NHL STENDEN
computer vision & data science

Few-Shot Class Incremental Learning for Insects Classification

NHL Stenden Lectoraat in Computer Vision & Data Science

Saman Kazeminia

Supervisors: Lucas Ramos

Abstract—Agricultural productivity is heavily impacted by pests such as aphids, which cause significant crop damage and act as vectors for plant diseases. Effective pest detection and management are crucial but challenging with traditional methods. This research introduces the Attention and Median Training-Free Prototype Calibration (AM-TEEN) model, a novel Few-Shot Class Incremental Learning (FSCIL) approach designed to improve the accuracy and robustness of pest detection systems. The model was evaluated using datasets including Aphids, Agricultural Pests from Kaggle, DLFautoinsects, and CIFAR-100. The experimental setup involved systematically varying one parameter at a time to comprehensively analyze its impact on model performance. AM-TEEN model significantly outperforms the base TEEN model. Specifically, in the first incremental session, accuracy improved from 87.50% to 97.92%, and in the last incremental session, accuracy increased from 67.08% to 74.17%. These improvements underscore the model's enhanced ability to handle incremental learning challenges and maintain high accuracy across diverse datasets. The enhanced model not only offers better accuracy and robustness but also demonstrates the potential for more efficient and scalable pest detection systems.

Index Terms—Computer Vision, Classification, Class Incremental Learning, Few-shot Learning, Aphids

1 Introduction

Protecting crops is crucial for sustaining agricultural productivity by minimizing the effects of weeds, pathogens, and pests like aphids. Aphids, in particular, pose a significant challenge in agriculture because of their role as disease carriers. These pests are a persistent issue for all types of growers, from small-scale gardens to large agricultural enterprises. Specifically, aphids like the green peach potato aphid not only damage plants but also transmit several harmful viruses, such as the potato y virus, which impacts economically important crops including potatoes [1][2][3][4].

Aphids significantly impair agricultural productivity not only by damaging crops but also by acting as vectors for various plant diseases. Consequently, effective detection and management of aphids are critical. Traditional methods, which typically involve meticulous visual inspection and counting of insects using approaches such as sticky plates, are labor-intensive, prone to errors, and may inadvertently harm beneficial insects [16]. In response to these challenges, there has been a shift towards developing more sophisticated techniques, including the application of artificial intelligence (AI), especially advancements in computer vision and deep learning (DL), which are now being used to detect and classify insects more efficiently and accurately, marking a significant improvement in how we tackle this pest problem [5].

With these limitations for manual counting and classification in mind, researchers are using computer vision and DL to enhance the process of managing these pests and improve the accuracy of insects image classification [5]. Although the newly designed DL algorithms have shown promising results, acquiring enough training data remains a challenge. The limited availability of images for training these models poses significant obstacles, particularly due to difficulties in data collection in the field. One innovative solution to this problem is few-shot learning (FSL), which utilizes minimally labeled data to enable the model to adapt insect species. Such adaptability enhances the model's utility in many real world scenarios related to managing a scarce amount of training data [6].

Another challenge in this context is the continuous emergence of new insect classes that might not have been on the training data. and the need to integrate these efficiently and effectively into the existing model. Few-shot class incremental learning (FSCIL) offers a promising solution to this problem [7][10]. This approach not only addresses the challenge of image classification in situations with scarce data but also allows for the progressive inclusion of new insect classes into the model without the need for repeated re-training of the entire system. Importantly, it manages to do this while maintaining or even improving the accuracy of previously trained classes as new data is introduced. This capability of FSCIL to adapt incrementally suggests that it may be one of the potential solutions for managing dynamic agricultural environments, where different types of insects are regularly encountered.

This paper aims to enhance an existing FSCIL (called Training Free Prototype Calibration - TEEN) approach by integrating an attention mechanism and optimizing the prototype generation method. Our objective is to develop a model for the classification of insects, such as aphids, using minimal data collected in the field and to incrementally incorporate new insect classes encountered after the initial training, in a resource-efficient manner. The subsequent sections will review related works, provide an in-depth explanation of the proposed solution, describe the methodologies, detail the experimental scenarios, and present the results and conclusions.

2 State of the art

Few-shot learning (FSL) is an approach in which a model learns to make predictions by training on a very small number of labeled examples. The work of [8] proposes a solution to recognizing insect pests using a FSL approach, and to handle the challenge of the high similarity between insect variations at similar maturity stages. A subset of insect images from the IP102 dataset is used for representing 45 young and 97 adult classes of insects. Then a FSL prototypical network was

Saman Kazeminia is a Computer Vision & Data Science student at the NHL Stenden University of Applied Sciences, E-mail: saman.kazeminia@student.nhlstenden.nl.

Lucas Ramos is a researcher at the NHL Stenden Lectoraat in Computer Vision & Data Science, E-mail: lucas.ramos@nhlstenden.com.

employed to divide young and mature classes into seperate groups on this dataset in comparison with mini-imagenet dataset as a baseline. Regarding the Kullback-Leibler divergence metric, the best results yielded an accuracy of 86.33% for adults and 87.91% for early stages.

Another approach for counting and classifying in images would be the use of class incremental learning (CIL). The work proposed in [9] selects the Dynamically Expandable Representation (DER) as the baseline. They combine the intrinsic properties of sonar images of marine debris and sea bed objects with deep learning theories and optimize both the backbone and the CIL strategies of DER. The culmination of this optimization is the introduction of DER-Sonar, a CIL network tailored for sonar images. Evaluations on SonarImage20 shows that it can outperform competing CIL networks with an average accuracy of 96.30%, an improvement of 7.43% over the baseline.

The study presented in [11] focuses on memory and computational constraints in model training. Specifically, it highlights three key points: each class is represented by only a limited number of training samples, adding a new class requires a fixed amount of computational effort, and the model's memory usage increases linearly as more classes are added. They propose C-FSCIL, a model designed with a frozen meta-learned feature extractor, a trainable fixed-size fully connected layer, and a dynamically growing data structure that stores vectors based on the number of introduced classes. The model strikes a kind of balance between accuracy and the cost of computing. It utilizes hyperdimensional embedding to represent classes in a fixed vector space while minimizing overlap. The experiments were conducted on the CIFAR100, miniImageNet, and Omniglot datasets show that C-FSCIL can outperform some state of the art methods by achieving the 50.47% accuracy in the final session for CIFAR-100, 51.41% for miniImageNet, and 85.70% for the Omniglot dataset.

In their study [12], researchers introduced a distillation algorithm specifically tailored for FSCIL, utilizing semantic information such as word embeddings to aid the training process. These embeddings are not only cost-effective but also improve the distillation process. Additionally, they developed a technique using an attention mechanism to synchronize visual and semantic vectors across multiple embeddings to help mitigating the problem of catastrophic forgetting. Their experiments on the MiniImageNet, CUB200, and CIFAR100 datasets set new benchmarks, showed a 39.04% accuracy on MiniImageNet's final session, substantially higher than the next best result of 24.4% by TOPIC, showing a more than 14% lead. Similar results were seen on CIFAR100 and CUB200, where they recorded accuracies of 34.80% and 32.96%, respectively, again leading the field.

An alternative FSCIL approach was proposed in [13] named the forward compatible training (FACT) approach. They utilized virtual prototypes to squeeze the embeddings of existing classes while making room for future ones. Additionally, the system anticipated potential new classes and sets the stage for subsequent updates. These virtual prototypes served as placeholders within the embedding space, enhancing the model's capability to integrate future changes and strengthening the classifier during inference. Comparative results on the CUB200 dataset show that the model surpassed the CEC method, achieving improvements of 2.8%, 6.6%, and 6.4% in the accuracy of base classes, new classes, and their harmonic mean, respectively.

While existing FSCIL approaches typically involve adding new learnable components or using a frozen feature extractor to reduce issues like catastrophic forgetting and overfitting, these methods often incorrectly classify samples of new classes as belonging to base classes, resulting in poor performance for the new categories. Paper [14] discovered that although the feature extractor is initially trained only on base classes, it can still capture the semantic similarities between these and the untrained, new classes. Based on these insights, they developed a novel approach called training-free calibration (TEEN). This strategy improves the recognition of new classes by merging new prototypes (i.e., the mean features of a class) with weighted base prototypes. Tested against CIFAR100, CUB200, and miniImageNet, TEEN achieved better results in comparison to methods like CEC, TOPIC, and FACT.

Given the current state of the art in FSCIL and related methodologies, it is clear that while existing techniques like TEEN and FACT have set state-of-the-art benchmarks, there is still room for enhancement, especially in handling newly introduced insect classes more dynamically and efficiently. Current approaches often lack the incorporation of attention mechanisms, which have shown to produce better results in various machine learning tasks [19][20]. Our approach addresses this gap by integrating an attention mechanism just before the fully connected layer. This integration effectively amplifies important features and suppresses less relevant ones, leading to more precise classification. Additionally, we have advanced the prototype generation process by adopting other statistics for prototype calculation. These contributions aimed at not only maintain high accuracy across both established and newly introduced classes and improve the overall accuracy but also to reduce the computational overhead typically associated with traditional FSCIL approaches. As a result, our method is seeking a more scalable and robust solution in the diverse agricultural settings.

3 Materials and Methods

This section describes the development of a FSCIL approach for image classification with images taken in a farm environment and newly introduced classes over time. The model addresses the challenge of continuously learning new classes with minimal data for each new class and without training for new classes.

3.1 Dataset

The main dataset used in this study was created using an imaging setup located on a farm, which captures images of moving and flying insects. Full specifications of the setup are detailed in Appendix A.1. The dataset comprises 1,131 insect images of varying sizes, which have been resized to 224x224 pixels for consistency. These images are categorized into 10 distinct classes with the distribution showd in Table 1. The dataset is divided into training, validation, and testing sets, with a split ratio of 70%, 10%, and 20%, respectively. Figure 7 displays sample data classes from the aphids dataset.

In addition to the main dataset, two other datasets were utilized in this study. Agricultural Pests Dataset published on Kaggle by Gaurav Dutta, 2023, comprises 5,179 images across 12 insect classes, namely Ants, Bees, Beetles, Caterpillars, Earthworms, Earwigs, Grasshoppers, Moths, Slugs, Snails, Wasps, and Weevils. The images were collected from Flickr using an API and resized to have a maximum width or height of 300 pixels. This dataset is designed to aid researchers and practitioners in the development and evaluation of DL models for pest detection and classification in agricultural settings. The per-class distribution varies, providing a diverse range of images that cover various shapes, colors, and sizes. The Agricultural Pests from Kaggle dataset were cleaned and split as part of this study.

DLFautoinsects Dataset published by Chengjun Xie et al., 2018, contains 4,500 images spanning 40 pest insect classes collected from crop fields. The classes include Dolycoris baccarum that affects various crops including berries and

Fig. 1: Sample images from the datasets: (first row) Aphids, (second row) Agricultural Pests, (third row) DLFautoinsects

Table 1: Aphids class distribution

Class	Number of Instances
Thrips	74
Melanogaster	124
Wasp	40
Aphid winged-outdoors	263
Spider	40
Aphid wingless	219
Crane Fly	123
Reflection	115
Water Drop	109
Moth	24
Total	**1131**

legumes, Lycorma delicatula an invasive species damaging a wide range of plants and trees, Eurydema dominulus which affects cruciferous plants, and many more. Each class has a varying number of images, providing a comprehensive dataset for evaluating the performance of DL models on diverse insect species. The images were resized to maintain consistency. These datasets sample images and data distributions are depicted in the appendix B3.

These datasets were used alongside the main dataset to test the robustness of the AM-TEEN model. The combination of Aphids and Agricultrual Pests and the combined Aphids, Agricultural Pests, and DLFautoinsects datasets enabled the assessment of the model's ability to generalize across different insect species and environmental conditions. The images from all datasets were preprocessed to maintain uniformity in size and format, ensuring compatibility with the model's input requirements.

3.2 Approach

Our approach builds upon the TEEN model introduced in [14], a FSCIL approach. The TEEN model trains the feature extractor on base classes. Base classes are the initial set of classes that the model is trained on before it begins to learn new classes incrementally. These classes form the foundation of the model's knowledge and are used to initialize the model's parameters. The TEEN model then employs semantic similarities to adapt new, untrained classes into the model in incremental sessions. The incremental sessions process involves loading the previously trained model, updating the fully connected layer with new class prototypes, and applying soft calibration. The model is then evaluated on the test dataset, and the results are logged and compared to previous sessions. The overview of this process is illustrated in the Fig. 4.

3.2.1 Feature Extraction

The feature extractor is built upon a pre-trained ResNet-18, a convolutional neural network widely used in image classification. This network uses residual blocks to enable the

training of deeper networks by effectively handling the vanishing gradient problem. The architecture consists of the following major parts:

The initial convolution and pooling, that involves a convolutional layer with a 7x7 kernel and stride 2, followed by batch normalization, a ReLU activation, and a max pooling layer. This setup initially reduces the spatial dimensions of the image (height and width) while increasing the depth (number of channels).

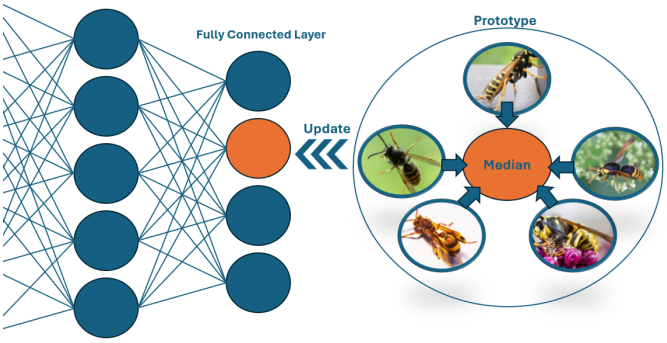


Fig. 2: Updating the FC layer

A series of residual blocks (Layers 1-4), which are sequential residual blocks that further process the data. These blocks can adjust their behavior using dilated convolutions instead of standard convolutions if specified, helping the network maintain a larger effective receptive field without significantly reducing the spatial dimensions. After these features are extracted, the next step involves updating the fully connected layer.

3.2.2 Updating Fully Connected Layer Weights

The FC layer weights are updated using calculated class prototypes. These prototypes are derived from the mean of the embeddings of data examples belonging to the new class. The process involves encoding the data examples to obtain their embeddings, computing the mean of these embeddings for each new class, and assigning these prototypes to the corresponding class indices in the FC layer. If no data initialization strategy is used, the weights are initialized randomly with a uniform distribution. Otherwise, the prototypes are computed directly from the embeddings of the new class examples. Fig. 2 is a simple illustration of this step. However, directly using these prototypes can lead to a misalignment with the base classes, which necessitates the next crucial step of soft calibration to ensure the new classes integrate seamlessly without degrading the model's performance.

3.2.3 Soft Calibration

To prevent catastrophic forgetting of previously learned classes, a technique known as soft calibration is employed. This method adjusts the prototypes of the new classes based on their similarity to the prototypes of the base classes. The adjustment involves computing a weighted sum of the original prototype and a calibration term derived from the base prototypes, scaled by a hyperparameter. This ensures the new prototypes are aligned with the previously learned ones, maintaining performance across both new and old classes. The Fig. 3 depicts a simple illustration of this step.

Then the soft calibration method performs a process to adjust the prototypes of the new classes based on their similarity to the prototypes of the base classes. Given a new class prototype c_n (where $B \leq n \leq B+C-1$), the calibrated new prototype $\bar{c}_n$ is

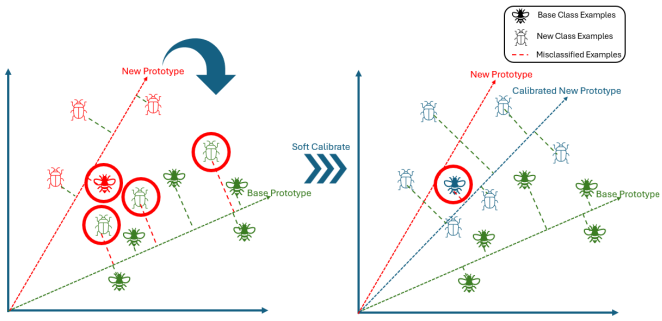


Fig. 3: The insect icons represent examples of base and new classes. The dotted lines between examples and prototypes illustrate the classification results: blue and green dotted lines indicate correctly classified samples, while red dotted lines and red circles denote misclassifications. The right figure shows the prediction changes after applying soft calibration.

computed as a weighted sum of the original prototype and a calibration term based on the base prototypes:

$$\bar{c}_n = \alpha c_n + (1-\alpha)\Delta c_n \tag{1}$$

where α is a hyperparameter controlling the strength of calibration. The calibration term Δc_n is derived from the base prototypes as:

$$\Delta c_n = \sum_{b=1}^{B-1} w_{b,n} c_b \tag{2}$$

with the weights $w_{b,n}$ calculated as the softmax of the cosine similarities between the new class prototype and all base class prototypes, scaled by a hyperparameter τ:

$$S_{b,n} = \frac{c_b \cdot c_n}{\|c_b\|\|c_n\|} \cdot \tau \tag{3}$$

Here, $S_{b,n}$ represents the scaled cosine similarity between the new class prototype c_n and the base class prototype c_b. The term $\frac{c_b \cdot c_n}{\|c_b\|\|c_n\|}$ computes the cosine similarity, which measures how similar the two vectors are in the feature space. This similarity is then scaled by τ, which controls the sharpness of the softmax distribution. The weights $w_{b,n}$ are then computed as:

$$w_{b,n} = \frac{e^{S_{b,n}}}{\sum_{i=0}^{B-1} e^{S_{i,n}}} \tag{4}$$

This formula normalizes the scaled similarities using the softmax function, ensuring that the weights sum to 1. The weights $w_{b,n}$ indicate the relative importance of each base class prototype in calibrating the new class prototype.

The soft calibration thus adjusts the new class prototype towards the direction of base class prototypes, accounting for the similarity between them, which helps to alleviate the bias in the prototype due to limited data in the incremental session. This calibration strategy is training-free, requiring no additional learning or updates to the model parameters, and aims to enhance the discriminability of the new class prototypes by incorporating the knowledge distilled from the base classes.

3.3 AM-TEEN

In this section, we outline the methodology used to develop our FSCIL approach, the Attention (An adapted variation of the channel wise attention mechanism proposed in [20]) with Median on Training-Free Prototype Calibration (AM-TEEN). We leverage a pre-trained ResNet-18 model as the backbone for

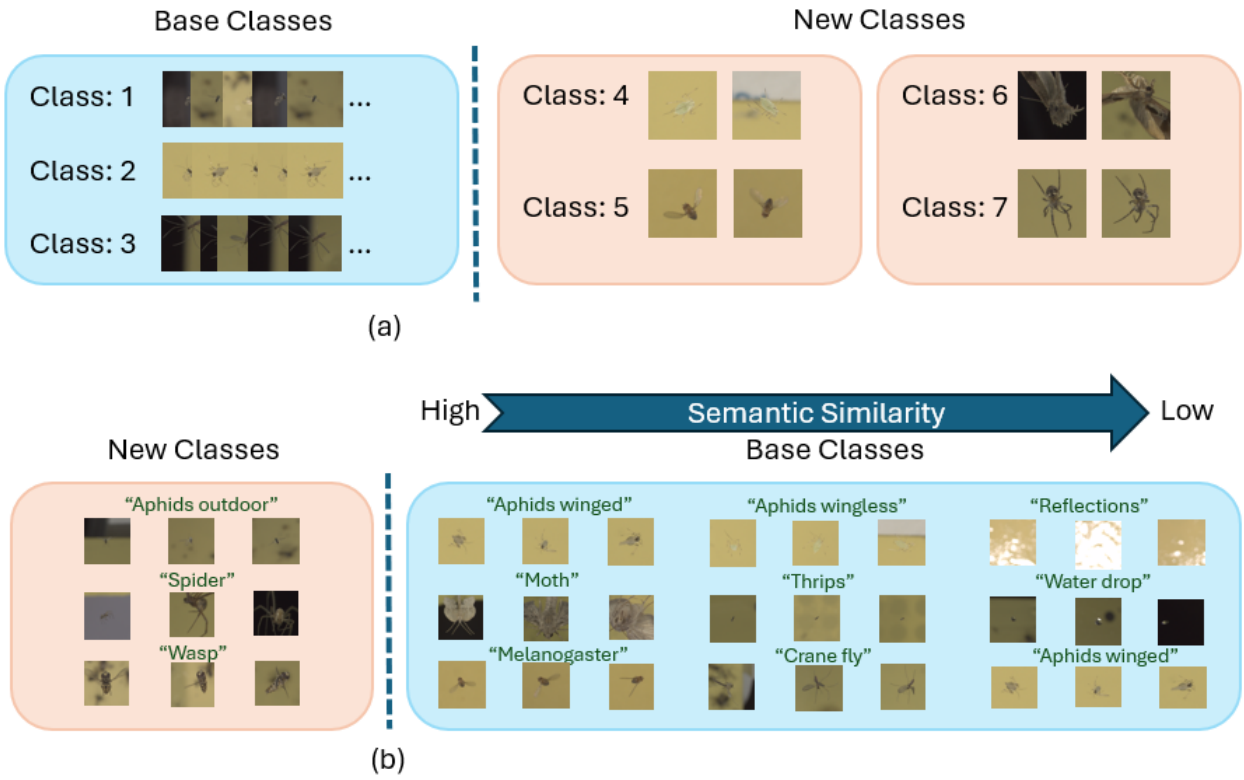


Fig. 4: (a) The base session includes sufficient examples of base classes for training. Subsequent incremental sessions contain only few-shot examples of novel classes. FSCIL aims to develop a unified classifier for all seen classes. (b) The most similar base classes are identified by computing the cosine similarity between base and novel prototypes. The results demonstrate that the feature extractor, trained solely on the base classes, effectively represents the semantic similarity between the base and novel classes.

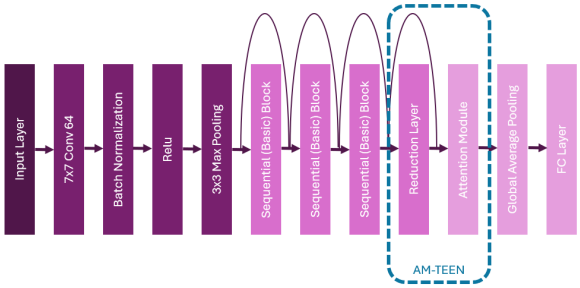


Fig. 5: AM-TEEN model architecture

feature extraction, just as the TEEN base model does. Initially, the network is trained with a predetermined number of base classes, forming the base knowledge of the model. Following this, the model is prepared to accept new classes incrementally. However, unlike the TEEN model which uses the mean of the embeddings, our approach updates the fully connected (FC) layer weights with the median of the embeddings (prototypes) of the new classes. The feature extractor remains unchanged, while the FC layer, responsible for classification, is updated with the weights of the new classes. In each increment, a defined number of new classes are introduced using a few-shot learning approach (e.g., 5-10 shots per class) and the prototype of the new class examples is computed and used to update the FC layer's weights. We also employ the soft calibration method to maintain the performance of previously learned classes while integrating new ones incrementally. Fig. 5 shows the architecture of the AM-TEEN model.

To address the challenge of focusing on the most relevant features and reducing computational complexity, we incorporate a reduction convolution layer and the channel attention mechanism. The reduction convolution, a 1x1 convolutional layer, reduces the depth of the feature maps from 512 to 64, lowering the computational load and simplifying the feature space before passing it to subsequent layers. This forms the basis for the attention mechanism, a novel contribution not present in the TEEN model. The attention module applies attention across the channels of the feature maps, starting with average pooling to squeeze spatial dimensions, resulting in a 1D vector per channel. This vector is processed through a small network to predict attention scores for each channel, which are used to rescale the original feature maps. This process allows the network to adaptively emphasize or de-emphasize certain channels, potentially focusing on more relevant features for the task. By computing attention scores across feature map channels, the model dynamically highlights the most relevant features, improving accuracy and adaptability. These steps boost the model's capacity by focusing on the most informative parts of the feature space, enhancing the overall performance and robustness of the AM-TEEN model in addressing the challenges of few-shot class incremental learning.

3.4 Training

Algorithm 1 Few-Shot Class-Incremental Learning Training

```
Initialize model with given architecture and arguments
if pre-trained model is available then
    Load pre-trained model weights
end if
for each session in total sessions do
    Load training and testing datasets
    Initialize session-specific transformations
    if session is base training session then
        for each epoch do
            Train model on base classes
            Evaluate on test dataset
            Save model if there is improvement
        end for
        if data initialization strategy is enabled then
            Replace FC with average embeddings
            Save initialized model
        end if
    else
        Set model to evaluation mode
        Update dataset transformations if required
        if soft prototype strategy is employed then
            Update FC with median of new class embeddings
            Perform soft calibration
            Save updated model after FC update and calibration
        end if
        Evaluate model on test dataset
        Update training log with current session's performance
    end if
    Save incremental session results
    Log results to TensorBoard
end for
Compute and save final results to file
Log final results to TensorBoard
Output total training time
```

The training process involves two primary steps: training on the base classes and incrementally training on new classes in subsequent sessions. This section elaborates on the detailed procedure for each step, highlighting how the model adapts to new data over time. The training algorithm is shown in the Algoritm 1.

3.4.1 Base Session

The model is trained on the base classes until convergence. During each epoch, the cross-entropy loss is computed using the ground truth and predicted labels of the training data. Additionally, the accuracy on the training data is evaluated. After training, the evaluation process involves validating the model on data unseen during training, and the test loss and accuracy are calculated for each epoch. If the evaluation accuracy surpasses the previous best accuracy, the model parameters are stored. The training progress and results, including loss, accuracy, and time per epoch, are logged.

3.4.2 Incremental Session

During incremental sessions, the fully connected layer weights are updated using the calculated class prototypes by the median of the embeddings of data examples belonging to the new class. These prototypes are assigned to the corresponding class indices in the fully connected layer. The soft calibration method adjusts the prototypes of the new classes based on their similarity to the prototypes of the base classes. This calibration strategy is training-free, requiring no additional learning or updates to the model parameters, and aims to enhance the discriminability of the new class prototypes by incorporating knowledge distilled from the base classes. Fig. 2 is a simple illustration of this step.

3.5 Evaluation Metrics:

When evaluating the performance of our model, we employ several metrics to ensure a comprehensive assessment, particularly given the imbalanced nature of some datasets. These metrics include average accuracy, seen accuracy, unseen accuracy, the harmonic mean of seen and unseen accuracy, and the F1 score.

Seen Accuracy measures the accuracy on previously encountered classes, providing insight into how well the model retains knowledge of classes it has already learned. Unseen Accuracy assesses the accuracy on newly introduced classes, reflecting the model's ability to generalize to new information. The harmonic mean of the seen and the unseen provides a balanced measure of performance across both known and new classes, and Average Accuracy is the overall accuracy across all classes seen so far.

In datasets with class imbalances, such as our aphids dataset, accuracy alone can be misleading. For example, in a highly imbalanced dataset, a model might achieve high accuracy by predominantly predicting the majority class, neglecting the minority class. This scenario highlights the need for more nuanced metrics like the F1 Score. The F1 Score is the harmonic mean of precision and recall, balancing the trade-off between these two metrics. Precision is the ratio of true positives to the sum of true positives and false positives, while recall is the ratio of true positives to the sum of true positives and false negatives. The F1 score provides a more reliable measure of a model's performance on imbalanced datasets by considering both false positives and false negatives.

We applied these metrics to our datasets, recognizing the significant imbalance present. In the aphids dataset, with a max-to-min imbalance ratio of 10.96, the need for reliable metrics like the F1 score is paramount. This imbalance is also observed in other datasets, albeit to a lesser extent, such as Agricultural Pests with a ratio of 2.96 and DLFautoinsects with a ratio of 4.76.

By utilizing the F1 score in addition to accuracy metrics, we ensure that our model's performance is robust and reliable across all classes, addressing the challenges posed by imbalanced data. This comprehensive evaluation approach confirms the effectiveness of our model in managing class imbalance, thereby enhancing its reliability and applicability in real-world scenarios.

4 Experiments & Results

We conducted experiments using the Aphids, Agricultural Pests from Kaggle, DLFautoinsects [17], and CIFAR-100 datasets with the TEEN, Attention Teen (A-TEEN), and AM-TEEN models. The experiments were conducted systematically, varying one parameter at a time while keeping others constant. This allowed for a comprehensive analysis of each parameter's influence on model outcomes.

4.1 Experimental setup

All experiments were conducted using 'Python 3.10.12', and for enhanced computational efficiency, we utilized the a 'NVIDIA A40-16C' GPU with 16 GB memory, and 'CUDA version 12.2', enabling parallel processing and rapid matrix calculations. Notably, access to computer hardware was facilitated through university servers, providing a setup-free Python environment for remote execution. The experiments were using a PC client featuring an i7-8550U CPU, a 1.99 GHz processor, and 12 GB of RAM. Specifically, on the Aphids

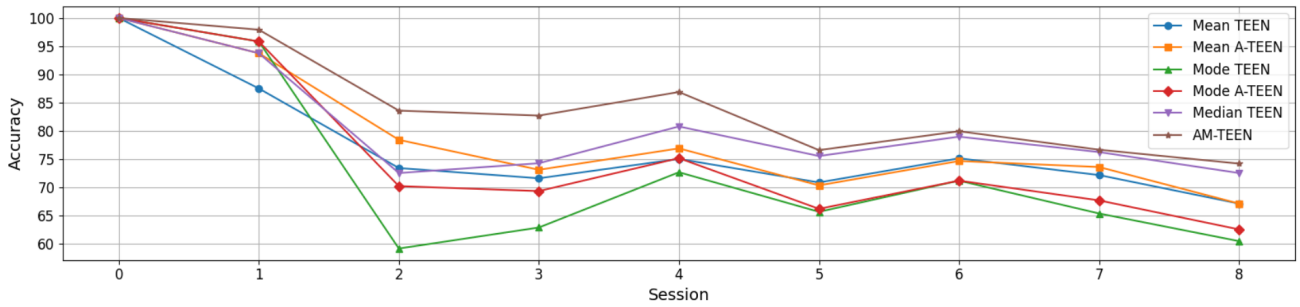


Fig. 6: Accuracy per session for the three different prototype generation methods (mode, mean and median) on Aphids dataset

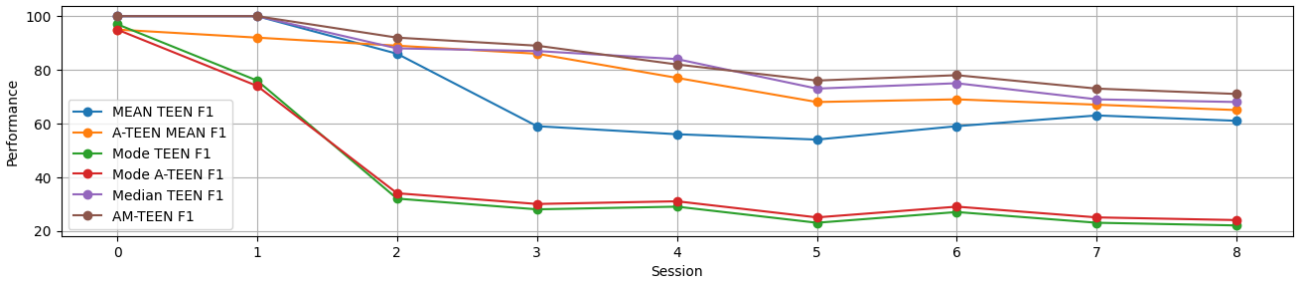


Fig. 7: F1 score per session for the three different prototype generation methods (mode, mean and median) on Aphids dataset

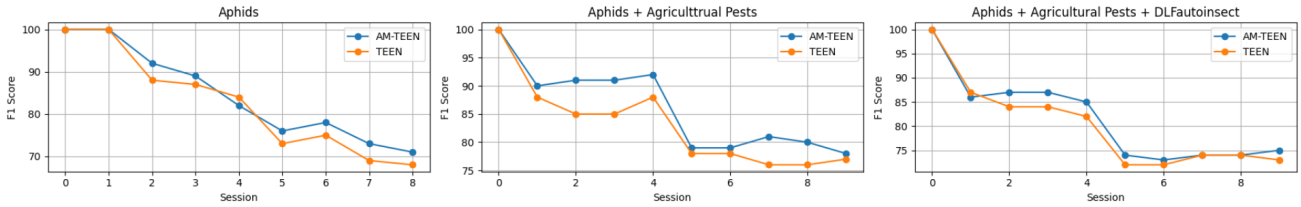


Fig. 8: F1 Score per session for TEEN and AM-TEEN models on Aphids, and two combinations of the Aphids and other datasets

dataset, the results were achieved with a batch size of 128 and a learning rate of 0.00003, using cross-entropy loss and SGD optimization. In terms of model complexity and training efficiency, the TEEN model consists of 11,221,568 parameters with an average training time of 20.74 minutes. In comparison, the AM-TEEN model, which includes the attention mechanism, has 1024 more parameters and an average training time of 21.76 minutes.

4.2 Experiment 1

In this first experiment, we compared different methods for generating prototypes from the embeddings of new class examples. We also assessed the influence of incorporating an attention mechanism in each model. Fig. 7 presents the results for base and incremental sessions on the Aphids dataset. The base classes are thrips and melanogaster and the incremental classes are wasp, aphid winged, spider, aphid wingless-outdoors, crane fly, reflection, water drop and moth.

Our evaluation focused on three prototype generation methods: mean, mode, and median. Across these methods, we measured the model's accuracy both with and without the attention mechanism.

The mean method shows strong initial performance in session 0, achieving 100% accuracy. However, accuracy drops considerably in subsequent sessions, falling to 67.08% by session 8. Incorporating attention yields (Mean A-TEEN, Mode A-TEEN, AM-TEEN) a noticeable improvement, with a smaller decline in accuracy over sessions. Attention mechanisms, despite adding a small number (around 1K here) of parameters and making the model to train slightly slower(5-7%), can provide significant improvements by dynamically highlighting important features in the data.

When using the mode method, the model also starts with 100% accuracy in session 0, but its performance drops more steeply than the mean method, reaching 60.42% by session 8. The addition of attention helps the mode method maintain higher accuracy across sessions, though the improvement is not as pronounced as with the mean method.

The median method demonstrates the best performance among the three, starting with 100% accuracy in session 0 and maintaining relatively high accuracy through subsequent sessions. By session 8, the model still achieves a 72.5% accuracy. With the attention mechanism, the median method achieves the highest sustained accuracy among all combinations, showing a smaller decline over time and maintaining accuracy above 74% through most sessions.

These results indicate that the median method provides better stability and accuracy compared to the mean and mode methods, especially when combined with attention. The use of attention consistently leads to improved performance, highlighting its importance regardless of the prototype generation method employed. The superior performance of the

Table 2: Accuracy per session for TEEN and AM-TEEN models on Aphids, CIFAR-100, and two combinations of the Aphids and other datasets

Dataset	Model	Session 0	Session 1	Session 2	Session 3	Session 4	Session 5	Session 6	Session 7	Session 8	Session 9
Aphids	TEEN	100	93.75	72.50	74.24	80.76	75.52	78.95	76.22	72.50	NA
	AM-TEEN	**100**	**97.92**	**83.57**	**82.69**	**86.87**	**76.56**	**79.91**	**76.64**	**74.17**	**NA**
Aphids + *	TEEN	**93.27**	83.30	80.36	72.09	66.44	65.69	64.67	60.52	59.78	61.71
	AM-TEEN	91.83	**84.72**	**81.92**	**75.97**	**70.92**	**70.28**	**68.73**	**66.27**	**62.83**	**64.32**
Aphids ++ **	TEEN	84.68	74.58	72.69	**63.93**	65.04	66.09	66.36	65.92	60.85	61.78
	AM-TEEN	**87.99**	**76.25**	**74.54**	63.73	**65.94**	**66.45**	**66.49**	**67.12**	**62.20**	**62.39**
CIFAR-100	TEEN	78.37	71.88	67.79	63.81	60.30	57.46	55.47	53.37	51.39	NA
	AM-TEEN	**78.83**	**72.35**	**68.53**	**64.67**	**61.60**	**58.68**	**56.56**	**54.54**	**52.27**	**NA**

* Denotes a combination with the previous datasets (Aphids + Agricultural Pests, 22 classes (4 Base + 18 Incremental) in total).
** Likewise (Aphids + Agricultural Pests + DLFautoinsects, 62 classes (8 Base + 54 Incremental) in total).

median method can be attributed to its robustness against outliers. By using the median of the embeddings to generate prototypes, this method effectively reduces the influence of anomalous data points, leading to more stable and representative prototypes. This characteristic is particularly beneficial in imbalanced datasets where outliers can disproportionately affect the mean, resulting in less reliable prototypes. Additionally, channel attention mechanism enhance the model's performance by dynamically emphasizing more relevant features and de-emphasizing less relevant ones. This selective focus allows the model to concentrate on the most informative parts of the feature space, improving both precision and recall. In imbalanced datasets, where certain classes may have fewer and more varied examples, the ability to highlight critical features ensures that the model can better distinguish between classes, thereby maintaining high accuracy and stability across incremental learning sessions.

The table 2 provides insights into two datasets, Aphids and CIFAR-100, as well as two combinations of the Aphids dataset with other datasets, comparing performance with and without attention across a base and incremental learning sessions. It is important to note that the combination datasets have one additional incremental session, making direct comparisons with the other datasets not straightforward due to differences in the base size, the number of prototypes in the incremental sessions, and the number of incremental sessions.

On the Aphids dataset (2 base classes and 1 incremental class per incremental session), the model achieves 100% accuracy in session 0 across all configurations. However, performance declines in subsequent sessions, notably without attention, with accuracy dropping to 72.50% by session 8. The incorporation of attention, however, boosts performance, with the model maintaining improved accuracy throughout all sessions and consistently outperforming the non-attention baseline, particularly in session 1 (97.92% with attention versus 93.75% without) and session 8 (74.17% versus 72.50%).

When evaluated on CIFAR-100, with more number of classes (100 classes (60 Base + 40 Incremental) in total), the benefits of incorporating attention are still pronounced. Although the initial accuracy starts slightly different than Aphids at 78.37% without attention and 78.83% with attention in session 0, the model demonstrates better resilience to incremental learning challenges when attention is used. Accuracy drops to 52.27% by session 8 with attention, compared to 51.39% without it. The difference is more noticeable in the earlier sessions, such as session 4 (61.60% with attention versus 60.30% without).

For the combination of Aphids and Agricultural Pests (Aphids +, 22 classes in total, 4 base classes and 2 incremental classes per incremental session), the TEEN model starts with 93.27% accuracy in session 0 and drops to 59.78% by session 8. In contrast, the AM-TEEN model starts with 91.83% and maintains a higher accuracy of 62.83% by session 8. This

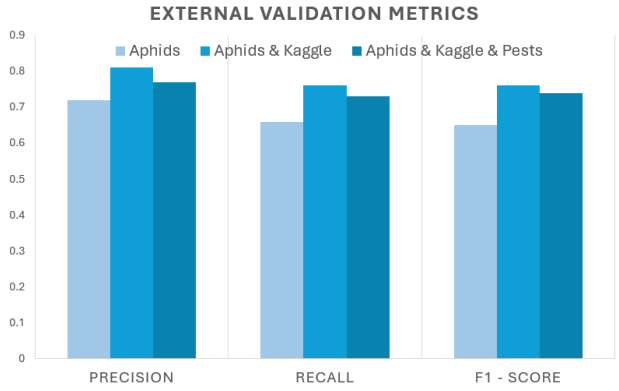


Fig. 9: External Validation Metrics for Different Datasets

dataset includes an additional incremental session (session 9), where the AM-TEEN model continues to show superior performance with an accuracy of 64.32% compared to 61.71% for the TEEN model.

For the combination of Aphids, Agricultural Pests, and DLFautoinsects (Aphids ++, 62 classes in total, 8 base classes and 6 incremental classes per incremental session), the TEEN model starts with 84.68% in session 0 and drops to 60.85% by session 8, while the AM-TEEN model starts with 87.99% and maintains a higher accuracy of 62.20% by session 8. This dataset also includes an additional incremental session, where the AM-TEEN model achieves an accuracy of 62.39% compared to 61.78% for the TEEN model.

While combining datasets and adding new classes does not always result in better performance, the Aphids, Agricultural Pests, and DLFautoinsects datasets demonstrates an improvement over the Aphids dataset but not over the Aphids, and Agricultural Pests dataset. This suggests that the model's performance is influenced by data similarities and quality. It becomes harder for the model to learn effectively when the incremental sessions introduce more variability or less relevant features. Therefore, while the addition of new classes can provide more diverse training data, it is beneficial up to a certain point beyond which the complexity and variability might negatively impact the model's performance.

The F1 score was particularly important in this evaluation, as it provided a more balanced view of the model's performance in the presence of class imbalances. For instance, in the aphids dataset, which had a max-to-min imbalance ratio of 10.96, accuracy alone could be misleading. By focusing on both precision and recall, the F1 score offered a more reliable measure, ensuring that the model's performance on minority classes was accurately reflected.

Table 3: External Validation per Class Metrics on Aphids and Agricultural Pests

Class	Precision	Recall	F1-Score
Thrips	0.80	1.00	0.89
Melanogaster	0.81	1.00	0.90
Wasp	0.67	0.50	0.57
Winged Aphid	0.86	0.44	0.59
Spider	1.00	0.50	0.67
Wingless Aphid	0.77	0.91	0.83
Crane Fly	0.50	0.46	0.48
Reflection	1.00	1.00	1.00
Water Drop	1.00	1.00	1.00
Moth	0.60	1.00	0.75
Weighted Average	**0.81**	**0.76**	**0.76**

Fig. 8 illustrates the comparison of the F1 scores between the TEEN and AM-TEEN models across different sessions and datasets. For the Aphids dataset, the AM-TEEN model consistently outperformed the TEEN model in terms of F1 score, demonstrating better handling of class imbalances. Similar trends were observed in the Aphids combinations with Agricultural Pests and DLFautoinsects datasets, where the AM-TEEN model maintained higher F1 scores across sessions, indicating its robustness in dealing with incremental learning scenarios.

These results indicate that combining the median method with attention mechanisms enhances the model's performance across different datasets. This combination consistently achieves higher accuracy and demonstrates better resilience to the challenges of incremental learning, outperforming baseline approaches and maintaining accuracy over time. The AM-TEEN model showcases its robustness and effectiveness across various datasets, proving its capability to handle incremental learning scenarios more efficiently than the TEEN model. Additionally, the use of the F1 score as an evaluation metric underscores the model's improved performance in managing class imbalances, further validating the improvements introduced by our proposed methods.

4.3 Experiment 2

The AM-TEEN model was tested to determine its robustness in classifying images from Aphids dataset when combined with one or two additional datasets. In this experiment the Aphids dataset included an external validation set of data, collected at a different location and time, providing a diverse and challenging test environment for the model. Additionally, we were focusing on assessing model performance through precision, recall, and F1-score metrics. The results are depicted in Fig. 9, which illustrates varying performance levels across the datasets.

For the Aphids dataset, the classification outcomes (see appendix B.2 for details) reveal a mixed performance with a weighted average F1-score of 0.71. Specific challenges were noted in accurately classifying classes with lower sample sizes or lower data quality, such as Moth (F1-score of 0.43) and Spider (F1-score of 0.36).

Upon integrating the Kaggle dataset, the model demonstrated improved performance metrics (Table. 3), including an overall accuracy increase to 76% and a weighted average F1-score of 0.76. Notably, precision and recall for classes like Thrips and Melanogaster remained high, underscoring enhanced model sensitivity and specificity with larger, more diverse datasets.

Further inclusion of the DLFautoinsects dataset, yielded a nuanced effect on performance (see appendix Tables 10 and 11). While some classes like Water Drop and Reflection maintained high scores, others, such as Spider and Moth, did not show expected improvements. This suggests a performance plateau, possibly indicating model saturation where adding more data does not proportionally enhance outcomes.

Table 3 shows the combined aphids and agricultural pests datasets and the related per-class metrics. The per-class confusion matrices and metrics for the additional datasets are detailed in the appendix B.2, providing a granular view of model performance across varied conditions. These findings emphasize the potential benefits of dataset diversity up to a saturation point, beyond which the marginal gains diminish.

4.4 Experiment 3

The goal in this experiment is to simulate practice as close as possible, where more classes are available as base, and classes are slowly added, to determine the point at which the model's performance begins to degrade significantly. We investigated the performance of the AM-TEEN model when trained with three combined Aphids, Agricultural Pests, and DLFautoinsects datasets. Once all datasets are combined, the total 62 classes were divided into 36 base classes and the 26 others were added one per incremental session. The results indicate that the model holds up well initially but begins to show a slight decline as more classes are added. Despite this, the performance drop is not drastic. As mentioned, we started with 36 classes (60% of the total) as the base and incrementally added the remaining classes one at a time per session.

Fig. 10 illustrate the accuracy and F1 score of the AM-TEEN model across the incremental sessions. Initially, the model maintains high accuracy, but this gradually decreases as more classes are introduced. After adding approximately 17 classes, the accuracy drops by about 7%, which seems to be the largest dip in performance.

4.5 Results

The results from our experiments on the Aphids, Agricultural Pests, DLFautoinsects, and CIFAR-100 datasets using the TEEN, A-TEEN, and AM-TEEN models demonstrate that the median method outperforms mean and mode methods in maintaining accuracy across sessions. The inclusion of attention mechanisms further boosts performance, ensuring more stable and representative prototypes. Despite these successes, challenges remain, such as the misclassification of non-insect elements like water drops and reflections as insects, highlighting areas for further model refinement. Fig. 11 shows one the misclassified examples.

5 Discussion, Conclusion & Future Work

As the global demand for agricultural productivity increases, the need for advanced pest management solutions becomes ever more critical. This paper presented the AM-TEEN model, a novel application of Few-Shot Class Incremental Learning (FSCIL) techniques designed to address the persistent challenges of pest detection in agriculture. By integrating median averaging and attention mechanisms, this model sets a new benchmark for accuracy and adaptability in comparison to base models. In this section, we summarize the key findings, discuss the implications of our research, and propose directions for future investigations to extend the capabilities of incremental learning models.

5.1 Discussion

The integration of median prototyping and a channel attention mechanism enhanced the performance of the AM-TEEN model,

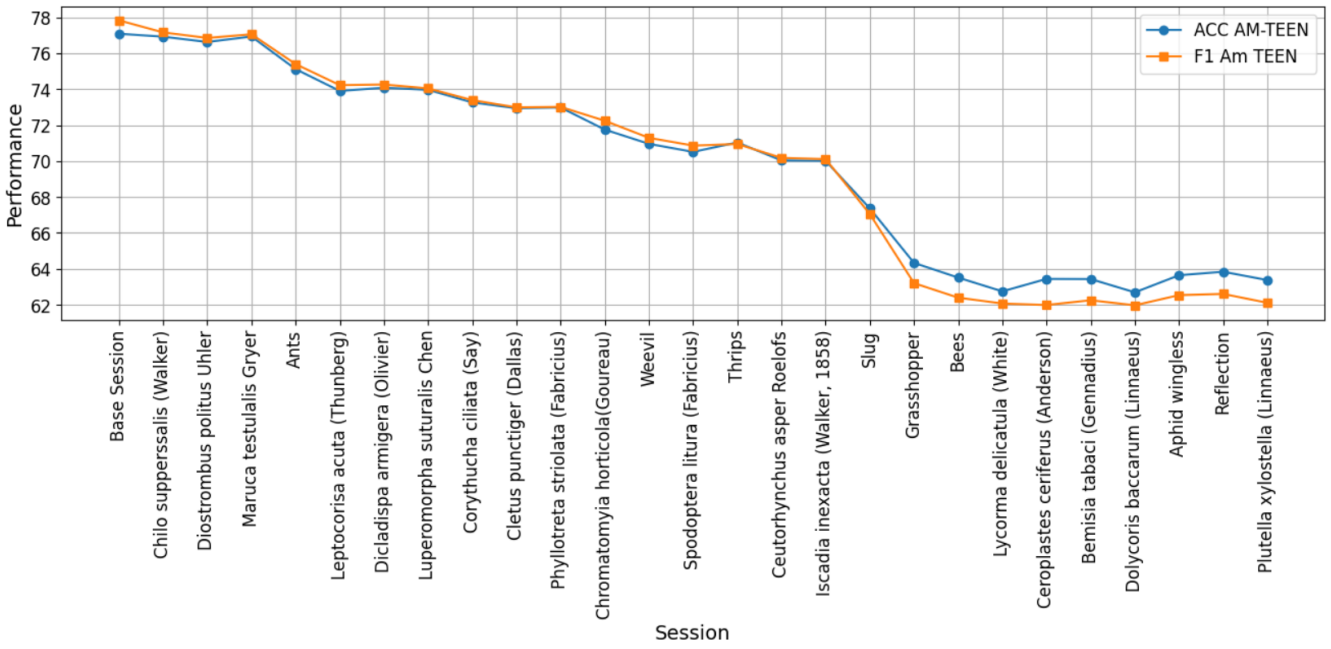
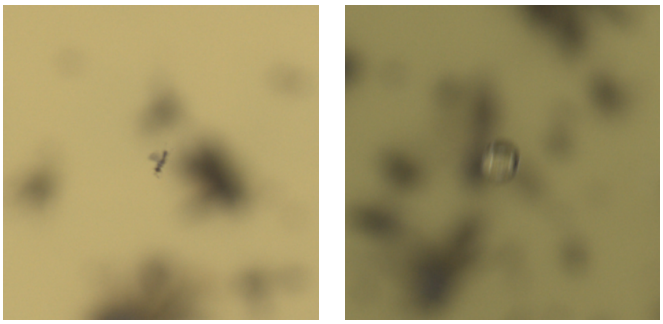


Fig. 10: Accuracy and F1 Score of the AM-TEEN Model across Incremental Sessions



(a) Wasp (b) Water drop

Fig. 11: The misclassification of a water drop for an insect

leading to more accurate classifications and demonstrating the effectiveness of these enhancements. The key takeaway is that these methodologies not only optimized prototype generation but also ensured that important features within the data were dynamically highlighted, contributing to more precise outcomes.

One of the main strengths of the AM-TEEN model is its robustness and adaptability. During external validation, the model maintained high accuracy levels across larger and more diverse datasets, highlighting its potential for real-world applications. The incorporation of attention mechanisms allowed the model to dynamically emphasize relevant features and de-emphasize irrelevant ones, thereby improving both precision and recall. This selective focus is particularly beneficial in imbalanced datasets where certain classes have fewer and more varied examples.

The performance of the median method can be attributed to its robustness against outliers [21]. By using the median of the embeddings to generate prototypes, this method effectively reduced the influence of anomalous data points, leading to more stable and representative prototypes. This characteristic is particularly beneficial in imbalanced datasets where outliers can disproportionately affect the mean, resulting in less reliable prototypes.

The channel attention mechanism [20] enhanced the model's performance by dynamically emphasizing more relevant features and de-emphasizing less relevant ones. This allowed the model to concentrate on the most informative parts of the feature space, thereby maintaining high accuracy and stability across incremental learning sessions.

The results from our experiments underscore the relevance of the AM-TEEN model for applications where new data classes are continually added over time. Its ability to maintain reasonable accuracy levels across multiple incremental sessions makes it an applicable tool for dynamic and evolving datasets, such as in agricultural pest monitoring or biodiversity studies where new species may be discovered and need to be integrated into existing models.

Overall, the AM-TEEN model significantly benefits from integrating an attention mechanism and median prototyping. It shows enhanced robustness in processing large datasets and better handles incremental learning scenarios compared to the baseline TEEN model. Despite these strengths, the model still struggles with accurately distinguishing between insects and non-insect elements, indicating limitations in its discriminative power. Variability introduced by different data collection methods (such as the Aphids data collection method versus other datasets) and class imbalances could affect performance, potentially biasing results towards the majority class. Additionally, unannounced instances of previously unseen classes present a challenge, creating a chicken and egg problem: we need several instances to update our model incrementally, but identifying these instances requires the model to be updated, which may require extending the model and/or user interface. Another critical consideration is domain shift, where the inference data distribution differs from the training data. While our experiments combined different datasets to enrich the feature space, future research should simulate domain shifts by training on one dataset and performing class-incremental learning on another to better reflect real-world scenarios.

5.2 Future Work

Looking ahead, there are several avenues for further research and development. One promising direction is inspired by the recent study "Future-Proofing Class Incremental Learning," [18] which explores the use of pre-trained text-to-image diffusion models to generate synthetic images for training feature extractors in class incremental learning settings. This approach could potentially address one of the key challenges in FSCIL by providing a method to enhance feature extractor training when limited classes are available initially.

Future experiments could integrate synthetic image generation into the AM-TEEN model, preparing the system for new, unseen classes without the need for extensive real data. This method has the potential to improve performance in data-scarce environments, leading to more cost-effective and scalable solutions for incremental learning. To address the challenge of identifying previously unseen classes, one proposed solution is to measure classification based on a threshold and filter out lower confidence predictions, assigning them to a new class category. When enough samples are accumulated, a new session can be triggered to add them to the model. Another approach could involve user interaction, allowing users to designate new examples as a new class and update the model once sufficient samples are collected. Additionally, future work should explore simulating domain shifts by training on one dataset and incrementally learning from another to assess the model's adaptability and robustness with varied data inputs. Incorporating these strategies will enhance the model's practicality and robustness for real-world applications, ultimately leading to improved pest management solutions in agriculture and other domains.

5.3 Conclusion

In conclusion, the AM-TEEN model shows promise in improving the efficiency and accuracy of pest detection systems through the innovative use of FSCIL techniques. Future research will focus on overcoming the existing limitations and exploring new technologies like synthetic data generation to enhance the model's adaptability and performance in dynamic agricultural environments.

Acknowledgements

- This project is financially supported by ELFPO and performed within the POP3+ Fryslan project Innovatie luizendetectie.

References

1 C. A. Dedryver, A. Le Ralec, F. Fabre: The conflicting relationships between aphids and men: A review of aphid damage and control strategies, C. R. Biologies 333, pp539–553, 2010.

2 D. W. Ragsdale, D. A. Landis, J. Brodeur, G. E. Heimpel, N. Desneux: Ecology and Management of the Soybean Aphid in North America, Annual Review of Entomology 56:1, 375-399, 2011.

3 J. C. K. NG, K. L. PERRY: Transmission of plant viruses by aphid vectors, MOLECULAR PLANT PATHOLOGY 5(5), 505–511, 2004.

4 C. Bass, A. M. Puinean, C. T. Zimmer, I. Denholm, L. M. Field, S. P. Foster, O. Gutbrod, R. Nauen, R. Slater, M. S. Williamson: The evolution of insecticide resistance in the peach potato aphid, Myzus persicae, Insect Biochemistry and Molecular Biology 51 41e51, 2014.

5 E. A. Linsa, J. P. Mazuco Rodrigueza, S. I. Scoloskia, J. Pivatob, M. B. Limab, J. M. C. Fernandesa, P. R. V. da Silva Pereira, D. Laub, R. Riedera: A method for counting and classifying aphids using computer vision, Computers and Electronics in Agriculture 169, 1052, 2020.

6 Y. Wang, Q. Yao, J. T. Kwok, L. M. Ni: Generalizing from a few examples: A survey on few-shot learning, ACM computing surveys (csur), 53(3), pp.1-34, 2020.

7 X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, Y. Gong: Few-shot class-incremental learning, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

8 J. C. Gomes, D. L. Borges: Insect Pest Image Recognition: A Few-Shot Machine Learning Approach including Maturity Stages Classification, Agronomy 12, 1733, 2022.

9 X. Chen, H. Liang: An Optimized Class Incremental Learning Network with Dynamic Backbone Based on Sonar Images, Mar. Sci. Eng. 11, 1781, 2023.

10 S. Tian, L. Li, W. Li, H. Ran, X. Ning, P. Tiwari: A survey on few-shot class-incremental learning, Neural Networks 169, 307–324, 2024.

11 M. Hersche, G. Karunaratne, G. Cherubini, L. Benini, A. Sebastian, A. Rahimi: Constrained few-shot class-incremental learning, In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 9057–9067), 2022.

12 A. Cheraghian, S. Rahman, P. Fang, S. Kumar Roy, L. Petersson, M. Harandi: Semantic-Aware Knowledge Distillation for Few-Shot Class-Incremental Learning, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2534-2543, 2021.

13 D. W. Zhou, F. Y. Wang, H. J. Ye, L. Ma, S. Pu, D. C. Zhan: Forward Compatible Few-Shot Class-Incremental Learning, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9046-9056, 2022.

14 Q.W. Wang, D.W. Zhou, Y.K. Zhang, D.C. Zhan, H.J. Ye: Few-Shot Class-Incremental Learning via Training-Free Prototype Calibration, P37th Conference on Neural Information Processing Systems NeurIPS, 2023.

15 K. He, X. Zhang, S. Ren, J. Sun: Deep Residual Learning for Image Recognition, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

16 H. F. van Emden, R. Harrington: Aphids as crop pests 2nd Edition, Wallingford, Oxfordshire, CAB International, pp. 362-381, 2017.

17 C. Xie, R. Wang, J. Zhang, P. Chen, W. Dong, R. Li, J. Yu: Multi-level learning features for automatic classification of field crop insects, Computers and Electronics in Agriculture, 152: 233–241, 2018.

18 Q. Jodelet, X. Liu, Y. Jun Phua, T. Murata: Future-Proofing Class Incremental Learning, arXiv preprint arXiv:2404.03200, 2024.

19 M. Ren, R. Liao, E. Fetaya, R, S. Zemel: Incremental few-shot learning with attention attractor networks, Advances in neural information processing systems 32, 2019.

20 G. Karunaratne, M. Schmuck, M. Le Gallo, G. Cherubini, L. Benini, A. Sebastian, A. Rahimi: Robust high-dimensional memory-augmented neural networks, Nat Commun 12, 2468, 2021.

21 Y. Li, Y. Chi, H. Zhang, Y. Liang: Non-convex low-rank matrix recovery with arbitrary outliers via median-truncated gradient descent, Information and Inference: A Journal of the IMA, Volume 9, Issue 2, Pages 289–325, 2020.

Appendix

A Materials & Methods

A.1 Imaging Setup

The images are gathered from an imaging setup placed in both indoor and outdoor locations on farms and greenhouses in the northern provinces of the Netherlands. The setup is installed at the location and powered on to start capturing images of moving, flying insects and objects. A secondary camera scans a line of light at the focal point of the main camera. If any changes occur in the pixel range of the streaming line of light, the main camera and the flashlights are triggered to take a shot. A yellow plate is also positioned within the range of the main camera to provide a uniform background for the images. The entire system is equipped with a GPU to process the captured images. The system can be controlled remotely via a Wi-Fi modem, and the network connection is also used to transfer the images to cloud storage. Fig. 12 shows the setup at a farm.

The hardware specification details of the setup in use for gathering the dataset is as follows:

- Main Camera:
  - IDS 3990CP-C-HQ
  - https://en.ids-imaging.com/store/u3-3990cp-rev-2-2.html
  - Main Camera lens:
  - Fujifilm CF25ZA-1S
  - https://en.ids-imaging.com/store/lens-fujifilm-cf25za-1s-25-mm-1-1.html
- Line scan Camera:
  - IDS 3060CP-M-GL
  - https://en.ids-imaging.com/store/ui-3060cp-rev-2.html
  - operating in 1936 x 2 resolution, 2600fps
  - Linescan Camera lens:
  - Fujifilm HF25XA-5M
  - https://en.ids-imaging.com/store/lens-fujifilm-hf25xa-5m-25-mm-2-3.html
- Waterproof casings for both cameras:
  - https://www.get-cameras.com/Machine-vision-aluminium-camera-housing-enclosure-waterproof-IP67?Product=864105090&Lng=en
- Main Illumination:
  - Custom-built by Vision Hardware Partner. 4x 20cm white LED bar lights
  - https://www.vhponline.nl/2017/04/04/bar-light-30mm-wide
  - Operates at 37V/12.5A for all four bars, 100µs pulse length
  - LED driver:
  - VHP Lightning 12F
  - https://www.vhponline.nl/2018/07/16/lightning-12f
- Line illumination:
  - 40cm Luxalight LED bar, wavelength 735nm, 60W power
  - https://www.luxalight.eu/en/products/led-engine/luxalight-industrial-led-fixture-opaline-cover-far-red-735nm-242x16mm-24-volt
  - Operates at 37V/12.5A for all four bars, 100µs pulse length
  - LED driver:
  - VHP Lightning 12F
  - https://www.vhponline.nl/2018/07/16/lightning-12f
- GPU:
  - NVIDIA Jetson Orin Nano 8GB with 1TB SSD
  - https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/
- 5G router:
  - Teltonika RUTX50
  - https://teltonika-networks.com/products/routers/rutx50

Fig. 12: Imaging setup in the farm

B EXPERIMENTS & RESULTS

B.1 Experiment 1

The full metrics tables

Table 4: Unseen Accuracy per session for the three different prototype generation methods (mode, mean and median) on Aphids

Method	Model	Session 0	Session 1	Session 2	Session 3	Session 4	Session 5	Session 6	Session 7	Session 8
Mean	TEEN	-	100	85.85	67.22	67.95	67.58	71.53	68.46	61.16
	A-TEEN	-	100	83.96	67.22	71.54	68.95	72.67	72.03	61.76
Mode	TEEN	-	75.00	54.48	57.15	66.16	60.55	65.04	59.00	52.69
	A-TEEN	-	75.00	62.03	65.09	70.60	61.71	63.92	61.28	53.39
Median	TEEN	-	100	82.07	70.75	76.46	74.39	76.89	73.78	67.63
	AM-TEEN	-	100	87.74	75.16	79.86	74.42	76.92	73.15	69.45

Table 5: Seen Accuracy per session for the three different prototype generation methods (mode, mean and median) on Aphids

Method	Model	Session 0	Session 1	Session 2	Session 3	Session 4	Session 5	Session 6	Session 7	Session 8
Mean	TEEN	-	81.33	81.33	81.33	81.33	81.33	81.33	81.33	81.33
	A-TEEN	-	91.33	84.67	84.67	84.67	84.67	84.67	84.67	84.67
Mode	TEEN	-	94.67	94.67	94.67	94.67	94.67	94.67	94.67	94.67
	A-TEEN	-	94.67	94.67	94.67	94.67	94.67	94.67	94.67	94.67
Median	TEEN	-	91.33	91.33	91.33	91.33	88.00	88.00	88.00	88.00
	AM-TEEN	-	98.00	98.00	98.00	98.00	91.33	91.33	91.33	91.33

Table 6: Harmonic-Mean per session for the three different prototype generation methods (mode, mean and median) on Aphids

Method	Model	Session 0	Session 1	Session 2	Session 3	Session 4	Session 5	Session 6	Session 7	Session 8
Mean	TEEN	-	89.71	83.53	73.60	74.04	73.82	76.12	74.34	69.82
	A-TEEN	-	95.47	84.31	74.94	77.55	76.00	78.21	77.84	71.42
Mode	TEEN	-	85.71	70.53	72.74	79.63	75.43	78.82	74.21	69.01
	A-TEEN	-	85.71	76.56	78.86	82.77	76.32	77.99	75.99	69.61
Median	TEEN	-	95.47	86.46	79.74	83.24	80.62	82.07	80.26	76.48
	AM-TEEN	-	98.99	92.58	85.07	88.00	82.01	83.51	81.24	78.90

Table 7: Unseen Accuracy per Session for Base and AM-TEEN Models on Aphids, CIFAR-100, Aphids + Agricultural Pests(Aphids+), and Aphids + Agricultrual Pests + DLFautoinsects(Aphids++)

Dataset	Model	Session 0	Session 1	Session 2	Session 3	Session 4	Session 5	Session 6	Session 7	Session 8	Session 9
Aphids	TEEN	-	100	82.07	70.75	76.46	74.39	76.89	73.78	67.63	NA
	AM-TEEN	-	100	87.74	75.16	79.86	74.42	76.92	73.15	69.54	NA
CIFAR-100	TEEN	-	27.60	24.80	22.87	21.85	22.64	23.77	23.11	22.87	NA
	AM-TEEN	-	30.40	28.40	25.53	25.05	25.16	25.50	24.71	24.10	NA
Aphids +	TEEN	-	74.38	73.89	67.72	65.79	61.00	61.80	60.25	59.33	62.86
	AM-TEEN	-	79.80	79.42	74.75	71.97	66.44	66.42	65.55	63.27	66.32
Aphids ++	TEEN	-	72.77	74.64	70.31	67.98	68.78	68.81	67.70	64.77	64.34
	AM-TEEN	-	73.47	76.14	70.03	69.79	69.82	69.08	69.15	66.08	65.84

B.2 Experiment 2

The full metrics tables and figures

B.3 Datasets

The datasets complementary images and information

Table 8: Seen Accuracy per Session for Base and AM-TEEN Models on Aphids, CIFAR-100, Aphids + Agricultural Pests(Aphids+), and Aphids + Agricultrual Pests + DLFautoinsects(Aphids++)

Dataset	Model	Session 0	Session 1	Session 2	Session 3	Session 4	Session 5	Session 6	Session 7	Session 8	Session 9
Aphids	TEEN	-	91.33	91.33	91.33	91.33	88.00	88.00	88.00	88.00	NA
	AM-TEEN	-	98.00	98.00	98.00	98.00	91.33	91.33	91.33	91.33	NA
CIFAR-100	TEEN	-	75.57	74.95	74.05	73.12	71.97	71.32	71.02	70.40	NA
	AM-TEEN	-	75.85	75.22	74.45	73.78	72.65	72.08	71.93	71.05	NA
Aphids +	TEEN	-	95.24	94.85	93.45	91.27	85.27	84.88	84.88	84.63	84.63
	AM-TEEN	-	93.15	92.76	92.51	90.47	80.08	80.08	78.90	78.90	78.65
Aphids ++	TEEN	-	82.96	81.37	79.58	79.25	76.43	75.84	75.59	74.82	74.82
	AM-TEEN	-	88.69	86.03	84.52	84.39	81.64	79.62	79.62	79.49	78.83

Table 9: Harmonic Mean per Session for Base and AM-TEEN Models on Aphids, CIFAR-100, Aphids + Agricultural Pests(Aphids+), and Aphids + Agricultrual Pests + DLFautoinsects(Aphids++)

Dataset	Model	Session 0	Session 1	Session 2	Session 3	Session 4	Session 5	Session 6	Session 7	Session 8	Session 9
Aphids	TEEN	-	95.47	86.46	79.74	83.24	80.62	82.07	80.26	76.48	NA
	AM-TEEN	-	98.99	92.58	85.07	88.00	82.01	83.51	81.24	78.90	NA
CIFAR-100	TEEN	-	40.43	37.27	34.94	33.65	34.44	35.65	34.88	34.53	NA
	AM-TEEN	-	43.40	41.23	38.02	37.40	37.38	37.67	36.79	35.99	NA
Aphids +	TEEN	-	83.53	83.07	78.53	76.46	71.12	71.53	70.48	69.76	72.14
	AM-TEEN	-	85.96	85.58	82.69	80.16	72.63	72.61	71.61	70.23	71.96
Aphids ++	TEEN	-	77.53	77.86	74.66	73.18	72.40	72.16	71.43	69.43	69.18
	AM-TEEN	-	80.36	80.78	76.59	76.40	75.27	73.98	74.01	72.17	71.75

Table 10: External Validation per Class Metrics on Aphids

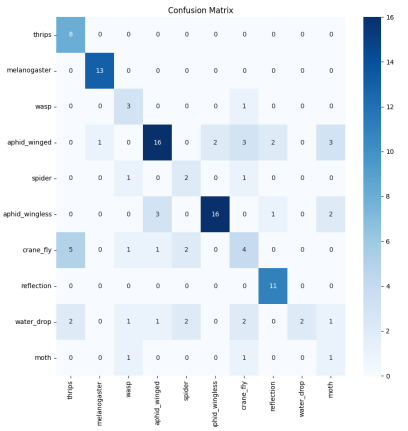
Class	Precision	Recall	F1-Score
Thrips	0.50	1.00	0.67
Melanogaster	1.00	1.00	1.00
Wasp	0.60	0.75	0.67
Winged Aphid	0.82	0.67	0.73
Spider	0.29	0.50	0.36
Wingless Aphid	0.84	0.73	0.78
Crane Fly	0.62	0.38	0.48
Reflection	0.92	1.00	0.96
Water Drop	1.00	0.27	0.43
Moth	0.27	1.00	0.43
Weighted Average	0.79	0.71	0.71

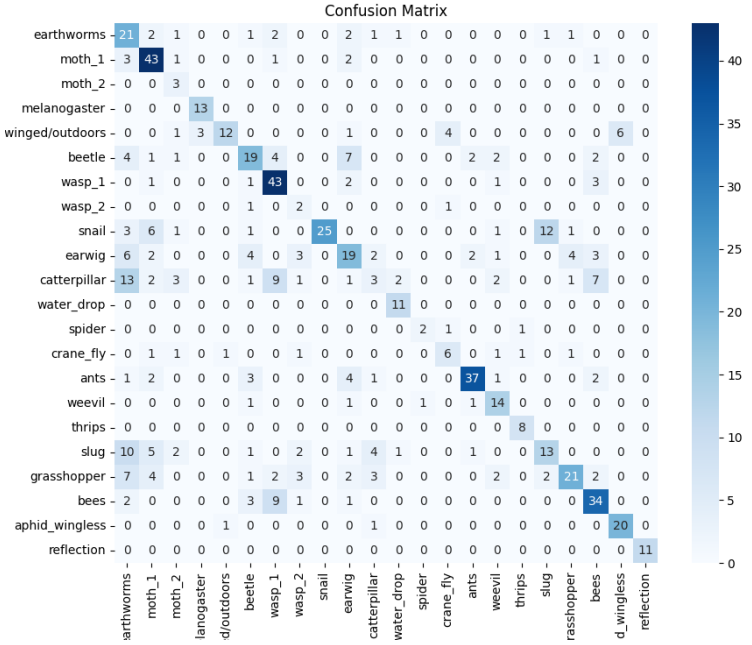
Table 11: External Validation per Class Metrics on Aphids, Agricultural Pests, and DLFautoinsects

Class	Precision	Recall	F1-Score
Thrips	0.73	1.00	0.84
Melanogaster	0.76	1.00	0.87
Wasp	0.67	0.50	0.57
Winged Aphid	0.93	0.48	0.63
Spider	0.25	0.25	0.25
Wingless Aphid	0.74	0.91	0.82
Crane Fly	0.67	0.62	0.64
Reflection	1.00	1.00	1.00
Water Drop	1.00	0.82	0.90
Moth	0.33	0.67	0.44
Weighted Average	0.80	0.75	0.75

Confusion Matrix per Class for Aphids and Agricultural Pests from Kaggle

Confusion Matrix per Class for Aphids





Confusion Matrix per Class for Aphids, Agricultural Pests from Kaggle, and DLFautoinsects

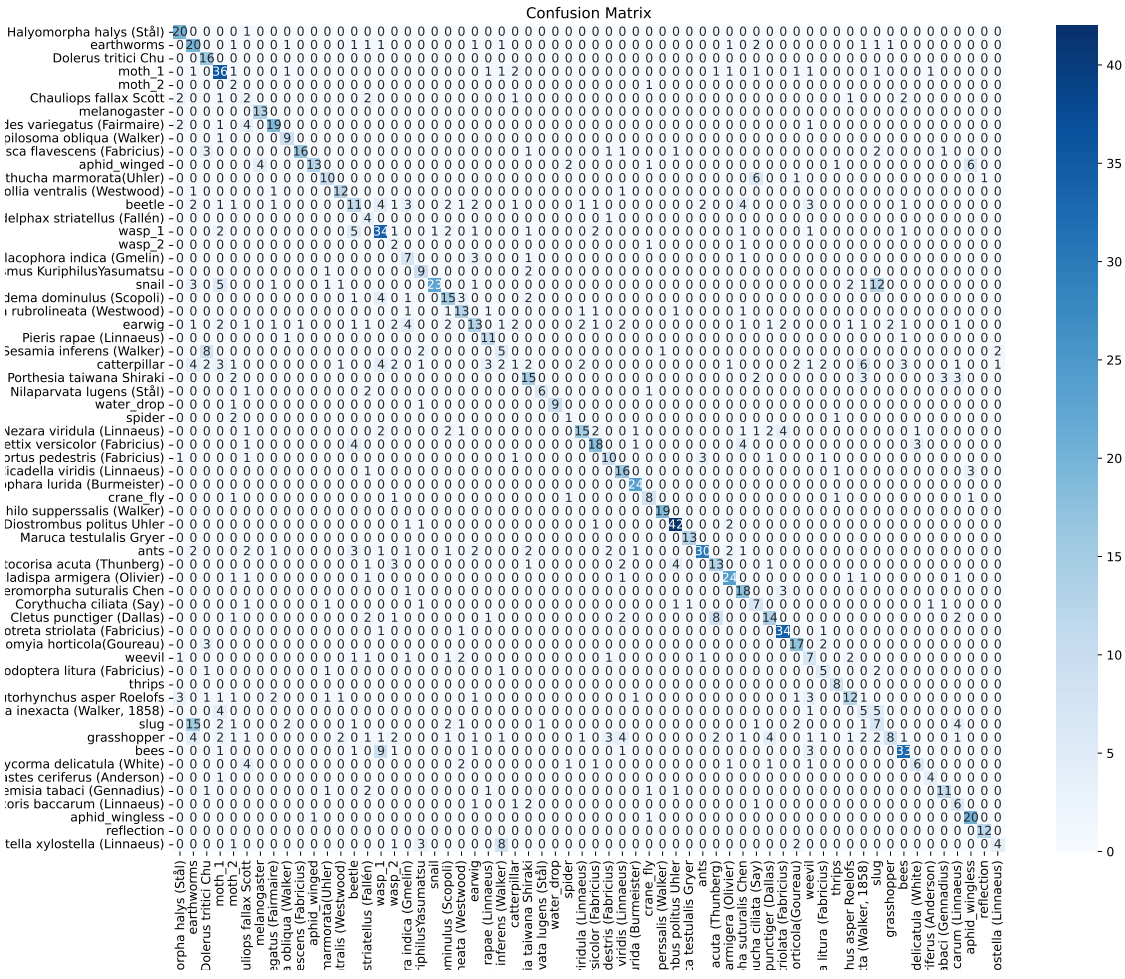


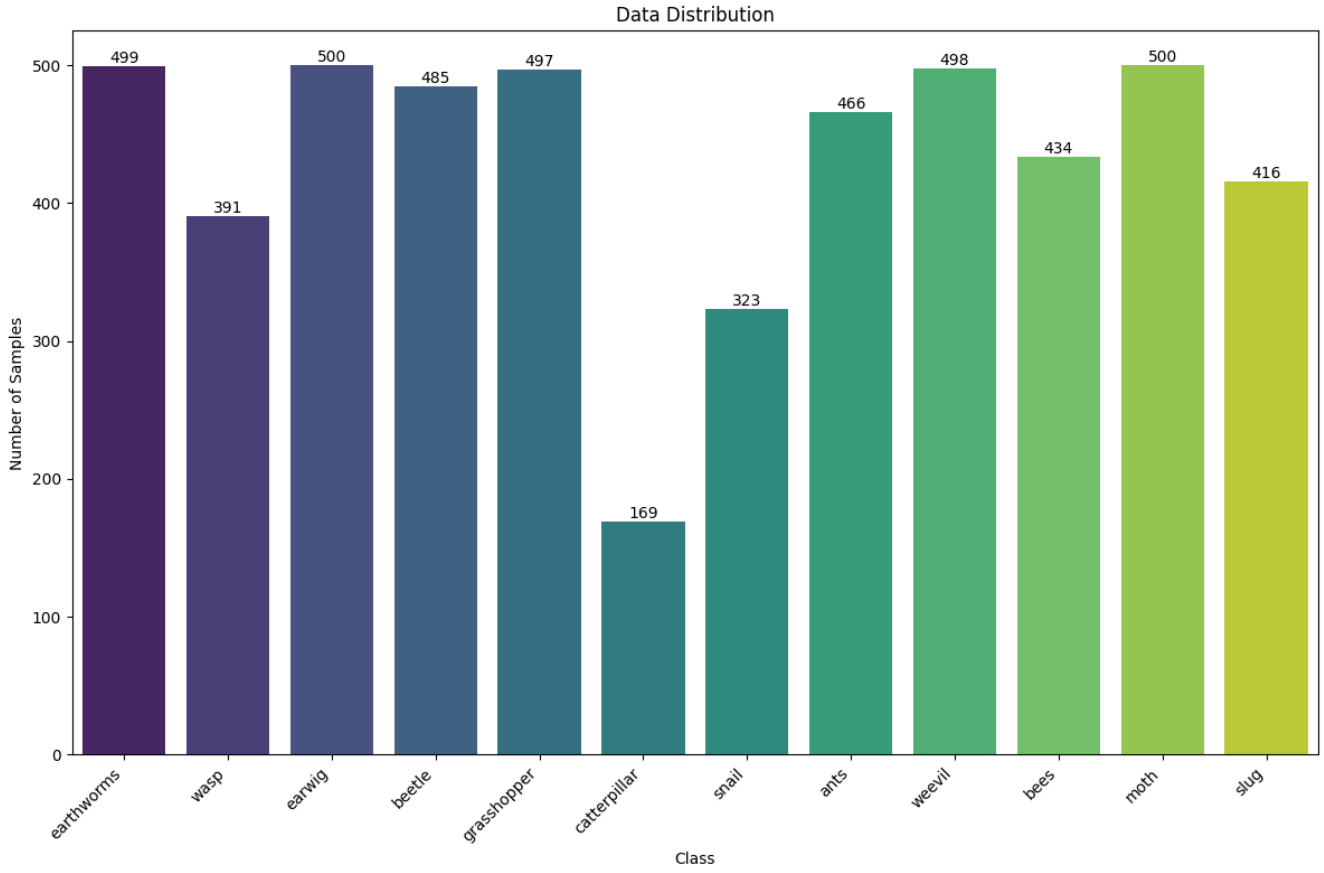
Fig. 13: Confusion Matrices for Various Datasets

The Aphids Classes: Aphid winged-outdoors, Aphid wingless, Crane fly, Melanogaster, Moth, Spider, Thrips, and Wasp

The Agricultural Pests Classes: Beetle, Snail, Catterpillar, Wasp, Earwig, Earthworms, Weevil, Moth, Grasshopper, Bee, Slug, Ants

The DLFautoinsects Classes: Halyomorpha halys (Stål), Dolerus tritici Chu, Chauliops fallax Scott, Strongyllodes variegatus (Fairmaire), Spilosoma obliqua (Walker), Empoasca flavescens (Fabricius), Corythucha marmorata(Uhler), Stollia ventralis (Westwood), Laodelphax striatellus (Fallén), Aulacophora indica (Gmelin), Dryocosmus KuriphilusYasumatsu, Eurydema dominulus (Scopoli), Graphosoma rubrolineata (Westwood), Pieris rapae (Linnaeus), Sesamia inferens (Walker), Porthesia taiwana Shiraki, Nilaparvata lugens (Stål), Nezara viridula (Linnaeus), Callitettix versicolor (Fabricius), Riptortus pedestris (Fabricius), Cicadella viridis (Linnaeus), Scotinophara lurida (Burmeister), Chilo supperssalis (Walker), Diostrombus politus Uhler, Maruca testulalis Gryer, Leptocorisa acuta (Thunberg), Dicladispa armigera (Olivier), Luperomorpha suturalis Chen, Corythucha ciliata (Say), Cletus punctiger (Dallas), Phyllotreta striolata (Fabricius), Chromatomyia horticola(Goureau), Spodoptera litura (Fabricius), Ceutorhynchus asper Roelofs, Iscadia inexacta (Walker, 1858), Lycorma delicatula (White), Ceroplastes ceriferus (Anderson), Bemisia tabaci (Gennadius), Dolycoris baccarum (Linnaeus), Plutella xylostella (Linnaeus)

Agricultural Pests Class Distribution



DLFautoinsects Class Distribution