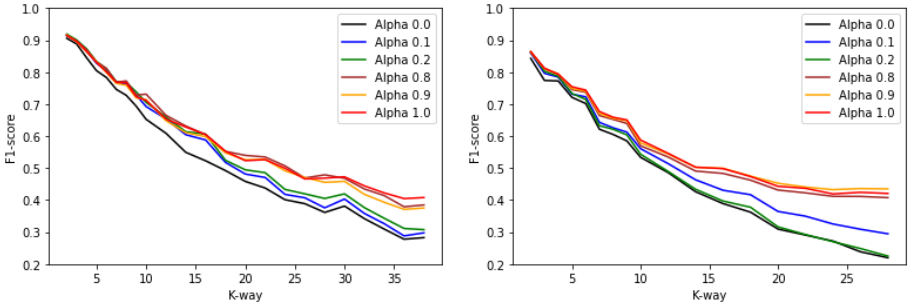
Fig. 6. Shows the distribution for known (TP) and outlier classes (TN) with pre-trained weights on ImageNet and fine-tuned model on EU Moths dataset with fine-tuning. The black vertical line shows the learned Bayes threshold.

Figure 6 illustrates the distribution of known and outlier classes for ResNet models. These models are fine-tuning with $\alpha = 0.5$ on the EU Moths dataset. Notably, the distributions exhibit a more distinct separation after the fine-tuning process, as depicted in Fig. 6b. Experiments did show that fine-tuning alone with $\alpha = 0$ do also contribute to a better separation. While the outlier class appears more akin to a beta distribution post fine-tuning, the enhanced separation is evident. The fine-tuned model achieves a notable improvement in few-shot accuracy, reaching 0.978 compared to the baseline accuracy of 0.938 with pre-trained ImageNet weights. Furthermore, the few-shot-novelty accuracy

experiences a substantial boost from 0.764 to 0.928. This improvement is attributed to an increase in novel class precision, rising from 0.394 to 0.844 and recall, ascending from 0.249 to 0.805.

Figure 7 reveals a decline in F1-score as the K-ways increase, suggesting challenges in classifying queries with an elevated number of classes in the support set. Interestingly, the F1-score peaks for different values of α, especially when the number of classes in the support set rises, with the highest performance observed at $\alpha = 1.0$. Performance metrics for fine-tuned ResNet18 models on the EU Moths and CUB datasets, specifically with the best-performing models at $\alpha = 1.0$, are detailed in Tab. 4 and Tab. 5, respectively. These tables (5-shot) illuminate a diminishing trend in F1-score as the number of support set classes increases, primarily driven by a notable decline in recall. Of particular interest is the observation that, for models fine-tuned on the CUB dataset (as detailed in Tab. 5), the FSNL accuracy at 0.751 is close to the FSL accuracy at 0.771 for the 29-way classification scenario. This indicates that FSL do also have difficulties with many classes. Performance results for 1-shot is detailed in Appendix E and has as expected a lower performance than 5-shot especially the precision are low with many classes in the support set.

These outcomes underscore the efficacy of our proposed combined scatter and cross-entropy loss function in the fine-tuning process, demonstrating its capability to enhance model performance in FSL scenarios with novelty detection.



(a) F1-score for novel class on EU Moths.

(b) F1-score for novel class on CUB dataset.

Fig. 7. The plots show F1-score for the novelty class with different numbers of classes (K-way) in the support set with ResNet18 models fine-tuned on the EU Moths and CUB dataset.

Table 4. Shows the performance metrics for few-shot and few-shot-novelty classification with $\alpha = 1.0$ and 5-shot trained on the EU Moths dataset with ResNet18.

Metric	5-way Avg (SD)	10-way Avg (SD)	20-way Avg (SD)	30-way Avg (SD)	40-way Avg (SD)
Acc. (FSL)	0.977 (0.002)	0.956 (0.002)	0.926 (0.001)	0.903 (0.001)	0.886 (0.001)
Acc. (FSNL)	0.929 (0.003)	0.915 (0.001)	0.896 (0.001)	0.882 (0.001)	0.867 (0.001)
Precision	0.888 (0.008)	0.805 (0.005)	0.653 (0.013)	0.573 (0.008)	0.486 (0.007)
Recall	0.766 (0.011)	0.617 (0.012)	0.441 (0.010)	0.384 (0.014)	0.315 (0.005)
F1-score	0.822 (0.007)	0.699 (0.008)	0.526 (0.011)	0.459 (0.012)	0.382 (0.006)

6 Conclusion

In this study, we introduce a novel method for classifying samples within the domain of few-shot learning with novelty detection. Our approach involves the identification of new classes not encompassed in the FSL support set through the use of a threshold, learned by the Bayes probabilistic decision function. The threshold is learned by the distribution of known and outlier samples evaluated using the cosine similarity measure on a validation dataset. Additionally, we propose the integration of a novel scatter loss function during episodic training to effectively segregate similarities between known and outlier samples. Evaluating our method on the Omniglot dataset, we achieve a noteworthy accuracy range of 0.954 to 0.967 across scenarios involving 5 to 30 classes in the support set, each with 5-shot learning. Through episodic fine-tuning of ResNet models with pre-trained weights from ImageNet, our method showcases few-shot-novelty learning accuracy rates of 0.929 on the EU Moths dataset and 0.877 on the CUB datasets under a 5-way 5-shot configuration, each with an additional 1-novelty class.

Our solution presents a significant stride in addressing the challenge of detecting novel samples, particularly in real-life applications where sparse or non-existent samples for each class prevail. The demonstrated accuracy and efficacy of our proposed method underscore its potential as a valuable tool in scenarios characterized by limited labeled data, offering promise for a broad spectrum of practical applications.

Table 5. Shows the performance metrics for few-shot and few-shot-novelty classification with $\alpha = 1.0$ and 5-shot trained on the CUB dataset with ResNet18.

Metric	5-way Avg (SD)	10-way Avg (SD)	20-way Avg (SD)	29-way Avg (SD)
Acc. (FSL)	0.936 (0.004)	0.885 (0.001)	0.816 (0.001)	0.771 (0.001)
Acc. (FSNL)	0.877 (0.002)	0.843 (0.002)	0.791 (0.001)	0.751 (0.001)
Precision	0.858 (0.003)	0.781 (0.011)	0.654 (0.010)	0.606 (0.009)
Recall	0.668 (0.014)	0.487 (0.010)	0.343 (0.010)	0.327 (0.005)
F1-score	0.751 (0.009)	0.600 (0.011)	0.450 (0.010)	0.425 (0.006)

Acknowledgement. During the preparation of this work, the first author utilized ChatGPT [25] to enhance the clarity and formulation for parts of the written text. After using this tool, the authors reviewed and edited the content as needed and takes full responsibility for the content of the publication.

References

1. Amarbayasgalan, T., Jargalsaikhan, B., Ryu, K.H.: Unsupervised novelty detection using deep autoencoders with density based clustering. Applied Sciences (Switzerland) **8**(9) (2018). https://doi.org/10.3390/app8091468
2. Bjerge, K., Geissmann, Q., Alison, J., Mann, H.M., Høye, T.T., Dyrmann, M., Karstoft, H.: Hierarchical classification of insects with multitask learning and anomaly detection. Ecological Informatics **77**, 102278 (2023). https://doi.org/https://doi.org/10.1016/j.ecoinf.2023.102278

3. Bodesheim, P., Freytag, A., Rodner, E., Kemmler, M., Denzler, J.: Kernel null space methods for novelty detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2013). https://doi.org/10.1109/CVPR.2013.433
4. Böhlke, J., Korsch, D., Bodesheim, P., Denzler, J.: Exploiting Web Images for Moth Species Classification. In: Lecture Notes in Informatics (LNI), Proceedings - Series of the Gesellschaft fur Informatik (GI). vol. P-314 (2021)
5. Boudiaf, M., Masud, Z.I., Rony, J., Dolz, J., Piantanida, P., Ayed, I.B.: Transductive information maximization for few-shot learning. In: Advances in Neural Information Processing Systems. vol. 2020-December (2020)
6. Carreño, A., Inza, I., Lozano, J.A.: Analyzing rare event, anomaly, novelty and outlier detection terms under the supervised classification framework. Artificial Intelligence Review **53**(5) (2020). https://doi.org/10.1007/s10462-019-09771-y
7. Chen, W.Y., Wang, Y.C.F., Liu, Y.C., Kira, Z., Huang, J.B.: A closer look at few-shot classification. In: 7th International Conference on Learning Representations, ICLR 2019 (20 19)
8. Cui, P., Wang, J.: Out-of-Distribution (OOD) Detection Based on Deep Learning: A Review. Electronics (Switzerland) **11**(21) (2022). https://doi.org/10.3390/electronics11213500
9. Dasgupta, S., Sheehan, T.C., Stevens, C.F., Navlakha, S.: A neural data structure for novelty detection. Proceedings of the National Academy of Sciences of the United States of America **115**(51) (2018). https://doi.org/10.1073/pnas.1814448115
10. Dhillon, G.S., Chaudhari, P., Ravichandran, A., Soatto, S.: A baseline for few-shot image classification. In: 8th International Conference on Learning Representations, ICLR 2020 (2020)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. vol. 2016-Decem (2016). https://doi.org/10.1109/CVPR.2016.90
12. Hermann, M., Umlauf, G., Goldlücke, B., Franz, M.O.: Fast and Efficient Image Novelty Detection Based on Mean-Shifts. Sensors **22**(19) (2022). https://doi.org/10.3390/s22197674
13. Hospedales, T., Antoniou, A., Micaelli, P., Storkey, A.: Meta-Learning in Neural Networks: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(9) (2022). https://doi.org/10.1109/TPAMI.2021.3079209
14. Hou, R., Chang, H., Ma, B., Shan, S., Chen, X.: Cross attention network for few-shot classification. In: Advances in Neural Information Processing Systems. vol. 32 (2019)
15. Huang, X., Choi, S.H.: SAPENet: Self-Attention based Prototype Enhancement Network for Few-shot Learning. Pattern Recognition **135** (2023). https://doi.org/10.1016/j.patcog.2022.109170
16. Joachims, T.: Transductive inference for text classification using support vector machines. In: Proceedings of the 20th international conference on machine learning (2000)
17. Kim, J., Im, S., Cho, S.: ProFeat: Unsupervised image clustering via progressive feature refinement. Pattern Recognition Letters **164** (2022). https://doi.org/10.1016/j.patrec.2022.10.029
18. Kim, J., Kim, T., Kim, S., Yoo, C.D.: Edge-labeling graph neural network for few-shot learning. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. vol. 2019-June (2019). https://doi.org/10.1109/CVPR.2019.00010
19. Lake, B.M., Salakhutdinov, R., Tenenbaum, J.B.: Human-level concept learning through probabilistic program induction. Science **350**(6266) (2015). https://doi.org/10.1126/science.aab3050
20. Li, D., Zhang, J., Yang, Y., Liu, C., Song, Y.Z., Hospedales, T.: Episodic training for domain generalization. In: Proceedings of the IEEE International Conference on Computer Vision. vol. 2019-October (2019). https://doi.org/10.1109/ICCV.2019.00153

21. Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., van der Laak, J.A., van Ginneken, B., Sánchez, C.I.: A survey on deep learning in medical image analysis. Medical Image Analysis **42** (2017). https://doi.org/10.1016/j.media.2017.07.005
22. Liu, J., Song, L., Qin, Y.: Prototype Rectification for Few-Shot Learning. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **12346 LNCS**, 741–756 (2020). https://doi.org/10.1007/978-3-030-58452-8-43
23. Liu, Y., Lee, J., Park, M., Kim, S., Yang, E., Hwang, S.J., Yang, Y.: Learning to propagate labels: Transductive propagation network for few-shot learning. In: 7th International Conference on Learning Representations, ICLR 2019 (2019)
24. Markou, M., Singh, S.: Novelty detection: A review - Part 1: Statistical approaches. Signal Processing **83**(12) (2003). https://doi.org/10.1016/j.sigpro.2003.07.018
25. OpenAI: ChatGPT 3.5 (2023), https://chat.openai.com/
26. Pang, G., Cao, L., Aggarwal, C.: Deep Learning for Anomaly Detection: Challenges, Methods, and Opportunities. In: WSDM 2021 - Proceedings of the 14th ACM International Conference on Web Search and Data Mining (2021). https://doi.org/10.1145/3437963.3441659
27. Pang, G., Shen, C., Cao, L., Hengel, A.V.D.: Deep Learning for Anomaly Detection: A Review. ACM Computing Surveys **54**(2) (2021). https://doi.org/10.1145/3439950
28. Qiao, L., Shi, Y., Li, J., Tian, Y., Huang, T., Wang, Y.: Transductive episodic-wise adaptive metric for few-shot learning. In: Proceedings of the IEEE International Conference on Computer Vision. vol. 2019-October (2019). https://doi.org/10.1109/ICCV.2019.00370
29. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings (2017)
30. Ruff, L., Kauffmann, J.R., Vandermeulen, R.A., Montavon, G., Samek, W., Kloft, M., Dietterich, T.G., Muller, K.R.: A Unifying Review of Deep and Shallow Anomaly Detection. Proceedings of the IEEE **109**(5) (2021). https://doi.org/10.1109/JPROC.2021.3052449
31. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision **115**(3) (2015). https://doi.org/10.1007/s11263-015-0816-y
32. Sabokrou, M., Fathy, M., Zhao, G., Adeli, E.: Deep End-to-End One-Class Classifier. IEEE Transactions on Neural Networks and Learning Systems **32**(2) (2021). https://doi.org/10.1109/TNNLS.2020.2979049
33. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems. vol. 2017-December (2017)
34. Song, Y., Wang, T., Cai, P., Mondal, S.K., Sahoo, J.P.: A Comprehensive Survey of Few-shot Learning: Evolution, Applications, Challenges, and Opportunities. ACM Computing Surveys **55**(13s) (2023). https://doi.org/10.1145/3582688
35. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to Compare: Relation Network for Few-Shot Learning. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2018). https://doi.org/10.1109/CVPR.2018.00131
36. Villa-Pérez, M.E., Álvarez-Carmona, M., Loyola-González, O., Medina-Pérez, M.A., Velazco-Rossell, J.C., Choo, K.K.R.: Semi-supervised anomaly detection algorithms: A comparative summary and future research directions. Knowledge-Based Systems **218** (2021). https://doi.org/10.1016/j.knosys.2021.106878
37. Villon, S., Iovan, C., Mangeas, M., Claverie, T., Mouillot, D., Villéger, S., Vigliola, L.: Automatic underwater fish species classification with limited data using few-shot learning. Ecological Informatics **63** (2021). https://doi.org/10.1016/j.ecoinf.2021.101320

38. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D.: Matching networks for one shot learning. In: Advances in Neural Information Processing Systems (2016)
39. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: Caltech-UCSD Birds-200-2011 (CUB-200-2011). California Institute of Technology **CNS-TR-2011-001** (2011)
40. Wang, Y., Chao, W.L., Weinberger, K.Q., van der Maaten, L.: SimpleShot: Revisiting Nearest-Neighbor Classification for Few-Shot Learning. arXiv (2019), http://arxiv.org/abs/1911.04623
41. Wang, Y., Yao, Q., Kwok, J.T., Ni, L.M.: Generalizing from a Few Examples: A Survey on Few-shot Learning. ACM Computing Surveys **53**(3) (2020). https://doi.org/10.1145/3386252
42. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Advances in Neural Information Processing Systems (2004)
43. Ziko, I.M., Dolz, J., Granger, E., Ayed, I.B.: Laplacian regularized few-shot learning. In: 37th International Conference on Machine Learning, ICML 2020. vol. PartF168147-15 (2020)

Appendix A Proof of Decision Threshold Function

We have to solve the Bayes decision function in Eq. (9) to prove Eq. (10)

$$\begin{aligned} &\frac{1}{\sqrt{2\pi\sigma_o}} \exp^{-\frac{(s_d-\mu_o)^2}{2\sigma_o^2}} P(\omega_o) \\ &= \frac{1}{\sqrt{2\pi\sigma_k}} \exp^{-\frac{(s_d-\mu_k)^2}{2\sigma_k^2}} P(\omega_k) \end{aligned}$$

$$\Updownarrow$$

$$\exp^{\frac{(s_d-\mu_o)^2}{2\sigma_o^2}-\frac{(s_d-\mu_k)^2}{2\sigma_k^2}} = \frac{\sqrt{2\pi\sigma_k}}{\sqrt{2\pi\sigma_o}} \frac{P(\omega_o)}{P(\omega_k)}$$

inserting $P(\omega_k)$ in Eq. (6) and $P(\omega_o)$ in Eq. (7) we get

$$\frac{(s_d-\mu_o)^2}{2\sigma_o^2} - \frac{(s_d-\mu_k)^2}{2\sigma_k^2} = ln(\frac{\sqrt{2\pi\sigma_k}}{\sqrt{2\pi\sigma_o}} \frac{M}{K})$$

$$\Updownarrow$$

$$\frac{\sigma_k^2(s_d-\mu_o)^2 - \sigma_o^2(s_d-\mu_k)^2}{2\sigma_o^2\sigma_k^2} = ln(\frac{\sqrt{\sigma_k}}{\sqrt{\sigma_o}} \frac{M}{K})$$

$$\Updownarrow$$

$$\begin{aligned} 2\sigma_o^2\sigma_k^2 ln(\frac{\sqrt{\sigma_k}}{\sqrt{\sigma_o}} \frac{M}{K}) &= \sigma_k^2(s_d-\mu_o)^2 - \sigma_o^2(s_d-\mu_k)^2 \\ &= \sigma_k^2(s_d^2 - 2\mu_o s_d + \mu_o^2) - \sigma_o^2(s_d^2 - 2\mu_k s_d + \mu_k^2) \\ = (\sigma_k^2-\sigma_o^2)s_d^2 - 2(\sigma_k^2\mu_o - \sigma_o^2\mu_k)s_d &+ \sigma_k^2\mu_o^2 - \sigma_o^2\mu_k^2 \end{aligned}$$

$$\Updownarrow$$

$$\begin{aligned} (\sigma_k^2-\sigma_o^2)s_d^2 - 2(\mu_o\sigma_k^2 - \mu_k\sigma_o^2)s_d \\ +\sigma_k^2\mu_o^2 - \sigma_o^2\mu_k^2 - 2\sigma_o^2\sigma_k^2 ln(\frac{M\sqrt{\sigma_k}}{K\sqrt{\sigma_o}}) = 0 \end{aligned}$$

■

Appendix B Parameters for Training

Table 5 shows the arguments used for training and validation of models with the four different datasets. The task of support sets with 5-way and 5-shot is used during episodic training and validation after each epoch. The number of queries for training on the EU moths dataset is limited to 6 since only 11 sample images are present for each class. The learning rate is smaller for fine-tuning the pre-trained models on ImageNet.

Table 5. Shows the arguments used for model training and validations with support sets of 5-way and 5-shot. α values are varied in steps of 0.1 for each version of trained model.

Dataset	Model	Pre-tained	Epochs	Milestone 1	Milestone 2	Tasks train	Tasks val.	Queries	Learn rate	α
Omniglot	ResNet12	No	350	120	250	200	100	10	0.05	0.0 - 1.0
miniImageNet	ResNet18	No	250	120	190	500	100	10	0.01	0.0 - 1.0
EU Moths	ResNet18	ImageNet	250	120	190	500	100	6	0.001	0.0 - 1.0
CUB	ResNet18	ImageNet	250	120	190	500	100	10	0.001	0.0 - 1.0

Appendix C Threshold Experiments on EU Moths

Figure 8 shows the performance on the EU Moths dataset with varying number of novel classes in the query of the test dataset. Figure 9 shows the threshold sensitivity experiment on the EU Moths dataset.

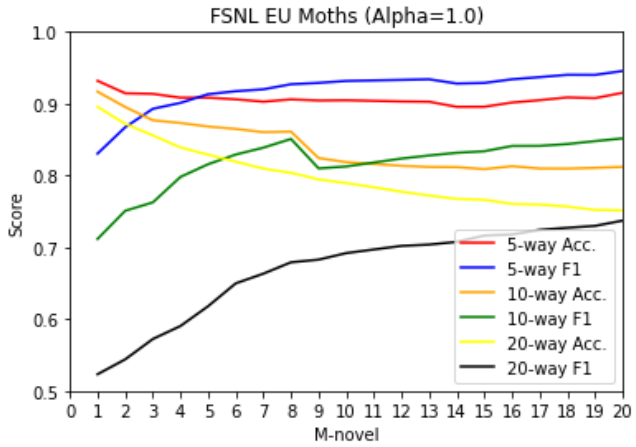
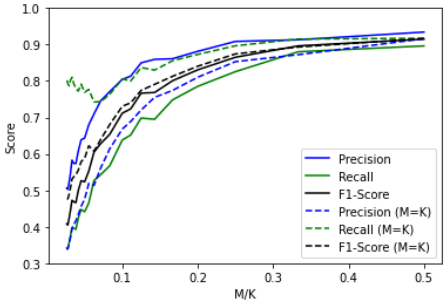


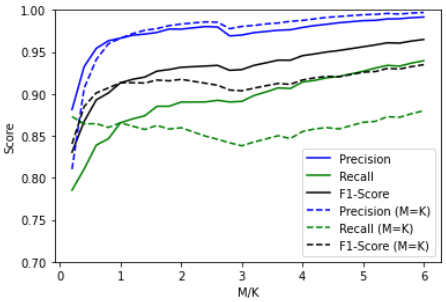
Fig. 8. Results of FSNL (trained with $\alpha = 1.0$) on the EU Moths dataset with varying numbers of novel classes (M-novel), fixed with either 5-way, 10-way, or 20-way, and with 5-shot images in the support set. The plot shows the few-shot-novelty accuracy on test episodes and F1-score for the novelty class.

Appendix D Training with miniImageNet

Results of ResNet18 models, trained with varying values of α on the miniImageNet dataset, are shown in Fig. 10.

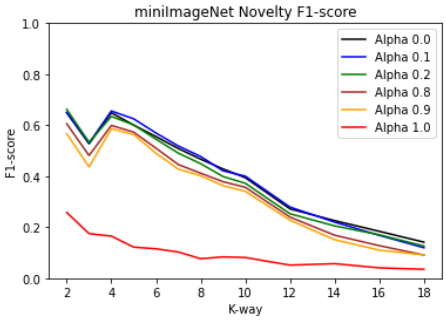


(a) Threshold sensitivity for one ($M = 1$) novel class and K-way.

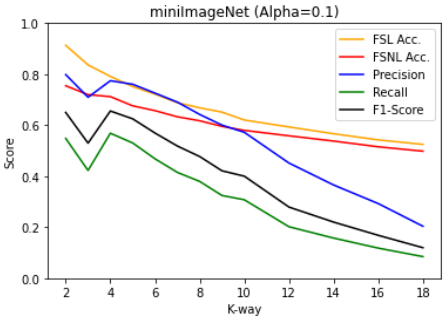


(b) Threshold sensitivity for 5-way and M-novel classes.

Fig. 9. Shows the precision, recall and F1-scores for different relative probabilities of M/K on the EU Moths dataset. The dotted lines shows when threshold is learned assuming equal prior probabilities ($M = K$).



(a) F1-score for different numbers of support classes and different α values.



(b) Different performance metrics depending on the number of support classes

Fig. 10. The top plot shows F1-score for the novelty class with different numbers of classes (K-way) on the miniImageNet dataset. The bottom plot shows the FSL and FSNL accuracy on test episodes with precision, recall and F1-score for the novelty class with $\alpha = 0.1$.

Table 6. Shows the performance metrics for few-shot and few-shot-novelty classification with $\alpha = 0.1$ and 1-shot trained on the miniImageNet dataset with ResNet18.

Metric	5-way Avg (SD)	10-way Avg (SD)	15-way Avg (SD)	19-way Avg (SD)
Acc. (FSL)	0.614 (0.006)	0.466 (0.003)	0.395 (0.002)	0.357 (0.002)
Acc. (FSNL)	0.532 (0.002)	0.431 (0.002)	0.369 (0.001)	0.335 (0.002)
Precision	0.400 (0.004)	0.267 (0.006)	0.171 (0.004)	0.129 (0.003)
Recall	0.680 (0.008)	0.501 (0.010)	0.344 (0.010)	0.270 (0.007)
F1-score	0.504 (0.003)	0.349 (0.007)	0.228 (0.006)	0.174 (0.004)

Appendix E Test with 1-shot Support Set

Table 6 shows the performance with $\alpha = 0.1$ and 1-shot learned on the miniImageNet test dataset with 20 classes. Table 7 shows the performance with $\alpha = 1.0$ and 1-shot

learned on the EU Moths test dataset with 50 classes. Table 8 shows the performance with $\alpha = 1.0$ and 1-shot learned on the CUB test dataset with 30 classes. Acc. (FSL) is the accuracy for few-shot learning and Acc. (FSNL) is the accuracy for few-shot-novelty learning with precision, recall and F1-score for the novelty class.

Table 7. Shows the performance metrics for few-shot and few-shot-novelty classification with $\alpha = 1.0$ and 1-shot trained on the EU Moths dataset with ResNet18.

Metric	5-way Avg (SD)	10-way Avg (SD)	20-way Avg (SD)	30-way Avg (SD)	40-way Avg (SD)
Acc. (FSL)	0.921 (0.004)	0.864 (0.004)	0.793 (0.003)	0.749 (0.002)	0.718 (0.001)
Acc. (FSNL)	0.769 (0.006)	0.739 (0.004)	0.702 (0.004)	0.673 (0.002)	0.651 (0.001)
Precision	0.443 (0.009)	0.294 (0.004)	0.177 (0.003)	0.130 (0.002)	0.102 (0.001)
Recall	0.947 (0.007)	0.878 (0.007)	0.822 (0.011)	0.796 (0.008)	0.759 (0.012)
F1-score	0.604 (0.009)	0.440 (0.005)	0.292 (0.005)	0.224 (0.003)	0.180 (0.001)

Table 8. Shows the performance metrics for few-shot and few-shot-novelty classification with $\alpha = 1.0$ and 1-shot trained on the CUB dataset with ResNet18.

Metric	5-way Avg (SD)	10-way Avg (SD)	20-way Avg (SD)	29-way Avg (SD)
Acc. (FSL)	0.845 (0.004)	0.744 (0.003)	0.629 (0.002)	0.562 (0.002)
Acc. (FSNL)	0.622 (0.005)	0.571 (0.003)	0.506 (0.001)	0.463 (0.001)
Precision	0.325 (0.003)	0.195 (0.002)	0.110 (0.001)	0.090 (0.001)
Recall	0.879 (0.007)	0.779 (0.006)	0.695 (0.004)	0.722 (0.003)
F1-score	0.474 (0.004)	0.312 (0.003)	0.190 (0.001)	0.159 (0.001)