Brief Paper

Exploiting Unlabeled Images via Pseudo-Labelling and Paste-In Augmentation for Insect Localisation in Automated Monitoring

Hui Yu[1] *Joachim Denzler*[1] *Dennis Böttger*[2] *Gunnar Brehm*[2] *Paul Bodesheim*[1]

[1] *Computer Vision Group, Friedrich Schiller University Jena, 07743 Jena, Germany*
[2] *Phyletic Museum, Friedrich Schiller University Jena, 07743 Jena, Germany*
** E-mail: {hui.yu, joachim.denzler, dennis.boettger, gunnar.brehm, paul.bodesheim}@uni-jena.de*

Abstract:
Insect monitoring using an automated deep learning pipeline has become increasingly important in understanding the crisis of insect decline. Advanced model architectures trained with high-resolution images are essential to ensure the quality of insect localisation and species identification. Recent methods struggle with limited annotated data, which requires time-consuming manual labelling for bounding boxes and domain expert-level knowledge for insect categorisation. In this paper, we present a comprehensive benchmark of object detection models for this task, evaluating YOLOv9 and SSD architectures across three distinct datasets: EU-Moths, NID-Moths, and AMI-Traps. Our experiments reveal that high-resolution inputs are a dominant factor for accurate insect localisation, with performance improving substantially with larger image sizes. In addition, we perform cross-dataset validation to verify the generalisation capabilities of YOLOv9 on these datasets, justifying the choice of the AMI-Traps dataset as our pre-training dataset for obtaining a robust detector. Finally, to leverage large amounts of unlabeled data, we investigate a pseudo-labelling and paste-in data augmentation strategy. While this technique provides only modest improvements in overall detection metrics, qualitative analysis demonstrates that it enhances model robustness, enabling the detection of insects in challenging, low-contrast conditions where a strong baseline model would otherwise fail. In our experiments, YOLOv9 outperforms SSD on the one-class NID-Moths and AMI-Traps datasets with average precisions of 0.951 and 0.742, respectively. On the binary-class AMI-Traps dataset, a larger YOLOv9 model with a 1280x1280 input resolution achieves an average precision of 0.972 for the *moth* category. These results indicate the importance of data-centric approaches and high-resolution imagery for building effective automated insect monitoring systems.

1 Introduction

To understand the reason for the crisis of insect decline [6, 13, 30, 34] in recent years, non-harmful camera trap systems that capture high-resolution images are becoming an increasingly essential tool for continuous and non-intrusive insect monitoring in their natural habitats [2, 8, 14, 20, 28, 33]. These modern systems ensure the observation of insect behaviour and population dynamics without causing additional harm. High-resolution images are essential within these pipelines as they provide fine-grained details for accurate localisation and classification of the insect specimens. This led to the availability of large, high-resolution image datasets, such as the AMI-Traps dataset [15] and the NID-Moths dataset [20], essential for developing robust automated monitoring pipelines. However, the decent performance of deep neural network models critically depends on large-scale monitoring datasets with high-quality annotations.

Hence, this data bottleneck is often the main challenge for practitioners to train and deploy robust AI models. Although modern camera traps can generate vast corpora of unlabeled images, the process of annotating images is time-consuming and expensive. Thus, there is a noticeable imbalance between unlabeled data and available labelled data. Therefore, our objective is to effectively leverage the rich information source of large unlabelled data, specifically a large fraction of the NID-Moths dataset, to improve the performance and robustness of insect detectors without manual annotation.

A common semi-supervised learning approach involves generating pseudo-labels for unlabeled data using a model trained on a labeled set. However, simply training on all generated pseudo-labels can be detrimental, as it introduces noise and low-quality instances, potentially degrading model performance. To overcome this issue, we propose a novel paste-in data augmentation strategy that only leverages high-quality pseudo-labels from unlabeled camera trap images, as shown in Fig. 1. Our approach is inspired by [10], which incorporates additional data by pasting pseudo-labelled objects from

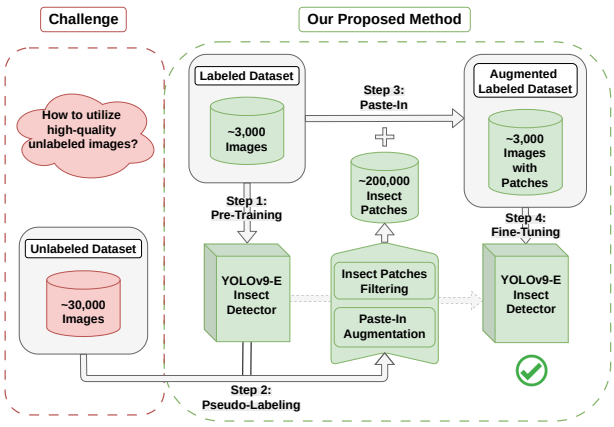


Fig. 1: Overview of our pseudo-labelling and paste-in approach with pseudo-label generation for bounding boxes. A pre-trained YOLOv9-E insect detector generates pseudo-labels for images from an unlabeled dataset. The high-quality insect patches were then filtered and pasted into images of the labelled dataset as data augmentation. Lastly, we fine-tune the same YOLOv9-E model on the augmented data, which improves the overall localisation performance.

the unlabeled dataset to the labelled one [23]. Unlike conventional methods, which add entire unlabeled images with potentially noisy backgrounds to the training set, our approach first uses a robust, pre-trained detector to generate high-quality pseudo-labels from the unlabelled dataset. Then it selectively extracts high-confidence insect patches and pastes them into existing labelled images. This technique enriches the training data by increasing the number of insect instances per image while preserving the content of the original high-quality labelled scenes, thereby improving the robustness of the detector.

In this paper, we demonstrate the effectiveness of this strategy and provide a comprehensive benchmark for insect detection in images from automated insect monitoring camera systems. Our key contributions are as follows. First, we benchmark the detection models YOLOv9 [36] and SSD [24] on three distinct insect datasets: EU-Moths, NID-Moths [19, 20] and AMI-Traps, providing an updated performance baseline. Second, we perform a cross-dataset validation to quantitatively assess model generalizability, justifying our choice of the AMI-Traps dataset for training a robust "teacher" detector for our pseudo-labelled paste-in strategy. Finally, we introduce and evaluate our pseudo-labelled paste-in augmentation method. We show that while the quantitative gains are modest, the strategy can improve a strong YOLOv9 baseline's robustness by leveraging unlabeled data.

2 Related Work

Deep learning in computer vision has become increasingly popular in image classification and detection tasks, as deep learning algorithms with advanced model architectures clearly outperform traditional methods. Using a large-scale dataset with high-quality annotations is important for training deep learning models. Thus, researchers and specialists across different academic fields work together in interdisciplinary projects to improve the architecture of computer vision models and to maintain high data quality by verifying annotations with expert knowledge in the insect domain.

Detection, segmentation, and classification of insects are vital in agriculture, environmental monitoring, and biodiversity assessment [31]. Most of the research focuses on insect detection. For example, SSD (Single Shot MultiBox Detector) [24] is used particularly for applications like monitoring nocturnal insects with light-based camera traps on datasets such as EU-Moths and NID-Moths [19, 20], where it provides robust detection results. Another popular architecture is the YOLO (You Only Look Once) family of models, including YOLOv4 [3] for small insect pests [11] and YOLOv5 [16] for a lightweight insect detection system with field adaptation [21], which is highly effective due to its real-time capabilities and decent performance to detect small insects. These models are frequently deployed in automated monitoring trap systems, as demonstrated in studies using YOLOv8 for an optimised "Yolo-pest" system [12]. In addition, research on developing transformer-based models in object detection for general object detection tasks has recently become popular, with architectures such as DINO [38] and its extension Mask-DINO [22] pushing the limits of detection performance. Fine-grained distribution refinement of D-FINE [26] improves object localisation stability.

The availability of large and diverse datasets is vital for training these deep learning models. Insect-1M and the resulting insect foundation model [25] improve visual insect understanding and offer large-scale annotated data. Similarly, there is the BIOSCAN-1M insect dataset [9]. Given that each image includes only a single insect, these datasets are more appropriate for species classification tasks than for evaluating insect localisation performance. In addition, there are special datasets like the AMI dataset with a machine learning pipeline [14, 15], or the Flatbug dataset utilised with YOLOv8 and sliding window inference for terrestrial arthropod detection [29]. Furthermore, deep learning techniques, like knowledge distillation and GAN-based data augmentation from studies on the IDADP dataset [27], address data scarcity and improve model performance. Integrating these AI models into citizen science tools, such as smartphone-based applications for plant disease and insect pest detection using YOLOv8 [5], further improves insect monitoring and data collection.

Despite significant improvements in automated insect detection, the challenge of learning from limited annotated training data persists, especially when ensuring high-quality annotations, which are fundamental for training robust deep learning models. Tight bounding-box labels that cover all insects in an image are essential for training detectors, as low-quality annotations and incomplete sets of bounding boxes will mislead the model. Thus, annotation expertise plays an important role for insect labelling.

3 Method

3.1 Problem Statement

Training on large datasets with high-quality annotations is necessary to obtain robust insect detectors. Unlabeled images by far outnumber labelled ones, as insect monitoring devices constantly record new images, and it is time-consuming and simply infeasible to perform all annotations manually. However, we still want to exploit the huge amount of visual information in unlabeled images by proposing an augmented training set with pseudo-labels that still maintains high quality. Instead of training on all additional images with all generated pseudo-labels, which would also include those with high uncertainty, low quality, and noise, we adopt a pseudo-labelling and paste-in strategy for semi-supervised training to augment the annotated training data with high-quality patches only.

3.2 Baseline Detector

We selected YOLOv9 [36] as our baseline architecture because it represented a state-of-the-art object detection model at the beginning of our study and provides a high-performance, easy-to-use framework. While newer models have since been released, YOLOv9 serves as a robust baseline to evaluate the impact of our data augmentation strategy. Future work could explore the application of these methods to more recent architectures such as YOLOv10-12 [17, 32, 35].

For comparison, we also explored transformer-based object detectors. For instance, initial experiments with the insect detector D-FINE [26] have demonstrated its high localisation precision.

3.3 Paste-In Strategy

For our paste-in strategy, we exploit the bounding boxes generated from the baseline detector as pseudo-labels to augment the training data, as shown in Fig. 1. Hence, the first step consists of pre-training a baseline insect detector on manually labelled images only. The second step focuses on generating pseudo-labels for many unlabeled images, specifically from the NID-Moths dataset. This is achieved using the robust pre-trained object detector, YOLOv9-E, which was trained on AMI-Traps images with a training resolution of 1280x1280. The detector predicts the bounding boxes and

(a) Hard-edged paste-in. (b) Smooth-edged paste-in.

Fig. 2: Examples of images from the AMI-Traps dataset with paste-in patches extracted from the NID-Moths unlabeled dataset. Hard-edged patches (**left**) preserve the original insect visual features while bringing rectangular bias. Smooth-edge patches (**right**) contain less clear edges while mostly having a smearing effect.

their corresponding confidence scores for each image. Only predicted bounding boxes with a confidence score above a preset threshold are retained. This confidence threshold ensures that only high-quality and reliable pseudo-labels are generated, thereby minimising the introduction of noise into the training process. These pseudo-labels, comprising bounding box coordinates and class identifications, are then systematically stored to be readily accessible during the subsequent data loading phase of further model training or fine-tuning.

The last two steps integrate these generated pseudo-labels directly into the model fine-tuning process as a form of data augmentation. During the training phase, as images are loaded, a random selection from the pseudo-labelled patches are dynamically pasted into the current training image. To maintain the integrity of existing ground truth annotations on the training image and prevent detrimental overlap, the Intersection-over-Union (IoU) ratio is calculated between each pseudo-labelled patch and any pre-existing ground truth bounding box. As paste-in patches will replace the area of the original image, this calculation ensures that these patches do not obscure or interfere with ground-truth labels, thus preserving the quality of the training data. Naively pasting patches can introduce strong rectangular edge artifacts, which the model might learn as features. To mitigate this, we employ a seamless cloning technique from the OpenCV library [4]. This method smoothly blends the pasted insect patches with the background of the target image, creating a more realistic composite image and reducing edge-related biases, as shown in Fig. 2. This visualisation of the augmented images illustrates the difference between a hard-edged and our smooth-edged paste-in approach. The degree of this paste-in strategy is configurable. For each training image, we pre-define the number of unlabeled images to load and the maximum number of patches to be pasted. During the training phase, due to a high amount of image pre-processing time, we set these numbers to 1 and 10, respectively, to reduce the time for pre-processing.

As the pseudo-labelled insect patches are recorded with different devices and in other environments compared to the images of the dataset where they will be pasted in, we explicitly force the detector to learn more robust, background-invariant features. This process acts as a form of domain generalisation, training the model to recognise the intrinsic characteristics of the target insect regardless of the surrounding environment. This enhances the robustness and performance of the model, as supported by our experimental results. Furthermore, smooth-edged patches produced by the seamless cloning technique reduce background-related bias. The line chart in Fig. 3 shows that smooth-edged patches have a grey pixel intensity distribution more closely aligned with the AMI-Traps backgrounds than hard-edged patches, indicating that smooth-edged patches lead to less background-biased features.

4 Experiments

We focus on three datasets: EU-Moths [19], NID-Moths [20], and AMI-Traps [15]. They represent a gradient of visual and ecological complexity (Section 4.1). To ensure a fair comparison (Section 4.3.1), we adopt similar input resolutions, the same train-validation split, and one-class annotation format used in previous work [19]. For evaluation (Section 4.2), we report Average Precision (AP) scores using the PASCAL VOC metric [7], as these allow direct comparison with prior SSD results. In addition, we include the MS COCO (Microsoft Common Objects in Context) $AP_{50:95}$ metric [23], which offers a more rigorous evaluation of insect detectors under complex conditions. To justify our choice of the AMI-Traps dataset as the pre-training source for our pseudo-labelled paste-in strategy, we conduct cross-dataset validation (Section 4.3.2). We then perform experiments using a binary-class annotation format on the AMI-Traps dataset (Section 4.3.3), focusing specifically on *moth* detection and aiming to further enhance baseline performance using higher input resolutions and larger model sizes. Finally, we fine-tune a robust baseline detector on an augmented labelled dataset using our proposed pseudo-labelled paste-in augmentation method (Section 4.3.4). The experimental results confirm that our strategy

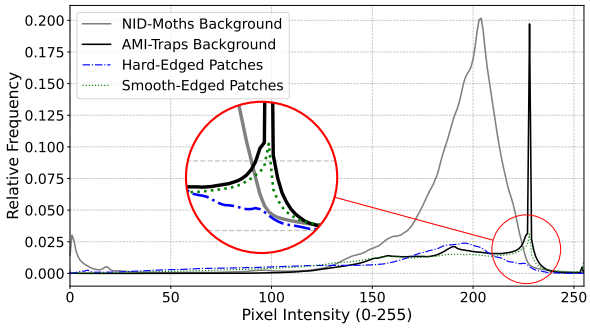


Fig. 3: Relative frequency of the grey pixel intensity from the patches of 100 random images from AMI-Traps and NID-Moths, hard-edged patches (NID-Moths) and smooth-edged patches (NID-Moths) by using our paste-in strategy. The backgrounds of NID-Moths images are darker and show a broader range of grey pixel intensities compared to those in the AMI-Traps dataset. Smooth-edged patches have a grey pixel intensity distribution more closely aligned with the AMI-Traps backgrounds than the hard-edged patches, indicated by the spike of the green curve in the highlighted region.

can effectively improve the detection performance of an already strong model.

4.1 Datasets

In the following, we provide a short overview of the three datasets we use in our experiments, which represent unique characteristics and challenges for insect detection in various real-world scenarios.

EU-Moths [19]: This dataset comprises 200 moth species commonly found in Central Europe. It features approximately 2,205 images with bounding box annotations for individual moths, making it suitable for fine-grained moth detection and species classification. As most images contain a single moth, this dataset is more suitable for classification than detection. However, to benchmark the performance of YOLOv9 and SSD on this dataset, we use the same train and validation split as in previous research [19] and evaluate it at the same image resolution.

NID-Moths [20]: The NID-Moths dataset consists of images captured by an automated moth scanner prototype in Central Europe, primarily focusing on nocturnal insects, especially moths. The images were taken with a 24-megapixel camera. It includes more than 27,000 images, with 818 images explicitly annotated with bounding boxes for 9,095 insects. Compared to EU-Moths, it focuses on moth detection, as most images contain multiple moths on a simple, uniform white panel background under different lighting conditions. All bounding boxes are of the same insect category. We adopted the same split for the annotated images as in [20].

AMI-Traps [15]: It is part of the larger AMI (Automated Monitoring of Insects) dataset. AMI-Traps consists of 2,893 expert-annotated images from automated camera traps, containing 52,948 labelled insects. The image resolution is 8 megapixels. This dataset presents more insect species compared to the NID-Moths dataset. Although both datasets have a similar white background, the lighting conditions are quite different. As the annotations ranged from family to species level for all insects on the images, we grouped them either into a single super-category, *insect*, or into two sub-categories, *moth* and *other-insect*, respectively. We do not consider bounding boxes from the *unidentifiable* category. As there are no statistics from previous research [15], we only report our experimental results.

4.2 Evaluation Metrics

For evaluating our insect detection models, we use the following evaluation metrics. First, we report average precision (AP) and mean average precision (mAP) following standard protocols from object

detection of Pascal VOC [7]. Since these metrics involve the computation of precision-recall curves, defining when a predicted bounding box is a true positive is required. As commonly done for object detection, we use different Intersection-over-Union (IoU) thresholds for the overlap of predicted and ground-truth bounding boxes. In particular, IoU thresholds 50% and 75% are used, with the latter focusing more on tight bounding boxes, and the corresponding value is reflected in the metrics as a subscript.

Second, we report the MS COCO benchmark metric [23], recognised for its comprehensive and nuanced assessment of detection performance. The core metric is $AP_{50:95}$, which calculates AP on multiple IoU thresholds ranging from 0.50 to 0.95 in increments of 0.05. $AP_{50:95}$ provides a robust measure of localisation accuracy, taking into account various levels of granularity for a predicted bounding box, including AP_{50} and AP_{75} from above.

Note that we use AP metrics to report results for single classes, which could either be one of the two classes *moth* or *other-insect*, or when considering all insects as part of one super-category *insect* and a trained detector for this single class only. We denote the latter as one-class detection. In contrast, mAP is reported when averaging the performance for the two classes *moth* and *other-insect* to obtain a single value for a whole dataset.

4.3 Results

4.3.1 One-Class Detection: The YOLOv9 architecture incorporates downsampling layers with a cumulative factor of 32, necessitating input image dimensions multiples of 32. Thus, we adopted an input size of 320 for YOLOv9 instead of 300 for SSD. For comparative analysis with SSD, separate training sessions were conducted using the compact YOLOv9 model (YOLOv9-C). Input resolutions were set to 320 for the EU-Moths dataset and to 320 and 512 for the other datasets. A pre-trained backbone on the MS COCO dataset has been used. Both detectors were trained with 100 epochs using the SGD optimiser with a learning rate of 0.01 and a batch size of 8. These preliminary experiments were crucial for understanding YOLOv9's performance in insect detection.

Our evaluation, presented in Table 1, indicates that SSD outperformed YOLOv9 in terms of AP_{50} on the EU-Moths dataset. This outcome can be attributed to the intrinsic characteristics of the EU-Moths dataset, containing manually recorded images. It is plausible that this dataset exhibits reduced variability in backgrounds, lighting conditions, or insect poses. Due to manual image capturing, the depicted insects primarily possess size and distribution for which the simpler yet well-optimised SSD architecture is already highly sufficient. In such scenarios, the more complex architectural advancements of YOLOv9, while generally offering superior performance in highly challenging and diverse contexts, may not yield substantial improvements.

Furthermore, for the NID-Moths dataset, which features much higher image resolution, different lighting conditions, and various levels of occlusion, YOLOv9 demonstrated superior performance over SSD in both AP_{50} and AP_{75} (see Table 1). In particular, even with an input size of 320, YOLOv9 outperforms SSD pre-trained with a higher input size of 512. The improvement of AP_{75} provides robust evidence of the effective localisation capabilities of YOLOv9.

Table 1 Experiment results with PASCAL VOC metrics on one-class datasets. The performance between SSD and YOLOv9 is similar on the EU-Moths dataset. For the NID-Moths dataset, For the NID-Moths dataset, YOLOv9, with a lower input size of 320, outperforms SSD, which has a size of 512. For the AMI-Traps dataset, YOLOv9 outperforms SSD with the same input size and further improves with a larger size of 512.

Model Name	Dataset	Input Size	AP_{50}	AP_{75}
SSD	EU-Moths	300	**0.990**	0.889
YOLOv9-C	EU-Moths	320	0.977	**0.898**
SSD	NID-Moths	512	0.912	0.262
YOLOv9-C	NID-Moths	320	0.920	0.478
YOLOv9-C	NID-Moths	512	**0.951**	**0.526**
SSD	AMI-Traps	300	0.361	0.067
YOLOv9-C	AMI-Traps	320	0.552	0.221
YOLOv9-C	AMI-Traps	512	**0.742**	**0.374**

As there are no quantitative experimental results for the AMI-Traps dataset, we trained SSD with an input size of 300 on this dataset, treating all bounding boxes as one class *insect*, with the same VGG16 pre-trained backbone and hyperparameter set as in [19]. For comparison, we trained YOLOv9-C with input sizes of 320 and 512 separately with the same training hyperparameters for the NID-Moths dataset. The performance of SSD is not satisfactory, especially in terms of AP_{75}, as shown in Table 1. YOLOv9 outperforms SSD in both the PASCAL VOC AP_{50} and AP_{75} metrics, showcasing its capability to deal with much more complex scenarios. In addition, with a larger input size of 512, the performance of YOLOv9 improves.

Table 2 Experiment results with MS COCO metrics of different sizes of YOLOv9 and input sizes on one-class AMI-Traps dataset. The metric scores improve more by increasing the input resolution than by adopting a larger model size.

Model Name	Input Size	AP_{50}	AP_{75}	$AP_{50:95}$
YOLOv9-C	640	0.808	0.438	0.452
YOLOv9-C	1280	**0.833**	0.547	0.517
YOLOv9-E	1280	0.832	**0.552**	**0.520**

As YOLOv9 achieved better performance in the AMI-Traps dataset, we conducted training with a larger model size and input size to push the limit of YOLOv9 further. Our experimental results are benchmarked using standard MS COCO metrics, as shown in Table 2. The findings consistently demonstrate that enhancing both the model size and the input resolution leads to substantial improvements in the metric scores for insect localisation. This highlights the critical role of resolution during training, as higher resolutions enable the neural network to effectively capture the fine details essential for accurately detecting small insects. The slight drop in AP_{50} from YOLOv9-C to YOLOv9-E can be attributed to the limited number of annotated training images for the larger number of parameters in the extended model YOLOv9-E.

4.3.2 Cross-Dataset Validation: To investigate cross-dataset performance, we trained YOLOv9 on different datasets, with all

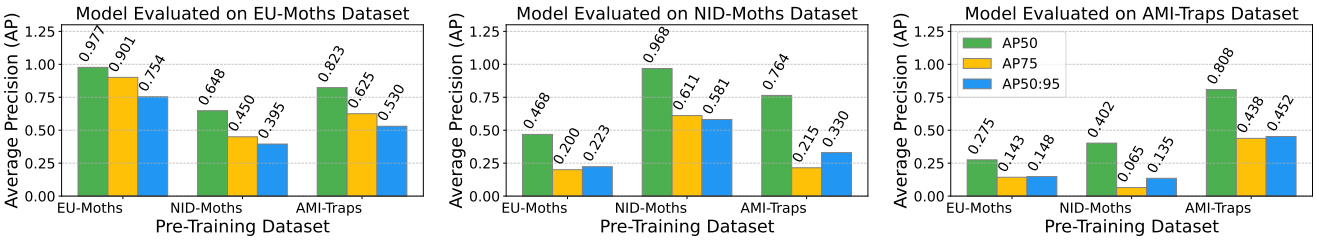


Fig. 4: Cross-dataset evaluation performance of YOLOv9 with MS COCO metrics on EU-Moths (**first**), NID-Moths (**second**), and AMI-Traps (**third**) datasets. The AMI-Traps dataset showcases more robust generalizability than the other two datasets.

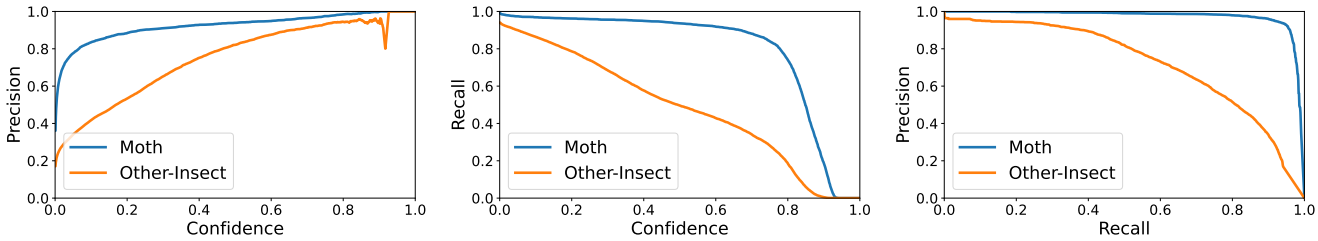


Fig. 5: Evaluation curves from YOLOv9-E-1280 on the AMI-Traps dataset: Precision-Confidence (**left**), Recall-Confidence (**middle**), and Precision-Recall (**right**). With a confidence threshold of 0.7, precision remains roughly 95%, and recall is 90%, indicating the pseudo labels from the *Moth* category are of high quality. However, the *Other-Insect* category does not show the same trend.

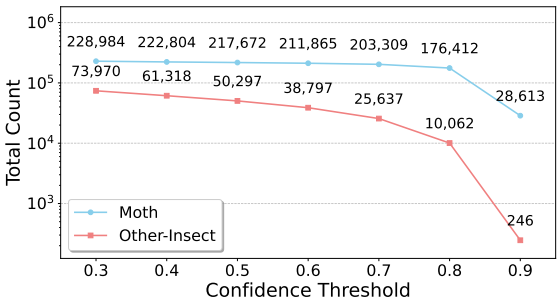


Fig. 6: The number of pseudo-labels from the NID-Moths unlabeled dataset under different confidence thresholds. The number of moths largely outnumbered other insects. Moreover, the number of moths remains above 200,000, with the confidence threshold lower than 0.7.

evaluation metric scores derived from MS COCO. We initially validated YOLOv9-C on the EU-Moths dataset. The model was trained on both the NID-Moths and AMI-Traps datasets at the same input size of 640. Notably, the YOLOv9-C model pre-trained on the AMI-Traps dataset demonstrates superior generalizability to the EU-Moths dataset compared to the model trained on NID-Moths, as shown in Fig. 4. This suggests that the diverse samples within the AMI-Traps dataset significantly contribute to enhanced cross-dataset performance.

For the NID-Moths dataset, the experimental results with YOLOv9-C mirror the trend observed in the EU-Moths dataset. However, the performance of the model pre-trained on NID-Moths is noticeably better than that on the AMI-Traps dataset, which placed second, as shown in Fig. 4. Several factors explain this gap. First, despite both datasets featuring insects on white panels, they originate from different geographical regions. AMI-Traps includes data from Northeast America, Western Europe, and Central America [15], while NID-Moths is merely from Germany [20]. Second, the ground-truth bounding boxes in the NID-Moths dataset are less tightly annotated than those in AMI-Traps, likely contributing to the relatively lower AP_{75} and $AP_{50:95}$ scores.

The experimental results for the AMI-Traps dataset are less satisfactory. This is likely due to the more complex scenarios present in AMI-Traps compared to the simpler settings of the EU-Moths and NID-Moths datasets, as shown in Fig. 4.

In conclusion, YOLOv9-C pre-trained on the AMI-Traps dataset shows better generalizability than that on the other two datasets. For our paste-in strategy (Section 4.3.4), the AMI-Traps dataset was used for robust insect detector pre-training.

4.3.3 Binary-Class Detection: Given the promising results of our YOLOv9 experiments, we expanded our training to include the binary-class annotation of the AMI-Traps dataset (*moth* vs. *other-insect*). We conducted training on YOLOv9-C at resolutions of 640 and 1280, and extended YOLOv9 (YOLOv9-E) at a resolution of 1280 to thoroughly evaluate performance across different model sizes and input scales. The experimental results are shown in Table 3.

Table 3 Experiment results with MS COCO metrics of different models on binary-class AMI-Traps dataset. With an input resolution of 1280x1280, YOLOv9-E gives the best overall performance among the experiments.

Model Configuration	mAP_{50}	mAP_{75}	$mAP_{50:95}$
YOLOv9-C-640	0.823	0.512	0.496
D-FINE-L-640	0.802	0.573	0.527
YOLOv9-C-1280	**0.848**	0.611	0.562
YOLOv9-E-1280	**0.848**	**0.617**	**0.565**

Without a doubt, YOLOv9-E-1280 shows the best overall detection performance. This pre-trained model was then used as the "teacher" model for generating pseudo-labels (Section 4.3.4). In addition, we trained D-FINE-L [26] with a resolution of 640. Under the same resolution, YOLOv9-C outperforms D-FINE-L based on the score mAP_{50}. For the other metrics, the localisation ability of D-FINE-L is better, with a difference of approximately 3% for the score $mAP_{50:95}$. However, we prioritise YOLOv9 over D-FINE, as we place more value on finding the insect correctly in general than on matching the insect edges tightly.

4.3.4 Pseudo-Labelling Paste-In: To leverage a large number of unlabeled images, we employed a pseudo-labelling strategy using our high-performing YOLOv9-E model with a resolution of 1280. Based on the analysis of precision confidence and recall confidence curves (see Fig. 5), we decided on a confidence threshold of 0.7 to extract pseudo-labelled patches. Under this confidence threshold, the precision and recall for the *moth* class are approximately 95% and 90%, respectively. This ensures both the quality and the quantity of the pseudo-labelled moth patches.

With a relatively high confidence threshold of 0.7 for moths, this process yielded more than 203,309 moth patches from the unlabeled dataset (see Fig. 6), significantly enriching our training data. We did not paste in *other-insect* pseudo-labelled patches because of the scarcity of *other-insect* instances. Another reason is that, with a relatively high confidence threshold like 0.9, the precision score decreased by approximately 10% (see the *other-insect* category of the precision-confidence curve from Fig. 5), meaning more uncertainty that the detector would have on the *other-insect* category, despite the high confidence score. Hence, we avoid another uncertainty factor in our strategy by only using patches with a *moth* annotation.

These extracted pseudo-labelled patches were then integrated into existing AMI-Traps training images through a paste-in procedure. This procedure carefully considered the IoU between the paste-in bounding boxes and existing training data bounding boxes to ensure realistic and effective augmentation. With a predefined number of candidate images and a maximum number of patches, several paste-in augmented images can be used for further training. To reduce the biases introduced by the sharp edges of the patches, we adopted a seamless cloning method from the OpenCV library to process the patches during the paste-in phase (see Fig. 2).

To reduce the training time, we fine-tuned our baseline models using these newly augmented paste-in images. It is important to note that the validation set remained consistent with the one used to

Table 4 Experiment results with MS COCO metrics by applying different fine-tuning strategies on YOLOv9-E-1280 pre-trained model on the AMI-Traps dataset. The model fine-tuned with paste-in smooth-edged moth patches shows the best overall performance, though it experiences a slight performance drop on the *moth* category.

Model Configuration	mAP_{50}	mAP_{75}	$\text{mAP}_{50:95}$
baseline	0.848	**0.617**	0.565
fine-tune	0.844	0.615	0.564
fine-tune-paste-in-patches	0.851	0.614	0.563
fine-tune-paste-in-patches-smooth	**0.852**	**0.617**	**0.567**
fine-tune-pseudo-labeled-full-images	0.793	0.533	0.502

Table 5 Experiment results with MS COCO metrics by applying different fine-tuning strategies on YOLOv9-E-1280 pre-trained model on the AMI-Traps dataset. The model fine-tuned with paste-in smooth-edged moth patches improves for *other-insect* class, though it experiences a slight performance drop on the *moth* category.

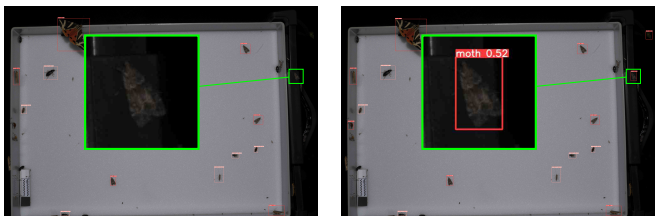
Model Configuration	AP_{50}	AP_{75}	$\text{AP}_{50:95}$
(Moth)			
baseline	**0.972**	**0.843**	0.731
fine-tune	**0.972**	0.842	**0.732**
fine-tune-paste-in-patches	0.971	0.838	0.724
fine-tune-paste-in-patches-smooth	0.971	0.838	0.727
fine-tune-pseudo-labeled-full-images	0.963	0.771	0.680
(Other-Insect)			
baseline	0.725	0.391	0.400
fine-tune	0.716	0.389	0.396
fine-tune-paste-in-patches	0.730	0.390	0.402
fine-tune-paste-in-patches-smooth	**0.734**	**0.396**	**0.406**
fine-tune-pseudo-labeled-full-images	0.624	0.296	0.324

train our baseline models, ensuring a fair comparison of performance improvements.

To rigorously evaluate the effectiveness of our paste-in strategy, we compare our fine-tuned model with the original baseline model of YOLOv9-E trained on the AMI-Traps dataset. The results are shown in Table 4 and 5. To verify whether the improvement is solely due to the addition of more pseudo-labelled patches rather than the effect of further fine-tuning, we fine-tuned YOLOv9-E with the same fine-tuning strategy without pasting in any of the pseudo-labelled patches. As the baseline model is already well-trained, further fine-tuning leads to overfitting and slightly degrades the overall performance. In addition, we fine-tuned YOLOv9-E on the fully pseudo-labelled images, which shows evident performance drops for both *moth* and *non-moth* classes compared to baseline results. This suggests that fine-tuning on a full image dataset without low-confidence pseudo-labels harms the model performance, as potential insects are incorrectly treated as background. Compared to the baseline result, the mean AP_{50} score for all classes improved by 0.4% after fine-tuning with paste-in moth patches with a smoothing strategy (Table 4), demonstrating that our method can further improve overall performance. However, the performance of *moth* detection in Table 5 degrades slightly while the performance of *other-insect* detection improves, compared to that of the baseline model, which is counterintuitive. This outcome is because more information from the *moth* category has led the model to learn more robust features, which inversely improves the overall generalizability and then leads to improvement over the *other-insect* category. From the prediction of the baseline and fine-tuned models on the image from the NID-Moths dataset (see Fig. 7), moths with bad lighting conditions can be well detected, and the confidence of the *other-insect* instances is relatively higher compared with the baseline model.

5 Conclusions and Future Work

In this work, we benchmarked SSD and YOLOv9 for one-class insect detection (one super-category *insect*) on three insect monitoring datasets, EU-Moths, NID-Moths and AMI-Traps, showcasing that YOLOv9 outperforms SSD with more robust and accurate



(a) Predicted bounding boxes from the **baseline** model.

(b) Predicted bounding boxes from the model **fine-tuned** with smooth-edged paste-in moth patches.

Fig. 7: Visualisation of predictions from YOLOv9-E-1280 models on an image from the NID-Moths dataset. The baseline model does not detect the moth in the dark region (**left**), while the model fine-tuned with smooth-edged paste-in moth patches can give a confident prediction and decent localisation (**right**).

detection results for complex scenarios. We carried out cross-dataset validation by training YOLOv9-C on these datasets and found that training on the AMI-Traps dataset, which contains a wide variety of insects, yields an insect detector that generalises better to other datasets when compared with other training datasets. As our research focuses on moths and YOLOv9 shows decent performance, we explored the performance of different model sizes and input image resolutions of YOLOv9, suggesting that high-resolution images are a key factor leading to better detection accuracy. Hence, we advise using high-resolution cameras for automated monitoring of nocturnal insects to yield the best performance from robust insect detectors. In addition, we introduced a paste-in method, demonstrated the feasibility of augmenting training data by pasting in high-quality pseudo-labelled moth patches, which were pseudo-labelled from the unlabeled camera trap NID-Moths dataset. This strategy further improved the detection performance of the YOLOv9-E pre-trained model with a high-resolution input size of 1280x1280, ensuring a seamless plugin during the training phase. Although the paste-in strategy led to only marginal gain in overall mAP scores, its primary value lies in enhancing model robustness.

However, our approach currently presents several limitations that require further exploration. First, the quality of the candidate pseudo-labelled patches is not perfect. Although Poisson blending reduces the hard-edge effect, it leads to the loss of features belonging to insects. Thus, advanced segmentation models can be explored, such as the Segment Anything Model (SAM) [18], to generate segmented insect patches. The recently proposed flatbug model for terrestrial arthropods might also be used for these segmentation purposes [29]. Another limitation is that the full potential of high-resolution camera trap images has not been exploited. Even with an input resolution of 1280x1280 for training images, fine-grained features of the insects can still be lost due to downscaling during processing. Future research could explore the SAHI framework (Slicing Aided Hyper Inference) [1], which has shown promise in maximising information from high-resolution imagery for small object detection, as demonstrated in studies utilising YOLOv8 with sliding window inference for terrestrial arthropods [29]. Third, relying solely on a fixed confidence threshold does not fully exploit the features of the insects in the unlabeled dataset. Thus, the model may overlook low-confidence insect patches that contain more complex features, which are crucial for enhancing the overall detection performance of the model. Future work could investigate more sophisticated semi-supervised learning paradigms, such as Soft-Teacher [37], which are designed to learn robustly from noisy and dynamic pseudo-labels, potentially leveraging both high- and low-confidence predictions. Finally, the counterintuitive performance trade-off observed between the *moth* and *other-insect* classes (see Table 4 and 5) requires further investigation to fully understand the dynamics of cross-category generalisation in this augmentation context.

6 Acknowledgement

The project on which this paper is based was funded by the German Federal Ministry of Research, Technology and Space (BMFTR) within the Research Initiative for the Conservation of Biodiversity (FEdA) under the funding code 16LW0653K. The responsibility for the content of this publication lies with the authors.

7 References

1 Fatih Cagatay Akyon, Sinan Onur Altinuc, and Alptekin Temizel. Slicing aided hyper inference and fine-tuning for small object detection. *2022 IEEE International Conference on Image Processing (ICIP)*, pages 966–970, 2022. doi: 10.1109/ICIP46576.2022.9897990.

2 Kim Bjerge, Jakob Bonde Nielsen, Martin Videbæk Sepstrup, et al. An automated light trap to monitor moths (lepidoptera) using computer vision-based tracking and deep learning. *Sensors*, 21:343, 2021. ISSN 1424-8220. doi: 10.3390/s21020343. URL https://www.mdpi.com/1424-8220/21/2/343.

3 Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. URL https://arxiv.org/abs/2004.10934.

4 G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

5 Panagiotis Christakakis, Garyfallia Papadopoulou, Georgios Mikos, et al. Smartphone-based citizen science tool for plant disease and insect pest detection using artificial intelligence. *Technologies*, 12:101, 2024. ISSN 2227-7080. doi: 10.3390/technologies12070101.

6 Rodolfo Dirzo, Hillary S. Young, Mauro Galetti, et al. Defaunation in the anthropocene. *Science*, 345(6195):401–406, 2014. doi: 10.1126/science.1251817. URL https://www.science.org/doi/abs/10.1126/science.1251817.

7 Mark Everingham, Luc Gool, Christopher K. Williams, et al. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88 (2):303–338, 2010. ISSN 0920-5691. doi: 10.1007/s11263-009-0275-4. URL https://doi.org/10.1007/s11263-009-0275-4.

8 Rebecca Gardiner, Stephanie Rowands, and Ben I Simmons. Towards scalable insect monitoring: Ultra-lightweight cnns as on-device triggers for insect camera traps. *arXiv preprint arXiv:2411.14467*, 2024. URL https://arxiv.org/abs/2411.14467.

9 Zahra Gharaee, ZeMing Gong, Nicholas Pellegrino, et al. A step towards worldwide biodiversity assessment: The BIOSCAN-1M insect dataset. In *Advances in Neural Information Processing Systems*, volume 36, pages 43593–43619. Curran Associates, Inc., 2023.

10 Golnaz Ghiasi, Yin Cui, Aravind Srinivas, et al. Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021. doi: 10.1109/cvpr46437.2021.00294. URL https://ieeexplore.ieee.org/document/9578639/.

11 Qingwen Guo, Chuntao Wang, Deqin Xiao, et al. Automatic monitoring of flying vegetable insect pests using an RGB camera and YOLO-SIP detector. *Precision Agriculture*, 24(2):436–457, 2023. ISSN 1573-1618. doi: 10.1007/s11119-022-09952-w. URL https://doi.org/10.1007/s11119-022-09952-w.

12 Ayesha Hakim, Amit Kumar Srivastava, Ali Hamza, et al. Yolo-pest: An optimized YoloV8x for detection of small insect pests using smart traps. *Scientific Reports*, 15(1):14029, 2025. ISSN 2045-2322. doi: 10.1038/s41598-025-97825-3. URL https://doi.org/10.1038/s41598-025-97825-3.

13 Caspar A. Hallmann, Martin Sorg, Eelke Jongejans, et al. More than 75 percent decline over 27 years in total flying insect biomass in protected areas. *PLOS ONE*, 12(10):e0185809, 2017. ISSN 1932-6203. doi: 10.1371/journal.pone.0185809. URL https://dx.plos.org/10.1371/journal.pone.0185809.

14 Aditya Jain, Fagner Cunha, Michael Bunsen, et al. A machine learning pipeline for automated insect monitoring. *arXiv preprint arXiv:2406.13031*, 2024. URL https://arxiv.org/abs/2406.13031.

15 Aditya Jain, Fagner Cunha, Michael James Bunsen, et al. Insect identification in the wild: The AMI dataset. In *Computer Vision – ECCV 2024*, pages 55–73. Springer Nature Switzerland, 2025. ISBN 978-3-031-72913-3.

16 Glenn Jocher. Ultralytics yolov5, 2020. URL https://github.com/ultralytics/yolov5.

17 Glenn Jocher and Jing Qiu. Ultralytics yolo11, 2024. URL https://github.com/ultralytics/ultralytics.

18 Alexander Kirillov, Eric Mintun, Nikhila Ravi, et al. Segment anything. *arXiv:2304.02643*, 2023. URL https://arxiv.org/abs/2304.02643.

19 Dimitri Korsch, Paul Bodesheim, and Joachim Denzler. Deep learning pipeline for automated visual moth monitoring: Insect localization and species classification. In *INFORMATIK 2021, Computer Science for Biodiversity Workshop (CS4Biodiversity)*, pages 443–460, 2021. doi: 10.18420/informatik2021-036.

20 Dimitri Korsch, Paul Bodesheim, Gunnar Brehm, et al. Automated visual monitoring of nocturnal insects with light-based camera traps. In *CVPR Workshop on Fine-grained Visual Classification (CVPR-WS)*, 2022.

21 Nithin Kumar, Nagarathna, and Francesco Flammini. YOLO-based light-weight deep learning models for insect detection system with field adaption. *Agriculture (Nitra, Slovakia)*, 13(741), 2023. ISSN 2077-0472. doi: 10.3390/agriculture13030741. URL https://www.mdpi.com/2077-0472/13/3/741.

22 Feng Li, Hao Zhang, Huaizhe Xu, et al. Mask DINO: Towards A Unified Transformer-based Framework for Object Detection and Segmentation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3041–3050. IEEE, 2023. doi: 10.1109/CVPR52729.2023.00297. URL https://ieeexplore.ieee.org/document/10204168/.

23 Tsung-Yi Lin, Michael Maire, Serge Belongie, et al. Microsoft COCO: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755. Springer International Publishing, 2014. ISBN 978-3-319-10602-1.

24 Wei Liu, Dragomir Anguelov, Dumitru Erhan, et al. SSD: Single shot MultiBox detector. In *Computer Vision – ECCV 2016*, pages 21–37. Springer International Publishing, 2016. ISBN 978-3-319-46448-0.

25 Hoang-Quan Nguyen, Thanh-Dat Truong, Xuan Bac Nguyen, et al. Insect-foundation: A foundation model and large-scale 1M dataset for visual insect understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21945–21955. IEEE Computer Society, 2024. doi: 10.1109/CVPR52733.2024.02072. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52733.2024.02072.

26 Yong Peng, Hao Li, Peng Wu, et al. D-fine: Redefine regression task in detrs as fine-grained distribution refinement. *arXiv preprint arXiv:2403.02324*, 2024. URL https://arxiv.org/abs/2410.13842.

27 N. Sabapathi. A unified deep learning framework for accurate pest detection and classification in agriculture. *Journal of Information Systems Engineering and Management*, 10:599–612, 04 2025. doi: 10.52783/jisem.v10i31s.5115.

28 Maximilian Sittinger, Johannes Uhler, Maximilian Pink, et al. Insect detect: An open-source DIY camera trap for automated insect monitoring. *PLOS ONE*, 19(4):e0295474, 2024. ISSN 1932-6203. doi: 10.1371/journal.pone.0295474. URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0295474.

29 Asger Svenning, Guillaume Mougeot, Jamie Alison, et al. A General Method for Detection and Segmentation of Terrestrial Arthropods in Images. *bioRxiv*, 2025, 2025.

30 Francisco Sánchez-Bayo and Kris A.G. Wyckhuys. Worldwide decline of the entomofauna: A review of its drivers. *Biological Conservation*, 232:8–27, 2019. ISSN 0006-3207. doi: 10.1016/j.biocon.2019.01.020. URL https://www.sciencedirect.com/science/article/pii/S0006320718313636.

31 Ana Cláudia Teixeira, José Ribeiro, Raul Morais, et al. A systematic review on automatic insect detection using deep learning. *Agriculture*, 13(3), 2023. ISSN 2077-0472. doi: 10.3390/agriculture13030713. URL https://www.mdpi.com/2077-0472/13/3/713.

32 Yunjie Tian, Qixiang Ye, and David Doermann. Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*, 2025.

33 Roel van Klink, Tom August, Yves Bas, et al. Emerging technologies revolutionise insect ecology and monitoring. *Trends in Ecology & Evolution*, 37, 07 2022. doi: 10.1016/j.tree.2022.06.001.

34 David L. Wagner, Eliza M. Grames, Matthew L. Forister, et al. Insect decline in the Anthropocene: Death by a thousand cuts. *Proceedings of the National Academy of Sciences*, 118(2):e2023989118, 2021. doi: 10.1073/pnas.2023989118.

35 Ao Wang, Hui Chen, Lihao Liu, et al. Yolov10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*, 2024. URL https://arxiv.org/abs/2405.14458.

36 Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, et al. Yolov9: Learning what you want to learn using programmable gradient information. *arXiv preprint arXiv:2402.13616*, 2024. URL https://arxiv.org/abs/2402.13616.

37 Mengde Xu, Zheng Zhang, Han Hu, et al. End-to-end semi-supervised object detection with soft teacher. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

38 Hao Zhang, Feng Li, Shilong Liu, et al. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. URL https://arxiv.org/abs/2203.03605.