

InsectDCT: A generalized pipeline for detection, taxonomic classification, and tracking of insects in camera-trap recordings

Kim Bjerger^{1*}, Simon F. A. Wogram^{2*}, Pau Enric Serra-Marin³, Otari Sakhiashvili⁴,
Toke T. Høye^{5,6},

1 Department of Electrical and Computer Engineering, Aarhus University, Aarhus C, Denmark

2 Community Ecology Department, Helmholtz Centre for Environmental Research, Halle, Germany

3 Global Change Research Group, Mediterranean Institute of Advanced Studies (IMEDEA, CSI-UIB), Esporles, Illes Balears, Spain

4 Caucasus Leibniz Biodiversity Research Center, Institute of Ecology, Ilia State University, Tbilisi, Georgia

5 Department of Ecoscience, Aarhus University, Aarhus C, Denmark

6 Arctic Research Center, Aarhus University, Aarhus C, Denmark

 These authors contributed equally to this work.

* kbe@ece.au.dk

Abstract

Automated monitoring of insect pollinators in natural environments with insect camera traps and trained deep learning algorithms provides novel data for insect ecological studies. However, efficient and accurate image recognition analysis of the recorded images or videos is challenging, particularly for images containing small insects against complex backgrounds with diverse vegetation communities. Even when insects can be detected in images, identifying their taxonomy remains difficult, particularly in footage with low image resolution, light conditions, and distances from the plants, and in cases where insects appear blurry or only partially visible.

In this work, we present *InsectDCT*, an AI-based pipeline for automated detection, hierarchical classification, and tracking of insects in footage of natural vegetation tested in different environments. The *InsectDCT* pipeline consists of three levels: insect Detection and localization, hierarchical taxonomic Classification, and spatio-temporal Tracking. In the first stage, insects are detected in time-lapse images or video recordings using the You Only Look Once (YOLO11) object detection architecture. Detection performance is improved using motion-enhanced images, which improve robustness in cluttered and 3 dimensional environments. The detector is trained on an extensive dataset that contains more than 60,000 images collected using camera traps deployed across a wide range of plant families and floral habitats. In the second stage, detected insects are classified using a hierarchical taxonomy-aware classification framework that covers 80 taxonomic groups. Classification is performed at multiple taxonomic levels, including order, family, and genus/species, allowing coarse and fine-grained ecological analyzes while accounting for varying levels of visual ambiguity. In the third stage, a multi-object tracking module is applied to high temporal-resolution image sequences and video data to associate detections of the same individual across time. *InsectDCT* code and all datasets are made publicly available.

Author summary

Insects are declining worldwide, creating an urgent need for efficient methods to monitor their abundance, activity, and diversity. Traditional insect surveys often require extensive fieldwork and expert taxonomic identification, which limits the scale and frequency of monitoring. In this study, we developed *InsectDCT*, an artificial intelligence-based pipeline that automatically detects, classifies, and tracks insects in camera-trap recordings collected from natural and semi-natural environments.

Our approach combines deep-learning methods for object detection, hierarchical taxonomic classification, and tracking of individual insect observations through time. Unlike many existing systems that are trained for a single habitat or plant species, we designed our framework using images collected across a wide range of flowering plants, camera systems, and insect groups. This makes the system more transferable to new ecological settings. The classifier can identify insects at multiple taxonomic levels and

can return higher-level classifications when species-level identification is uncertain.

We demonstrate that the pipeline can process large image datasets efficiently, including on low-power edge-computing devices such as Raspberry Pi systems. By providing both the software and the underlying datasets, we aim to support scalable, non-invasive insect monitoring and facilitate future ecological and conservation research.

Introduction

Insects are the most diverse animal group [1, 2], contributing to pollination, herbivory, decomposition, biological control, and food-web dynamics [3–5]. At the same time, mounting evidence indicates substantial declines in many insect taxa and regions, creating an urgent need for monitoring systems that can resolve changes in abundance, phenology, and community composition at ecologically significant spatial and temporal scales [4, 6]. Conventional approaches, including direct field observations, trapping, and expert-based taxonomic identification, remain indispensable, but are labor-intensive, spatially constrained, often destructive, and difficult to scale across many sites and seasons [7, 8]. These limitations restrict our ability to continuously monitor insect populations and link observed changes with land use, climate, habitat management, and insect-plant interactions [5, 7–9].

The ongoing global decline in insect populations has coincided with rapid advances in big data, machine learning, and low-cost portable technologies, collectively driving the development of automated monitoring pipelines for diurnal [10–12] and nocturnal [13–15] insects. Camera-based systems are particularly promising because they can record insects non-lethally, repeatedly, and under field conditions, generating highly taxonomically resolved information on abundance, activity patterns, phenology, and diversity of insects that would be difficult to obtain manually [9, 16–19]. Current approaches vary widely in their ecological context: some monitor the abundance of insects using artificial substrates such as colored panels or sticky surfaces [11, 15], while others detect insects against natural backgrounds, such as flowers, leaves, or fruits [20–24]. These systems have already accelerated the collection of data on ecological interactions, including pollination and herbivory [14, 25, 26].

Despite this progress, most existing pipelines remain tailored to specific study systems. Models are commonly trained on a limited set of camera types, plant backgrounds, recording distances, or target taxa, and their performance typically drops when applied to new geographic regions, plant assemblages, image characteristics, or insect communities [12, 22, 27]. This lack of transferability creates a practical bottleneck: Users who want to deploy automated monitoring in a new system often need to collect, annotate, and train on a local dataset before robust inference is possible. This repeated development of site-specific models slows adoption, duplicates effort between projects, and limits comparability among studies. Therefore, a more generalizable framework, trained on a diverse set of backgrounds, camera systems, and pollinator communities is needed to make automated insect monitoring more reusable in ecological contexts [17, 28, 29].

A second limitation concerns taxonomic resolution. For many questions in conservation, pollination ecology, and community ecology, detecting an “insect” or assigning a broad order is insufficient. Ecologically

meaningful inference often depends on the distinction of families, genera, or species, especially for focal
pollinator groups such as bees, hoverflies, and butterflies [9, 17, 25]. Fine-grained insect classification in field
imagery is difficult because organisms are often small, partially occluded, blurred, variably oriented, or
visually similar [10, 17, 27]. Moreover, training datasets typically have long-tailed class imbalance, with many
taxa represented by only a few images [27]. Hierarchical classification provides a biologically natural solution
by allowing predictions to remain useful at higher taxonomic ranks when species-level classification is
uncertain, but hierarchy-aware classification is still uncommon in operational insect-monitoring pipelines [27].

A third challenge is the detection of small objects in complex natural scenes. Camera-trap recognition has
long been affected by poor lighting, occlusion, camouflage, blur, and domain shifts between locations [28, 30].
Insect camera traps intensify these problems because target organisms may occupy only a few dozen pixels,
overlap with visually similar floral structures, or move out of focus in three dimensional vegetation [22, 31, 32].
Identification can be improved by placing cameras very close to flowers, restricting the field of view, or
accepting only relatively large detections [12, 32, 33]. Although these design choices can improve the accuracy
of the classification, they will reduce the taxonomic coverage of small insects [12, 32–34]. A restricted field of
view can also affect tracking, because insects are observed for shorter periods, leave the image frame more
quickly, and generate fewer consecutive detections from which individual trajectories and behavioral
information can be reconstructed [32, 35, 36]. A detector optimized for small insects is therefore essential for
monitoring smaller taxa and for recording wider scenes without losing sensitivity, while also supporting more
reliable tracking across longer image sequences.

Here we present *InsectDCT*, a generalized pipeline for insect detection, hierarchical taxonomic
classification, and tracking in camera recordings from natural and semi-natural floral environments. The
framework addresses three linked problems in automated insect monitoring: poor transferability across new
field conditions, limited fine-grained taxonomic resolution, and weak performance on small or distant insects.
The framework integrates motion-informed object detection, hierarchy-based classification with anomaly
handling, and track-by-detection to produce taxonomically annotated insect observations from time-lapse
and video recordings [27, 31, 36]. We trained models on various datasets that contain many floral
backgrounds, camera systems, and insect taxa. This design of a general-purpose framework aims to reduce
the need for each research group to train a bespoke algorithm from scratch, while still allowing future
extension to additional taxa from different geographical regions. In addition, we provide a large public
training and evaluation resource for insect detection and classification in natural camera-trap imagery. By
releasing both the code and the underlying datasets, *InsectDCT* can serve as a ready-to-use monitoring
pipeline and also as a reference and training resource for future algorithms. By combining heterogeneous

training data, small-object-oriented detection, hierarchical taxonomic classification, tracking, and open resources, *InsectDCT* provides a reusable computational foundation for scalable insect monitoring in ecological and conservation research.

Contribution

In this work, we propose a flexible and efficient processing pipeline for analyzing image recordings from insect camera traps in natural and semi-natural environments, enabling the detection, classification of insects at multiple taxonomic levels (including order, family, and genus/species) and tracking. A comprehensive dataset is presented, comprising images collected for training detection and classification models across a wide range of insects, flowering plants, and heterogeneity of backgrounds.

We evaluate the performance of the proposed pipeline in terms of precision, recall, and computational efficiency, and assess its processing speed on two edge computing devices: the Raspberry Pi 4 and Raspberry Pi 5. The proposed pipeline supports both time-lapse camera recording strategies and video-based tracking, enabling generic and robust monitoring of insects in natural environments.

In summary, our objectives for this work are the following:

- a generalized insect detection model trained on highly heterogeneous natural backgrounds.
- a hierarchical taxonomic classification model with anomaly-detection.
- to create an integrated tracking framework for insect monitoring in time-lapse recordings.
- public dataset for training and evaluating future models for insect detection and classification in camera recordings from natural environments.
- to evaluate different computing platforms, including edge devices, with respect to processing time.
- to connect the pipeline to insect ecology and conservation by demonstrating the proposed image processing pipeline in field-collected data.

Materials and methods

Processing pipeline

The *InsectDCT* tool first detects and localizes insects in the images, followed by taxonomic identification. The detection model is trained on a diverse set of images to improve robustness against false positives arising

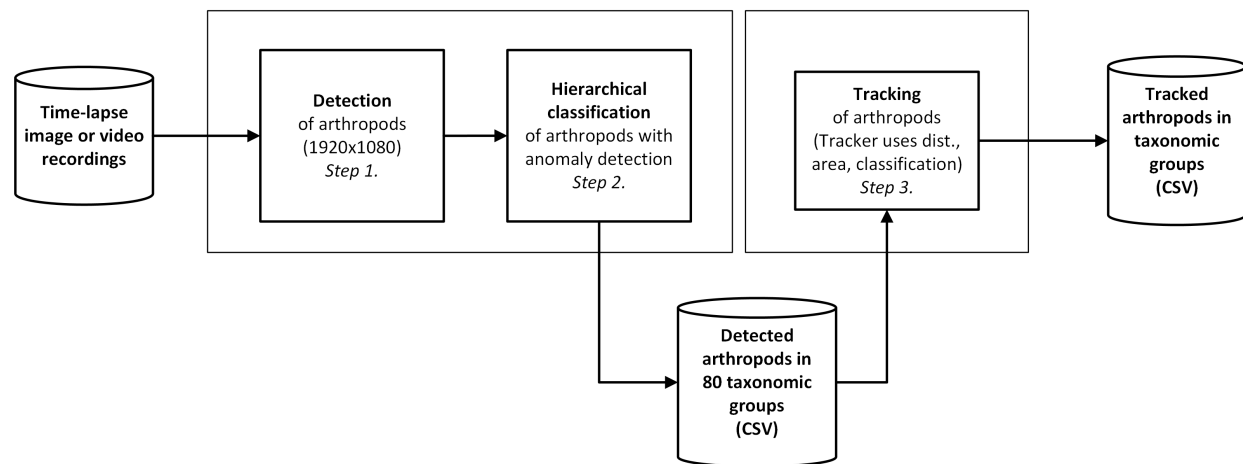


Fig 1. Processing pipeline to detect, classify and track insects from video or time-lapse recorded camera trap images. In step 1., the insects are detected and localized on resized images to 1920×1080 pixels using motion enhanced images and YOLO11. In step 2., insects are hierarchical classified into taxonomic ranks with 80 different taxonomic groups with anomaly detection of "unsure" insects. Localization and classification information of insect are used for the final tracking step 3. performed on time-lapse or video recorded with a sampling rate higher than 0.33 fps.

from small background objects that are visually similar to insects. A track-by-detection (TBD) approach was adopted due to its flexibility and because it does not require annotated tracking data. The pipeline can operate with or without tracking, depending on the selected time-lapse sampling interval. An overview of the proposed pipeline is illustrated in Figure 1 and elaborated below. The design prioritizes flexibility over computational efficiency and is based mainly on papers published by [27, 31, 36].

The first step in our pipeline is to detect and localize insects in each image. This is achieved with an object detection model based on YOLO11 [37], trained on a dataset (see Section "Data collection for insect detection") collected across diverse vegetation backgrounds and recorded with different cameras (see Tables 1 and 2). All images are resized to HD resolution (1920×1080 pixels), which reflects the typical resolution of the camera trap recordings used in the dataset. Medium and small versions of YOLO11 are trained on both standard color images and motion-enhanced images. Motion-enhanced images have previously been shown to improve recall [31] by enabling the detection of more small insects (area 50×50 pixels), including those that are partially visible or visually cluttered with the background.

The second step classifies the detected insects using a hierarchical classification model. The hierarchical classifier is trained on a dataset constructed from camera trap images collected in multiple locations and environments in Europe. The classifier architecture builds on the work of [27] and employs a multi-task learning framework combined with a dependency loss function that incorporates the taxonomic hierarchy. The hierarchical architecture consists of three taxonomic levels arranged in order, family, and genus/species.

However, some of the levels also contain higher taxonomic ranks of animals seen in the footage. For each detection, the classifier outputs a label and an associated confidence estimate at each of the three taxonomic levels. An anomaly detection mechanism is integrated into the classifier to identify insects with class scores that are outside the distribution of the training dataset across taxonomic groups. These outliers may correspond to false-positive background detections, partially visible or blurry insects, or representatives of previously unseen insect classes or other animals. The output of the insect detector and hierarchical classifier is a list of insect detections with additional information about the camera trap, image, bounding box coordinates, confidence, class labels (Levels 1-3), date, and time.

The third step is based on track-by-detection using the bounding box area, the distance between detections, and the class label to create insect tracks. The tracker is based on the work published in [36]. A final list of insect tracks with information about the predicted insect taxon, species, confidence, size, date, arrival time, and duration seen by the camera is generated.

The source code for the pipeline is available on Github: <https://github.com/kimbjerge/InsectDCT>.

Each step in the pipeline is described below, with a summary of the algorithms described in original papers [27, 31, 36].

Insect detection and localization

Deep learning-based object detection methods rely primarily on spatial image information to extract features and identify object regions within an image. You Only Look Once (YOLO) [38] is a one-stage object detector and is among the fastest available methods, making it well suited for processing millions of images or deployment on edge computing devices. In this work, YOLO11 [39], with CSPDarknet53 as the backbone, was used.

Monitoring insects in their natural environments can be achieved using time-lapse cameras, where images or video are recorded at fixed intervals ranging from 60 to 0.033 seconds. Small moving objects are often easier to detect when temporal information is incorporated into the training of DL models. To address this, motion-informed detection, as proposed by [31], is used in this study to improve the detection of insect objects by exploiting temporal image sequences.

The motion-informed detection approach estimates movement by computing differences between consecutive frames in a time-lapse sequence. These differences are used to generate motion-enhanced images, which are subsequently used for both training and inference of the DL object detector. By modifying the standard RGB image format with motion information, existing object detectors can be used without architectural changes. This approach therefore enables the reuse of widely adopted convolutional neural

network (CNN)–based object detectors such as YOLO [38].

Three consecutive images from the time-lapse recording were used to generate each motion-enhanced image. The color images were first converted to grayscale and blurred using a Gaussian kernel of 5×5 pixels (image size: 1920×1080 pixels). These grayscale, blurred images were then used to compute motion information to modify the color RGB images. Here, the red channel is replaced with a motion vector, whereas the blue channel is generated by combining the original red and blue channels with equal weighting (50% each). The result is blue/green images with red areas for where motion is detected. This approach makes it easier to find moving insects both visually and for the YOLO model. A detailed description of this process is provided in [31].

In this work, we trained four YOLO11 models (medium and small variants) using a comprehensive dataset consisting of both motion-enhanced images (MIE) and standard RGB images. This approach was adopted to develop a robust and generalizable model capable of detecting insects in various vegetation backgrounds and environmental conditions.

The performance of the insect detector and the localization model was evaluated in 5% of the training dataset, as well as in a separate test dataset containing images not used during training, including scenes with vegetation backgrounds previously unseen. The datasets for training, validation and testing are described in Section "Dataset for taxonomic classification".

Hierarchical taxon classifier with anomaly detection

The network architecture proposed by Bjerger et al. [27] is used and summarized in the following description. The architecture uses multitask learning by formulating taxonomy ranks as independent tasks to learn, which is implemented by modifying the output layer of a standard CNN. Specifically, the last layers are substituted with parallel fully connected (FC) layers for each taxonomy rank. Thus, a parameter sharing between all ranks while keeping rank-specific output layers. It is assumed that when more ranks are learned simultaneously, the more likely it is that the model will find a representation that captures all of the taxonomic ranks and dependencies.

The first part of the architecture is a CNN where we used ResNet [40], EfficientNetV2 [41] or ConvNext [42]. An input image X transformed by the network $N_{CNN}(\theta_{CNN})$, where θ_{CNN} means the trainable parameters of the CNN network is given. The network output is viewed as a root representation R_0 ,

$$R_0 = N_{CNN}(X, \theta_{CNN}) \quad (1)$$

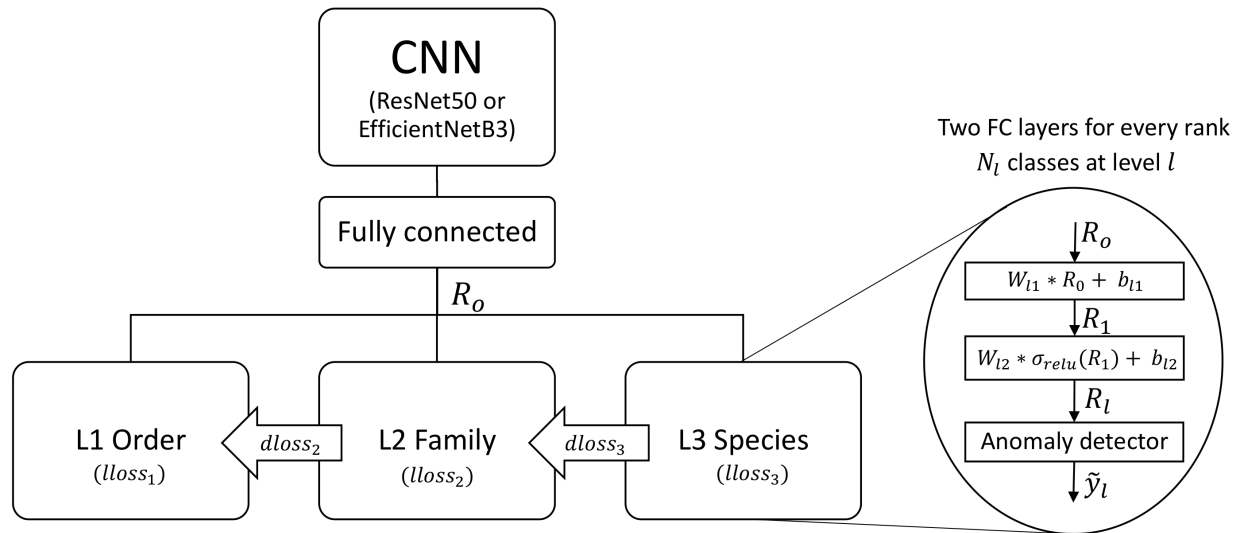


Fig 2. Network architecture with multitask learning to predict the three-level hierarchy. The multitask network shares the same backbone (ResNet50, EfficientNetV2 or ConvNext) with task branches at the end of parallel fully connected layers. A level (lloss) and dependency loss (dloss) function incorporates a level-dependent penalty to ensure hierarchy alignment. Figure from paper [27].

For each level l two FC layers and an activation function are added to perform a non-linear transformation of the input R_0 to the level output R_l .

$$R_l = (W_{l2} * \sigma_{relu}(W_{l1} * R_0 + b_{l1})) + b_{l2} \quad (2)$$

Here, W_{l1} and W_{l2} represent the weights and b_{l1}, b_{l2} represent the biases for the independent representation of the FC layer. Dropout regularization is applied in the forward pass during training so that random neurons W_{l1} are disabled in the FC branches. Each rank is handled by the parallel FC layers to learn independent multiple levels of the taxonomy. This independent task representation is hierarchy-free, and the level dependency is handled by a combined loss function. Two types of losses are used, by combining a balanced softmax loss for each level and a hierarchical dependency loss function [27]. The balanced softmax [43] loss function is applied to compensate for the long-tailed distribution of the imbalanced dataset.

The images were cropped and resized to 128×128 or 224×224 pixels, which matches the dimensions used for a traditional CNN classifier [40]. Training was performed using data augmentation, including image scaling, horizontal and vertical flip, and adding color jitter for brightness, contrast, and saturation. We selected a batch size of 64-256 for training our models, since it is faster to update and results in less noise than smaller batch sizes. The Adam optimizer with a fixed learning rate of $1.0 \cdot 10^{-4}$ was chosen based on previously published experiments [27]. We have trained three CNN models to classify insects according to

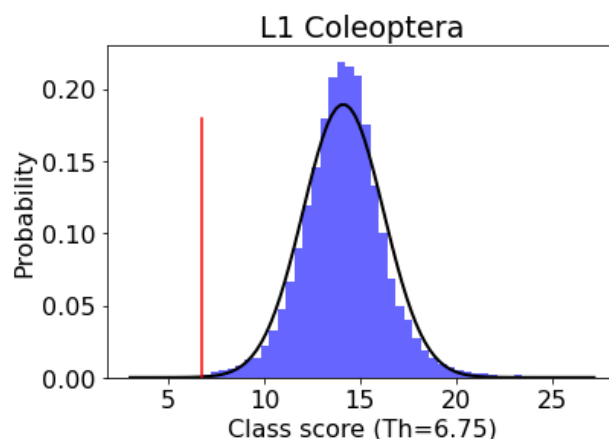


Fig 3. Probability density function for the output scores for Coleoptera and the chosen threshold for uncertain anomalies.

the hierarchical taxonomic groups defined by 80 classes described in Tables 3 and 4. CNN models were fine-tuned using pre-trained weights from ImageNet [44].

Anomaly detection The methodology of out-of-distribution detection [45] and threshold-based anomaly tagging is used to identify instances of "anomalies" such as uncertain classifications. In our application, these instances may manifest as debris, obscured or partially visible insects, or those exhibiting blurriness, characteristics that are not represented in the insect taxon training dataset.

Often, softmax is the last layer in a classification neural network, where the maximum value determines the predicted class. Here, we instead analyze the output distribution without the softmax layer to determine the anomalies and predict the classes. The distribution of the output x for each predicted class j^{th} follows a normal distribution $x_j \sim \mathcal{N}(\mu_j, \sigma_j^2)$.

An example of the output distribution is shown in Figure 3 which is generated on the sample training dataset (Coleoptera) for corrected classified input. If the output value x_j is below a threshold of $th = \mu - 3.0\sigma$, we label the input as anomaly. Consequently, when new unknown inputs are presented for the trained network and the output lies below the threshold, it will be classified as an "uncertain" prediction. The threshold is set to ensure that fewer than 1% of correctly classified inputs are marked "uncertain". However, thresholds between $\mu - 2.0\sigma$ and $\mu - 6.0\sigma$ can also be selected, depending on the desired strictness of the anomaly detector and the backbone classifier.

If a classification is correct at a higher taxonomic level, it is still considered valid and useful, even if it remains "unsure" at lower levels of the hierarchy. A classification is also labelled "unsure" if the predicted labels across the three levels are not taxonomically consistent.

This threshold-based filtering is applied to refine classifications to the highest reliable taxonomic

resolution, with pollinators identified at the genus or species level whenever sufficient confidence is achieved. 203
By incorporating anomaly detection, the classifier can assign specimens to higher taxonomic ranks when 204
predictions at lower levels are uncertain. For example, an individual classified as "unsure" at levels 2 or 3 205
could still be reliably identified as belonging to the order Hymenoptera at level 1. This hierarchical approach 206
preserves useful taxonomic information while reducing the risk of overconfident misclassification at finer 207
taxonomic resolutions. 208

Finally, the output scores x_j are assigned a probability $F(x_j)$ by estimating the cumulative distribution 209
function as the integral of the probability density function given by. This approach provides a statistical 210
measure of the predictions, in contrast to the softmax confidence score. 211

$$F(x_j|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x_j} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx \quad (3)$$

Tracking 212

The tracking algorithm proposed by Bjerger et al. [35] is used and described briefly below. The Hungarian 213
algorithm was selected to determine the optimal assignment within a given cost matrix, enabling the 214
association of detections across successive images in an efficient and globally optimal manner. In this 215
application, the cost matrix should represent the probability that an insect in the previous image had moved 216
to a given position in the current image. The cost function was defined as the weighted cost of distance and 217
area of the matching bounding boxes in the previous and current images. The Euclidean distance D between 218
the central position (x, y) in the two images was calculated as follows. 219

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (4)$$

This distance was normalized according to the diagonal of the image I : 220

$$D_{max} = \sqrt{(I_{height})^2 + (I_{width})^2} \quad (5)$$

The cost of the area was defined as the cost between the area A of the bounding boxes: 221

$$A_{cost} = \frac{A_{min}}{A_{max}} \quad (6)$$

A final cost function in Equation (7) was defined with a weighted distance cost W_d and a weighted area 222

cost W_a .

$$Cost = \frac{D}{D_{max}} W_d + (1 - A_{cost}) W_a \quad (7)$$

A cost threshold was established to determine whether successive insect detections should be associated. Subsequently, a track was established, stipulating a minimum of two detections per track. For each track, information such as the start date and time, duration, number of detections, and average size was recorded. Of particular importance was the recording of the predominant insect taxon, along with the accuracy of its classification. Ultimately, a track was considered valid when >50% of the detections corresponded to the predominant classification and comprised at least three detections. The tracker can be configured to incorporate classification results when forming tracks, ensuring that only classifications belonging to the same taxonomic group are associated. This optional extension is particularly beneficial when the classification results are sufficiently accurate.

Data collection and datasets

We collected time-lapse and video recordings from several monitoring sites using different cameras, including Logitech C922 web cameras (2MP), Wingscapes Time-lapse Pro Bird cameras (10MP), Raspberry Pi V3 cameras (2MP), and Raspberry Pi High Quality (HQ) cameras (HD video). Published studies by [21, 22, 25, 31, 35] include insect recordings from environments containing more than 24 different plant families and even more species, and a subset of these recordings was incorporated into our datasets. A balanced dataset was constructed by selecting 4,000-10,000 images from each source, representing various backgrounds and camera types, to train a generalized insect detector. In addition, four other projects (listed below) contributed recordings obtained with various cameras and plant types, including *Sedum*, mixed plants and flowers, herbs, and artificial flowers. For each project, between 10 and 30 cameras were used to collect images during the monitoring season. These additional projects, provide time-lapse recordings of a single flowering season (May-September).

- **Heather** - Wingscapes Pro camera recordings of heather plants (Denmark, 2019)
- **PollWatch** - Logitech C922 camera recordings of mixed plants and flowers (Denmark, 2020)
- **MAMBO** - Wingscapes Pro camera recordings of *Sedum* and herb plants (Denmark, France, Germany, Malta, Netherlands, United Kingdom, 2024)
- **Orchard** - Raspberry Pi V3 camera recordings of *Sedum* plants (Germany, 2025)

Two datasets were created: one for training a generalized insect detector model and another model for taxonomic classification of detected insects at the order, family, and species levels. The aim is to train one deep learning (DL) model to detect insects and a second DL model to classify the detected insects into taxonomic groups. The classification model is trained on the recordings described in this paper, but can be extended to include additional taxonomic groups with additional data provided. The two datasets for detection and taxonomic classification are described in the following and are available at: <https://doi.org/10.5281/zenodo.21154489>

Dataset for insect detection

With time-lapse images, a substantial proportion of these images may not contain any arthropods, resulting in false-positive detections of plant structures and other background elements [22]. Therefore, an insect detection dataset was constructed through an iterative process of progressively incorporating images that contain diverse insect species and vegetation backgrounds with false-positive detections. An initial dataset was formed by combining datasets published by Bjerger et al. [21, 22, 31], and a DL model was trained to

Table 1. A dataset of image samples was compiled from multiple sources to train insect detection and localization models across diverse vegetation backgrounds. The number of images, annotated arthropods, and empty background images without animals used for training and validation from each project is listed.

Project name (paper)	Images train/val.	Arthropods train/val.	Empty train/val.	Plants	Cameras
GreenRoof [21]	9,385/2,453	9,441/3,151	980/37	Sedum	Logitech
GreenHouse [31]	3,783/946	2,499/568	1,953/508	Clover, Sea rocket	Logitech
Arthropods [22]	6,901/525	4,770/766	2,654/299	Sedum	Wingscapes
PollWatch	2,867/720	468/118	2,488/625	Mixed plants	Logitech
MAMBO	4,313/1,163	5,725/1,635	806/206	Sedum, Herbs	Wingscapes
Orchard	3,157/798	2,705/696	703/173	Sedum, Artificial	Pi V3
ACSHQ [25]	4,123/1,036	3,360/745	1,001/200	>21 plant families	Pi HQ
Total	34,529/7,641	28,995/7,763	10,612/2,048	>30 plant families	4 cameras

Table 2. Test dataset of image samples was compiled from multiple sources to test insect detection and localization models across diverse vegetation backgrounds. The number of images, annotated arthropods, average pixel size and standard deviation (SD) used for testing from each project is listed.

Project name (paper)	Images	Arthropods	Avg size (SD)	Plants	Cameras
GreenHouse [31]	6,454	6,356	56 × 55 (24 × 21)	Clover, Rocket, Mallow	Logitech
Heather	1,166	946	79 × 74 (32 × 34)	White and pink heather	Wingscapes
MAMBO	2,349	3,185	64 × 61 (20 × 19)	Yellow Sedum	Wingscapes
Orchard	1,983	2,268	51 × 49 (16 × 17)	Sedum, Artificial	Pi V3
PollWatch	403	375	62 × 56 (25 × 21)	Mixed plants	Logitech
RT-Impact [35]	3,485	3,397	70 × 64 (23 × 21)	Yellow and red Sedum	Logitech
ACSHQ [25]	3,471	4,131	106 × 89 (33 × 31)	>21 plant families	Pi HQ
Total	19,311	20,658	70 × 64 (32 × 27)	>30 plant families	4 cameras

identify additional images containing arthropods from other projects with recorded data. Images containing insect detections were manually reviewed to remove false positives and identify missed detections by inspecting each image and correcting the corresponding bounding boxes using a labeling tool. This procedure was repeated for each project, with new DL models trained at every iteration as additional images were incorporated into the dataset. With each iteration, the number of false-positive detections was progressively reduced, improving the overall robustness of the detection model.

The final dataset, summarized in Table 1, comprises 42,170 images that contain arthropods captured against challenging backgrounds, including more than 30 different plant families. In total, the dataset includes 36,758 annotated arthropods and other animals identified in the images. Certain animals, such as slugs and snails, were excluded and therefore not annotated. The dataset consists of both color images and motion-enhanced images.

To evaluate the performance of the insect detector, a test dataset was created that includes two new projects (new locations and plants) and images from the same projects but from camera backgrounds not used in the training and validation dataset. The final test dataset, summarized in Table 2, comprises 19,311 images that contain insects captured against challenging backgrounds, including backgrounds with new plant families (different species of heater plants). Example images from the datasets are presented in the Supporting information.

Dataset for taxonomic classification

For the classification of detected insects, two datasets were created to train and evaluate hierarchical classification models. The datasets were organized into three taxonomic levels: order at level 1, family at level 2, and genus/species at level 3. The datasets were structured on the basis of the taxonomic groups of insects observed in the camera recordings. Figure 4 presents example images of selected pollinators grouped at the species level. However, many pollinator taxa could only be reliably classified or annotated at the genus level. This also explains why Level 3 combines genus- and species-level classifications rather than separating them into two distinct levels.

Most of the recordings, were collected from plots, dominated by Sedum plants, using images from projects that were also included in the detection datasets. The training dataset was compiled from recordings obtained in the projects: **GreenRoof** [21], **Arthropods** [22], **MAMBO**, and **Orchard**. These projects were selected because they contained a wide diversity of insect species, many of which had already been annotated in previous studies. The training data were further supplemented with GBIF images corresponding to the species identified in the camera recordings. This dataset was divided into 95% for

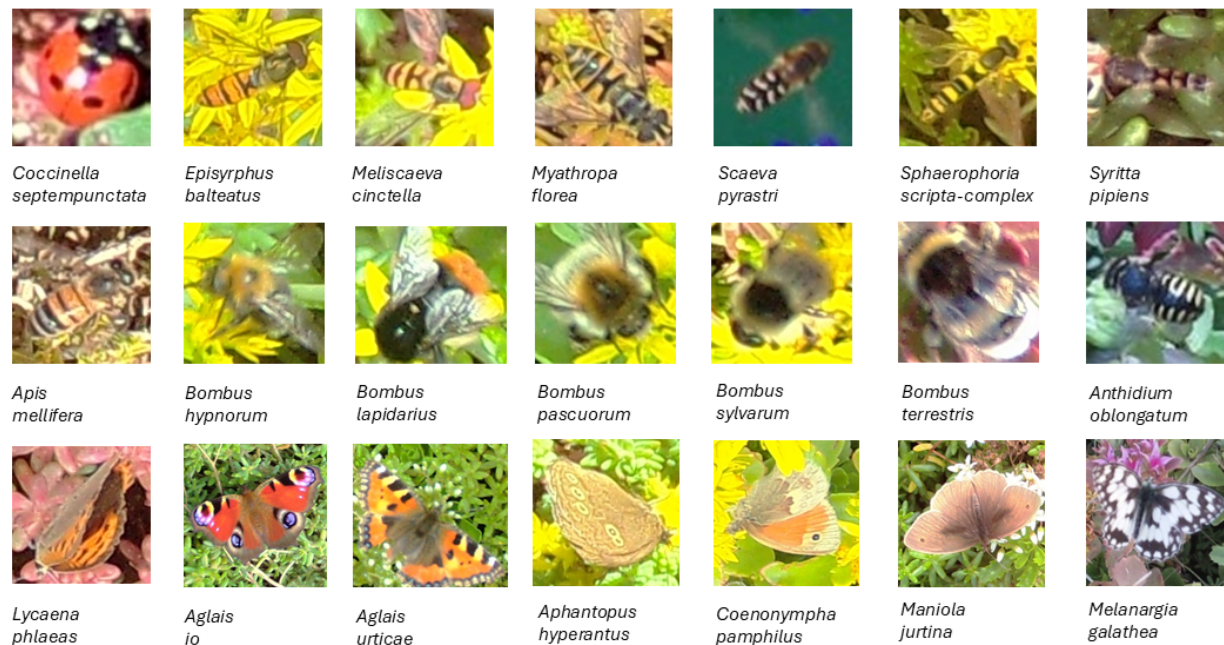


Fig 4. Shows example images of selected pollinators grouped at the species level, covering the families Coccinellidae, Syrphidae, Apidae, Megachilidae, Lycaenidae, and Nymphalidae.

training and 5% for validation.

The test dataset was created using recordings from some of the same projects but from different camera locations, including **RT-Impact** [35], **Arthropods** [22], **MAMBO** and **Orchard**. The purpose of the test dataset was to evaluate the models within the same insect domain while assessing generalization across different camera trap locations. The test dataset contains many of the same species as the training dataset; however, some species are not represented, representing a challenge to model generalization.

Tables 3 and 4 present the complete training and test datasets, comprising 19 taxa at level 1, 39 taxa at level 2, and 80 taxa at level 3. Some classes represent broader taxonomic categories, including animals and vegetation at higher taxonomic ranks. The classes were defined based on the organisms detected in the images, including false-positive detections of vegetation.

Special Apoidea groups were introduced at level 2 for bees that could not be reliably identified at the genus or species level due to limited image quality or insufficient visual detail. In several cases, taxa at level 3 were further divided into morphogroups such as Apoidea red_abdomen, small, and striped, where image size and quality were insufficient to reliably identify species-level characteristics. Only pollinator groups Syrphidae, Hymenoptera_bees, and Lepidoptera were consistently annotated at the genus or species levels, reflecting their primary relevance to the purpose of the classifier. Other groups were identified only at higher taxonomic levels and, therefore, assigned the same label across lower levels of the hierarchy. For example,

specimens classified as the order Hemiptera at level 1 are also labeled as Hemiptera at levels 2 and 3.

311

The average size of the insect images is 227×224 with a standard deviation of 235×234 pixels. However,

312

Table 3. Dataset (Part 1): Image samples used for training and testing a taxonomic classifier. Class names and the number of samples are listed for labels at hierarchical levels 1, 2, and 3 (L1, L2, and L3). The F1-score of the best-performing ConvNeXt-Base model is reported for each taxon at L3, evaluated on the test samples.

Taxa label L1	Taxa label L2	Taxa label L3	Train/val. samples	Test samples	F1-score (L3)
Araneae	Araneae	Araneae	2001	836	0.95
Birds	Birds	Birds	269	20	1.00
Coleoptera	Coleoptera	Coleoptera	2698	388	0.78
Coleoptera	Cantharidae	<i>Rhagonycha fulva</i>	1359	186	0.97
Coleoptera	Coccinellidae	<i>Coccinella septempunctata</i>	7270	1485	0.99
Dermaptera	Dermaptera	Dermaptera	439	85	0.88
Diptera	Diptera	Diptera	456	1124	0.92
Diptera	Bombyliidae	Bombyliidae	479	0	-
Diptera	Sarcophagidae	Sarcophagidae	811	298	0.98
Diptera	Syrphidae	Syrphidae	273	388	0.97
Diptera	Syrphidae	<i>Chrysotoxum</i>	317	0	-
Diptera	Syrphidae	<i>Episyrphus balteatus</i>	2732	1742	0.98
Diptera	Syrphidae	<i>Eristalis</i>	730	193	0.86
Diptera	Syrphidae	<i>Eupeodes</i>	4473	1110	0.97
Diptera	Syrphidae	<i>Eurimyia</i>	316	0	-
Diptera	Syrphidae	<i>Helophilus</i>	493	87	0.89
Diptera	Syrphidae	<i>Meliscaeva cinctella</i>	233	4	0.42
Diptera	Syrphidae	<i>Myathropa florea</i>	322	2	0.33
Diptera	Syrphidae	<i>Platycheirus</i>	406	176	0.76
Diptera	Syrphidae	<i>Scaeva</i>	278	18	0.77
Diptera	Syrphidae	<i>Scaeva pyrastris</i>	350	50	0.80
Diptera	Syrphidae	<i>Sphaerophoria scripta-complex</i>	832	884	0.97
Diptera	Syrphidae	<i>Syrpitta pipiens</i>	491	32	0.48
Diptera	Syrphidae	<i>Syrphus</i>	321	5	0.53
Diptera	Syrphidae	<i>Xanthogramma</i>	189	2	0.80
Hemiptera	Hemiptera	Hemiptera	1628	315	0.78
Herpetofauna	Herpetofauna	Herpetofauna	95	49	0.85
Hymenoptera_bees	Apidae	<i>Apis mellifera</i>	10499	1644	0.97
Hymenoptera_bees	Apidae	<i>Bombus</i>	435	80	0.90
Hymenoptera_bees	Apidae	<i>Bombus hypnorum</i>	208	1	0.09
Hymenoptera_bees	Apidae	<i>Bombus lapidarius</i>	2442	787	0.99
Hymenoptera_bees	Apidae	<i>Bombus pascuorum</i>	264	109	0.88
Hymenoptera_bees	Apidae	<i>Bombus sylvarum</i>	359	323	0.97
Hymenoptera_bees	Apidae	<i>Bombus terrestris</i>	4021	352	0.98
Hymenoptera_bees	Apidae	<i>Nomada</i>	419	1	0.22
Hymenoptera_bees	Apoidea	Apoidea	663	0	-
Hymenoptera_bees	Apoidea	Apoidea red_abdomen	280	8	0.36
Hymenoptera_bees	Apoidea	Apoidea reddish	350	50	0.91
Hymenoptera_bees	Apoidea	Apoidea small	1976	530	0.85
Hymenoptera_bees	Apoidea	Apoidea striped	1663	212	0.84
Hymenoptera_bees	Megachilidae	Megachilidae	2490	899	0.94
Hymenoptera_bees	Megachilidae	<i>Anthidium oblongatum</i>	382	48	0.82

as shown in Figure 5, the range of image sizes was wide, with GBIF images generally much larger than those captured by camera traps. Consequently, images recorded using insect camera traps contain less visual information, which can affect classification performance.

Table 4. Dataset (Part 2): Image samples used for training and testing a taxonomic classifier. Class names and the number of samples are listed for labels at hierarchical levels 1, 2, and 3 (L1, L2, and L3). The F1-score of the best-performing ConvNeXt-Base model is reported for each taxon at L3, evaluated on the test samples.

Taxa label L1	Taxa label L2	Taxa label L3	Train/val. samples	Test samples	F1-score (L3)
Hymenoptera_nobeas	Chrysididae	Chrysididae	792	0	-
Hymenoptera_nobeas	Crabronidae	Crabronidae	891	89	0.89
Hymenoptera_nobeas	Formicidae	Formicidae	968	1769	0.95
Hymenoptera_nobeas	Ichneumonidae	Ichneumonidae	666	12	0.27
Hymenoptera_nobeas	Pompilidae	Pompilidae	636	86	0.74
Hymenoptera_nobeas	Vespidae	Vespidae	607	46	0.84
Hymenoptera_nobeas	Vespidae	<i>Odynerus spinipes</i>	2531	166	0.96
Hymenoptera_nobeas	Vespidae	<i>Vespa vulgaris</i>	1568	256	0.97
Isopoda	Isopoda	Isopoda	880	132	0.94
Larvae	Larvae	Larvae	579	72	0.92
Lepidoptera	Lepidoptera	Lepidoptera	156	13	0.77
Lepidoptera	Hesperiidae	Hesperiidae	539	26	0.96
Lepidoptera	Lycaenidae	Lycaenidae	300	3	0.80
Lepidoptera	Lycaenidae	<i>Lycaena phlaeas</i>	277	2	1.00
Lepidoptera	Moths	Moths	218	167	0.95
Lepidoptera	Nymphalidae	Nymphalidae	31	0	-
Lepidoptera	Nymphalidae	<i>Aglaia io</i>	250	41	0.99
Lepidoptera	Nymphalidae	<i>Aglaia urticae</i>	1921	375	0.98
Lepidoptera	Nymphalidae	<i>Aphantopus hyperantus</i>	142	3	0.80
Lepidoptera	Nymphalidae	<i>Coenonympha pamphilus</i>	455	3	1.00
Lepidoptera	Nymphalidae	<i>Maniola jurtina</i>	645	23	0.92
Lepidoptera	Nymphalidae	<i>Melanargia galathea</i>	95	0	-
Lepidoptera	Pieridae	Pieridae	263	3	1.00
Lepidoptera_fw	Lepidoptera_fw	Lepidoptera_fw	271	33	0.62
Lepidoptera_fw	Nymphalidae_fw	<i>Aglaia urticae_fw</i>	352	13	0.13
Lepidoptera_fw	Nymphalidae_fw	<i>Maniola jurtina_fw</i>	3383	45	0.83
Lepidoptera_fw	Nymphalidae_fw	<i>Satyrinae_fw</i>	1288	14	0.31
Millipedes	Millipedes	Millipedes	177	38	0.97
Odonata	Odonata	Odonata	507	62	0.98
Orthoptera	Orthoptera	Orthoptera	241	6	0.86
Orthoptera	Acrididae	Acrididae	1738	108	0.88
Orthoptera	Acrididae	<i>Chorthippus</i>	982	70	0.98
Orthoptera	Acrididae	<i>Omocestus viridulus</i>	153	6	0.92
Orthoptera	Tettigoniidae	<i>Decticus verrucivorus</i>	192	0	-
Orthoptera	Tettigoniidae	<i>Pholidoptera griseoptera</i>	353	11	0.91
Slugs	Slugs	Slugs	1192	359	0.79
Snails	Snails	Snails	1687	811	0.90
Vegetation	Vegetation	Vegetation	5720	871	0.65
19 classes	39 classes	80 classes	90,358	20,257	0.76

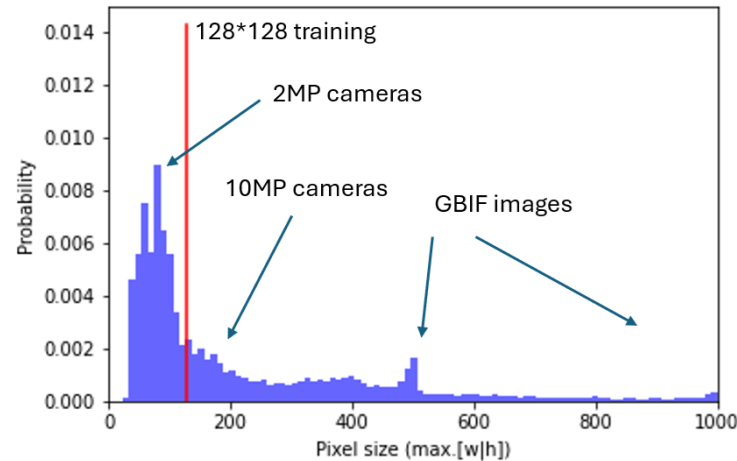


Fig 5. The distribution of insect image sizes in the training dataset. During training, all images are resized to 128×128 pixels. Insects captured in the 10 MP recordings and GBIF images are generally larger than those from the 2 MP recordings.

Results

Insect detection and localization

Precision, recall, F1-score, and mAP50 for the detection and localization stage of the pipeline are presented in Table 5 evaluated in the validation dataset. The larger YOLO11m model achieves the highest F1-score; however, the improvement is marginal considering the increased model size, with YOLO11m having 20.1M parameters compared to 9.4M for YOLO11s. Models trained on motion-enhanced images show improved F1-scores, with particularly higher values of mAP50 and mAP50-95. In general, the validation results in Table 5 indicate that there are no substantial differences between the models, as all perform well.

Table 6 presents the precision, recall, F1-score and mAP50 of the insect detector evaluated across seven different sources included in the test dataset, each comprising collected and annotated images with various backgrounds of plants and insects. The YOLO model trained with motion-enhanced images demonstrates a notable improvement in performance, achieving an average F1-score of 0.883 compared to 0.806 for the model trained on standard RGB images. In addition, the standard deviation across datasets is lower for the

Table 5. Validation results for insect detection and localization on 5% of the training dataset with 7,641 images and 7,763 labelled insects.

Model	Image	Parameters (M)	Precision	Recall	F1-score	mAP50	mAP50-95
YOLO11s	RGB	9.4	0.914	0.924	0.919	0.945	0.658
YOLO11m	RGB	20.1	0.909	0.933	0.921	0.946	0.654
YOLO11s	MIE	9.4	0.923	0.934	0.928	0.956	0.682
YOLO11m	MIE	20.1	0.923	0.935	0.929	0.951	0.711

model trained on motion-enhanced images, indicating a more consistent performance. The improvement is particularly evident in the recall, which increases from an average of 0.753 to 0.853. This indicates that the model trained on motion-enhanced images detects approximately 10% more insects compared to the standard model trained on RGB images. Compared to the validation results in Table 5, the performance decreases in the test dataset, as it contains images that were not used during training. However, the reduction in F1-score is only 4.6% for the model trained with motion-enhanced images, compared with 11.5% for the model trained on standard RGB images.

Hierarchical taxon classifier with anomaly detection

A total of 5% of the training dataset was reserved for validation and early stopping was applied after 45, 72, and 66 epochs, respectively. This relatively small percentage was chosen because the validation dataset was used solely for early stopping and not for the final model performance evaluation. The three models were then evaluated on the test dataset, with the results summarized in Table 7.

The F1-score increases at higher levels of the taxonomic hierarchy. The macro F1-score is higher than the micro F1-score across all models, reflecting the long-tailed and imbalanced distribution of both the training and test datasets. ConvNeXt-base achieves the best overall performance, followed by EfficientNetV2-S and ResNet50V2. However, EfficientNetV2-S has the lowest number of parameters, which makes it more suitable for applications where computational efficiency and inference speed are important.

Table 6. Test results for insect detection and localization using YOLO11m on the test dataset comprising 19,311 images and 20,658 labeled insects. The YOLO11m model is trained with color (RGB) images or motion-enhanced images (MIE). The table reports the average performance across the seven project datasets, along with the corresponding standard deviation (SD).

Project	Image	Precision	Recall	F1-score	mAP50	mAP50-95
GreenHouse	RGB	0.905	0.549	0.683	0.719	0.455
GreenHouse	MIE	0.902	0.747	0.817	0.886	0.565
Heather	RGB	0.930	0.801	0.860	0.888	0.674
Heather	MIE	0.896	0.901	0.898	0.953	0.788
MAMBO	RGB	0.812	0.699	0.751	0.810	0.670
MAMBO	MIE	0.908	0.879	0.893	0.955	0.846
Orchard	RGB	0.844	0.859	0.851	0.903	0.734
Orchard	MIE	0.890	0.891	0.890	0.949	0.791
PollWatch	RGB	0.853	0.665	0.748	0.794	0.621
PollWatch	MIE	0.899	0.795	0.844	0.902	0.723
RT-Impact	RGB	0.939	0.807	0.868	0.899	0.712
RT-Impact	MIE	0.961	0.926	0.943	0.979	0.822
ACSHQ	RGB	0.880	0.891	0.883	0.926	0.656
ACSHQ	MIE	0.924	0.868	0.895	0.947	0.790
Average (SD)	RGB	0.880 (0.043)	0.753 (0.112)	0.806 (0.072)	0.848 (0.070)	0.656 (0.090)
Average (SD)	MIE	0.911 (0.023)	0.858 (0.059)	0.883 (0.038)	0.939 (0.030)	0.761 (0.087)

The best-performing ConvNeXt-base model achieves average F1-scores of 0.90 at level 1 (L1), 0.84 at level 2 (L2), and 0.76 at level 3 (L3). Tables 3 and 4 show that the relatively low average F1-score of 0.76 is mainly driven by seven classes with very few test samples (1–13 samples per class). In contrast, the micro F1-score is substantially higher (0.92), indicating that the limited sample size in these classes introduces increased uncertainty in the overall evaluation.

The seven taxa with few test samples and correspondingly low F1-scores at level L3 were *Meliscaeva cinctella* (0.42), *Myathropa florea* (0.33), *Bombus hypnorum* (0.09), *Nomada* (0.22), Apoidea red abdomen (0.36), Ichneumonidae (0.27) and *Aglais urticae* fw (0.13). Taxa without dedicated test samples were evaluated using the 5% validation dataset and achieved high F1-scores, including Bombyliidae (0.98), *Chrysotoxum* (0.97), *Eurimyia* (0.94), Apoidea (0.74), Chrysididae (0.99), Nymphalidae (1.0), *Melanargia galathea* (1.0) and *Decticus verrucivorus* (0.9). However, these validation scores are likely optimistic, as the performance evaluated on the independent test datasets was generally lower.

The metrics reported in L2 and L3 allow for predictions within the same taxonomic hierarchy. For example, a prediction of *Bombus* is considered correct if the true label is *Bombus lapidarius*. This hierarchical correction is not applied in Figure 6, which illustrates how varying the threshold of anomaly detection affects performance. As more predictions are labeled as "unsure", the F1-score increases. A selected anomaly threshold value of 5.5 results in approximately 4%–8% of the predictions being labeled as "unsure". When a prediction is labeled as "unsure" in L2 or L3, the corresponding prediction is used instead at the next higher taxonomic level.

The confusion matrix for level 1, shown in Figure 7, indicates that the classes Lepidoptera and Lepidoptera_fw (folded wings) are frequently misclassified. In addition, 21% of slugs are predicted to be snails. Approximately 32% of vegetation samples are classified as "unsure"; however, this is acceptable, as the primary objective is to eliminate false-positive background detections including vegetation.

Computational speed and edge processing

The computational performance of the proposed pipeline was evaluated on several computing platforms, including edge-processing devices. These platforms included a standard workstation equipped with an Intel Xeon E5-2620 v4 @ 2.10 GHz CPU and an NVIDIA Quadro RTX 8000 GPU, a laptop-class system with an AMD Ryzen 7 PRO 6850U CPU (without GPU acceleration), and two edge devices: a Raspberry Pi 4 (RP4) with 4 GB RAM and a Raspberry Pi 5 (RP5) with 8 GB RAM.

Edge devices were selected due to their comparable cost and relevance for field deployment. The pipeline was evaluated by processing 35 images recorded with insect activity. Detailed runtime metrics are reported in

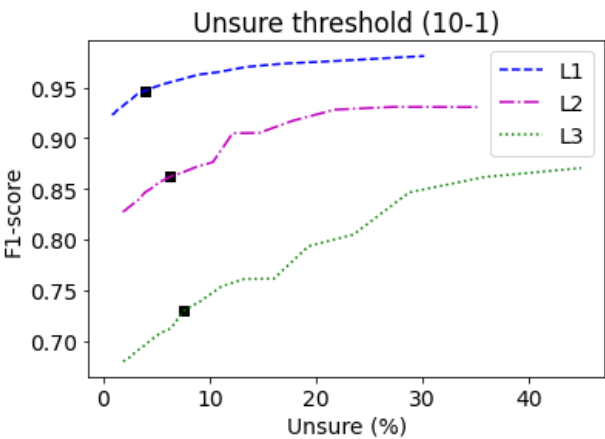


Fig 6. The F1-score at each taxonomic level for different unsure thresholds. As the uncertainty threshold increases, the proportion of classifications labelled as uncertain also increases, while the F1-score improves. The selected threshold value of 5.5 is indicated by a bold square using ConvNeXt-Base as the backbone.

Table 8. During testing, an external SSD containing the images was connected to the Raspberry Pi devices, which were accessed remotely over Wi-Fi without a connected display. The YOLO11s model was optimized using NCNN, a high-performance neural network inference framework, which resulted in improved performance only on the RP5 device.

Tracking in video recordings

Six days of recordings from different locations within the **Orchard** project were used to evaluate the entire pipeline, including tracking. The recordings were captured at a frame rate of 0.5 fps in HD resolution using Raspberry Pi 3 cameras. Each track was manually verified by inspecting the images contained, as illustrated in Figure 8. The tracks were manually categorized as correctly classified, containing multiple species, false

Table 7. Performance results for three hierarchical classifiers using ResNet50V2, EfficientNetV2-S, and ConvNeXt-base (CNB) backbones. Precision, recall, and F1-score are reported for each level (L1, L2, and L3) of the taxonomic hierarchy, evaluated on the test dataset. The backbone achieving the highest F1-score at each level is highlighted in bold. The number of model parameters and classes at each level is listed.

Model-Level	Parameters (M)	Classes	Precision	Recall	F1-score (macro)	F1-score (micro)
ResNet-L1	25.6	19	0.799	0.904	0.831	0.912
EfficientNet-L1	21.6	19	0.833	0.921	0.860	0.925
CNB-L1	88.5	19	0.897	0.926	0.903	0.948
ResNet-L2	25.6	39	0.768	0.880	0.783	0.907
EfficientNet-L2	21.6	39	0.790	0.894	0.804	0.921
CNB-L2	88.5	39	0.840	0.916	0.843	0.939
ResNet-L3	25.6	80	0.698	0.843	0.693	0.884
EfficientNet-L3	21.6	80	0.727	0.850	0.728	0.900
CNB-L3	88.5	80	0.758	0.889	0.759	0.922

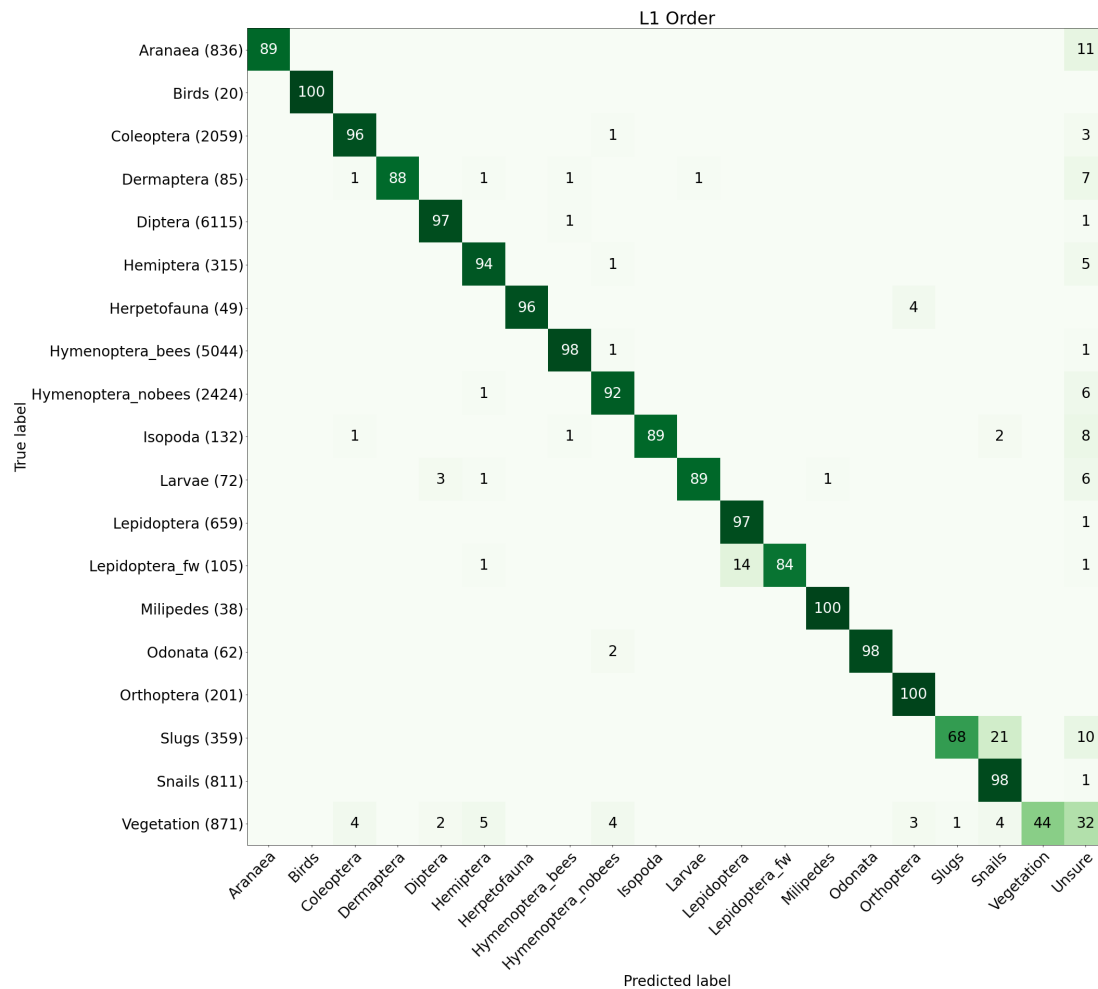


Fig 7. Confusion matrix for level 1 predictions on the test dataset using the ConvNeXt-base backbone. The diagonal values represent the percentage of true positive predictions for each class.

Table 8. Processing time performance for the *InsectDCT* algorithm executed on a workstation with GPU, a CPU-based laptop, and Raspberry Pi (RP) devices. All values are reported as the average processing time per image (in seconds), computed over a test set of 35 images containing 39 insect crops. The total pipeline processing time using YOLO11s with EfficientNetV2-S when processing an image containing a single insect is also listed.

Platform	YOLO11s	YOLO11s NCNN	YOLO11m	ResNet50V2	Efficient-NetV2-S	ConvNeXt-Base	Total time YOLO11s+EfficientNet
GPU	0.047	NA	0.083	0.0079	0.0175	0.0171	0.088
CPU	0.614	0.627	1.51	0.036	0.0614	0.0913	0.729
RP5	3.25	1.88	7.99	0.127	0.149	0.314	2.14
RP4	9.79	9.91	25.5	0.310	0.400	0.883	10.3

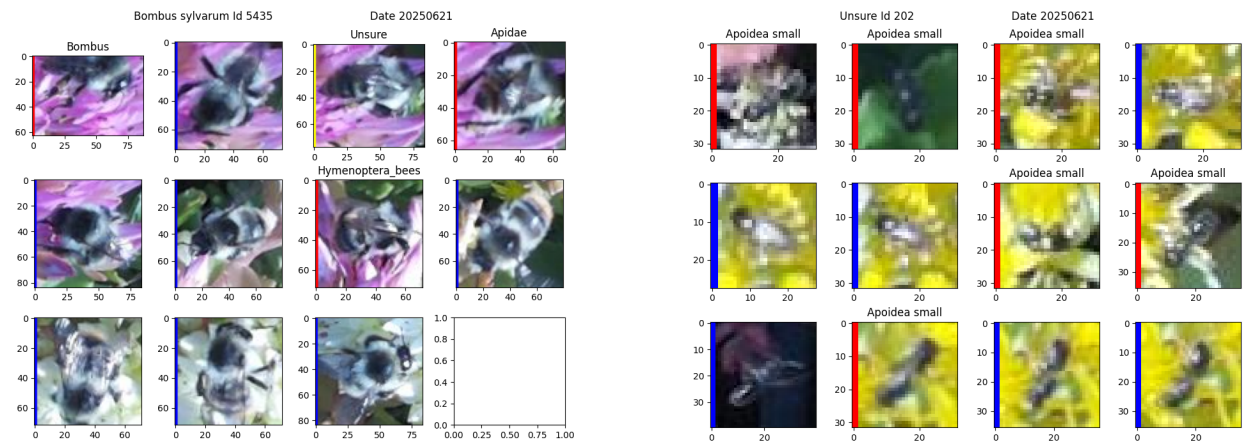


Fig 8. Shows examples of tracks with individual crops of detected and classified insects across image sequences. Left image crops contained in track of *Bombus sylvarum*. Right image crops contained in an "unsure" track with many crops classified as *Apoidea small*.

positives, or unsure due to small or blurry insects that were too difficult to reliably classify. Tracks containing multiple species were treated as biologically valid for the main accuracy analysis, as the target taxon was still present within the majority of the crops in the track.

In total, 4,753 tracks were inspected, of which 3,469 were correctly classified. An additional 141 tracks contained multiple species, resulting in a total of 3,610 valid tracks and an overall precision of 75.95%. False-positive and unsure tracks accounted for 363 and 780 tracks, respectively. If tracks categorized as unsure are also accepted as correctly classified, the precision increases to 90.86%. The final distribution of validated tracks for a subset of images in the **Orchard** dataset shown in Figure 9 indicates that small insects, particularly the *Apoidea small* and hymenoptera groups, account for a large proportion of unsure tracks. This is mainly due to the very small crop sizes, often around 30×30 pixels, as illustrated in Figure 8.

Monitoring insect abundance based on activity in camera recordings

The camera-derived activity data captured in the **Orchard** recordings show clear temporal patterns on both diel and seasonal scales. The daily activity curves differed between the pollinator groups (Figure 10): bees and butterflies showed predicted activity maxima around midday, peaking at approximately 13:20 h and 12:50 h, respectively, while hoverflies peaked earlier in the day at approximately 8:10 h. The estimated activity window above 50% of maximum activity was the broadest for hoverflies (12:20 h), followed by butterflies (9:00 h) and bees (7:10 h). Seasonal occurrence patterns also showed that the tracking data can resolve taxon-specific phenological windows throughout the sampling season (Figure 11).

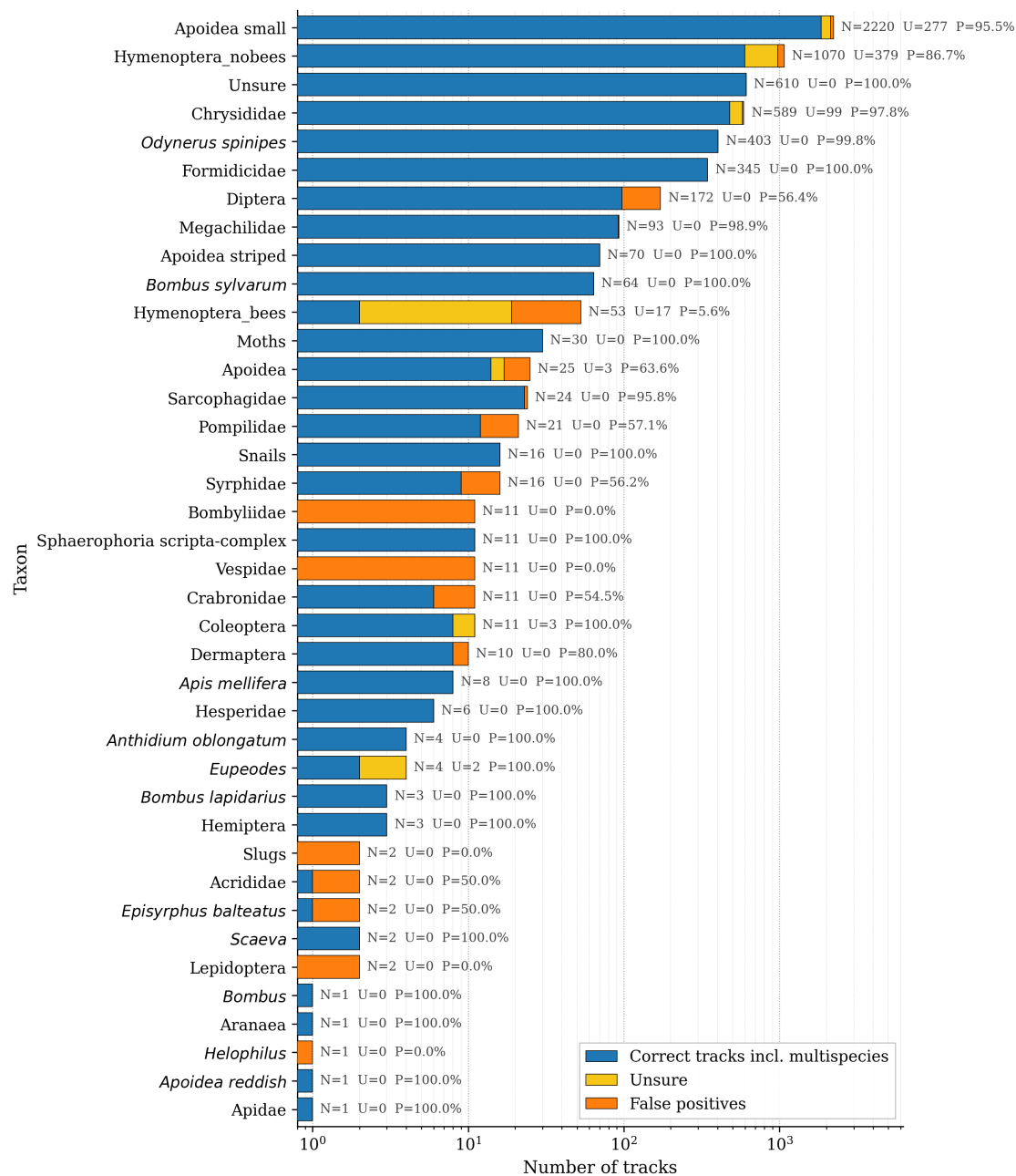


Fig 9. The number of tracks of insect taxa identified in recordings from six different camera locations from the *Orchard* dataset, each monitored over a one-day period. The correct tracks incl. multispecies are colored blue. Unsure tracks are shown in yellow and false positive tracks are shown in orange.

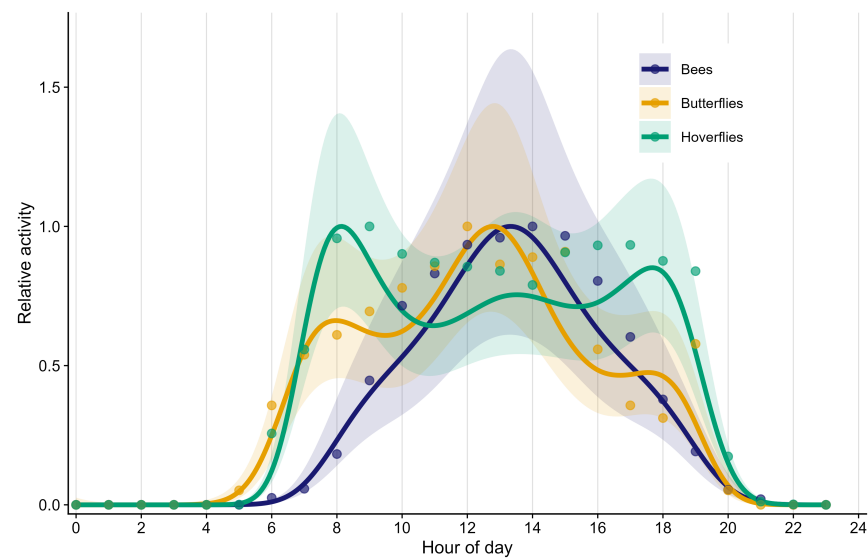


Fig 10. Daily activity patterns of butterflies, hoverflies, and bees in the all of the **Orchard** recordings. Smoothed curves show group-specific activity patterns across the day based on observation time. Points indicate hourly observation counts scaled to the maximum within each pollinator group. Shaded bands represent approximate 95% confidence intervals from negative-binomial GAMs. Activity is shown as relative activity to facilitate comparison of temporal patterns among groups rather than absolute abundance.

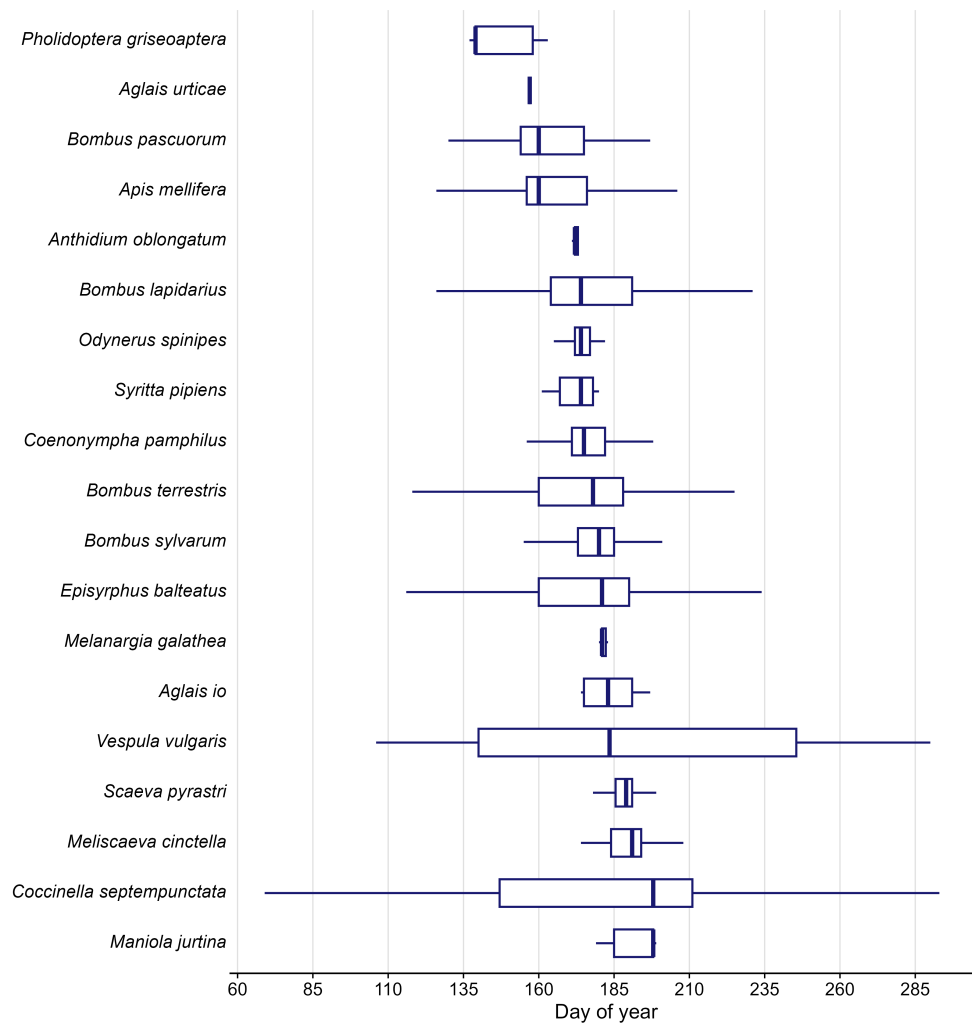


Fig 11. Phenology of observed insect species across the sampling season in the **Orchard** recordings. Horizontal boxplots show the distribution of observations for each species across day of year. Boxes indicate the interquartile range, whiskers show the observed temporal range excluding outliers according to the standard boxplot definition. Only taxa assigned to species level and represented by at least 20 observations are shown. Species are ordered by median day of observation.

Discussion

This study presents *InsectDCT* as a generalized deep learning pipeline for detecting, hierarchically classifying, and tracking insects and other small animals in camera-trap recordings from natural and semi-natural floral environments. The pipeline addresses three central limitations of current automated insect-monitoring systems: limited transferability across field conditions, restricted taxonomic resolution, and reduced sensitivity to small or distant insects in complex natural scenes [22, 27, 29, 31, 46]. By combining a detector trained on many different vegetation communities and camera systems, a hierarchy-aware classifier covering 80 taxonomic and functional groups, and track-by-detection algorithm for temporally linked observations, *InsectDCT* provides a reusable framework for converting large image streams into standardized ecological observations.

A central contribution of the pipeline is its broad applicability across diverse scenes and recording conditions. Rather than being trained for a single plant species, camera setup, or local insect community, *InsectDCT* was developed using image data from multiple projects, camera systems, flowering plants, and vegetation backgrounds. Our results show that this diversity translates into robust performance on independent test datasets including unseen recording contexts and vegetation backgrounds. Another major contribution of this work is the development and release of two complementary datasets: one for insect detection and localization using both RGB and motion-enhanced images, and one for hierarchical taxonomic classification across multiple arthropod groups. These datasets are broader in background structure, camera type, and taxonomic composition than many previous camera-trap studies, which were often constrained to specific plant species, camera setups, or target taxa [21, 22, 27, 31].

Beyond technical evaluation, field application demonstrates how *InsectDCT* can translate raw image streams into ecologically interpretable time series. The diel activity curves and phenological distributions extracted from the tracked detections are particularly relevant because pollinator activity varies throughout the day and season in response to taxon-specific thermal constraints, solar radiation, wind, and availability of floral resources [19, 47–50]. In this sense, the pipeline provides a non-lethal approach for quantifying relative activity, flower visitation, and phenology of common flower-visiting taxa at temporal resolutions that are difficult to achieve with repeated manual observations [32, 46, 51].

However, it should be taken into account that camera-derived counts should be interpreted as activity or visitation indices rather than direct estimates of population size unless additional assumptions are

met [46, 52–54]. Therefore, *InsectDCT* outputs are best treated as standardized, camera-specific activity-density or visitation-rate proxies unless they are calibrated against independent counts or analyzed with models that explicitly account for detection probability and sampling effort.

Svenning et al. [29] proposed a generic method for arthropod detection trained on a large dataset, allowing for scale and size-agnostic detection; however, the approach was developed using relatively homogeneous backgrounds. Previous studies [21, 22] have demonstrated insect detection in natural environments, but were limited to models trained primarily on the backgrounds of *Sedum* plants. The detector presented in this work builds on the approach in [31], extending it with a substantially greater diversity of vegetation backgrounds. This increased variability improves the robustness of the model and reduces the likelihood of false-positive detections in complex natural scenes. To our knowledge, *InsectDCT* is therefore among the most broadly trained and tested camera-based insect monitoring pipelines for natural vegetation backgrounds currently available, which makes it especially valuable for users who want to apply automated insect monitoring in new projects without first developing a fully project-specific detection algorithm from scratch.

The detector trained on motion-enhanced images achieved a higher average F1-score than the RGB-based detector and increased average recall, while also showing lower variation in F1-score among projects. The strongest improvements occurred in particularly challenging backgrounds, such as the **GreenHouse**, **MAMBO**, and **PollWatch** datasets, where recall increased substantially when motion-enhanced images were used. These results support the findings of Bjerger et al. [31], demonstrating that models trained with motion-enhanced images generalize better to new environments and domains.

Designing an appropriate camera setup for genus- or species-level insect identification involves a trade-off between frame rate, camera resolution, field of view, and the resulting size of insect crops. A larger field of view increases the likelihood of capturing insects within the scene and allows for monitoring larger floral patches, but it also reduces the pixel size of individual insects and can therefore lower classification reliability unless higher-resolution cameras are used. In contrast, close-range imaging can improve detection and classification by producing larger insect crops, but at the cost of a smaller sampling area. This trade-off is illustrated by previous camera-based monitoring systems. Zsofia et al. [34] detected and tracked bumblebees visiting flowers, but the relatively large distance between cameras and flowers made it difficult to obtain a reliable classification for diverse insect communities. Smith et al. [12] presented AutoPollS, which uses four Borescope USB cameras placed close to the plants and only accepted detections larger than 120×120 pixels. This design increases the robustness of detection and classification, but the resulting field of view is considerably narrower than in the datasets presented here.

In contrast, the detector in *InsectDCT* was trained specifically for small insects, with an average object

size of 71×66 pixels (standard deviation 36×33) in HD-resolution images. Robust detection at this scale allows *InsectDCT* maintain a broader field of view while being able to detect and classify smaller pollinators that would otherwise be missed or excluded. At the same time, this specialization introduces a scale-related trade-off: performance decreases for larger insects, which were less well represented in the small-object-focused training data. To address this, a separate version of the color-based detector was trained on additional images that contain larger specimens. However, this model performed less effectively on small insects, highlighting the difficulty of developing a single detector that performs equally well across the full range of insect body sizes and image scales, even though YOLO architectures are designed for multi-scale object detection.

A key strength of the *InsectDCT* classifier is its hierarchical structure, which makes the output useful even when fine-grained identification is uncertain. Instead of forcing every detection into a species-level label, the model can retain ecologically informative predictions at higher taxonomic levels, such as order or family, while providing genus- or species-level classifications where the image quality and taxonomic separability allow it. In the current implementation, the classifier already covers a broad range of arthropod groups, including the main flower-visiting taxa observed in the recordings, with bees, hoverflies, and butterflies represented at a relatively fine taxonomic resolution. Nevertheless, the dataset used to train the hierarchical classifier exhibits a long-tailed class imbalance. To compensate for this imbalance during training, a balanced softmax loss function was used. However, the method only partially mitigates the imbalance and classes with very few training samples remain challenging for the model to classify accurately. The hierarchical classifier could be further refined by resolving broad groups such as Coleoptera and Hemiptera into lower taxonomic ranks. Currently, these categories contain a high diversity of visually distinct taxa, which reduces the classification accuracy.

We recommend extending the dataset with additional taxonomic groups when applying the pipeline to environments containing insect species not represented in the current recordings. Even if species-level predictions are likely unreliable for previously unseen taxa or regions, the pipeline can still assign detections to broader taxonomic groups and generate standardized image crops that are already partially sorted. These outputs can then be reviewed and re-annotated more efficiently, providing a structured starting point for extending the classifier with additional taxa rather than requiring researchers to build a training dataset from unfiltered raw images. In this sense, *InsectDCT* can support both immediate ecological inference at coarser taxonomic levels and the iterative development of more region- or project-specific classifiers [27].

Tracking is a central part of *InsectDCT* since it converts frame-level detections into standardized activity and visitation indices. By reducing the repeated counting of the same individual within short image

sequences and generating track-level summaries such as visit duration, arrival time, and predominant taxonomic identity, the pipeline provides outputs that are directly relevant for estimating relative activity, visitation rates, diel patterns, and pollinator behavior.

Our evaluation of the complete tracking pipeline was conducted on a highly challenging dataset with an average insect image size of only 50×50 pixels, making it difficult to reliably identify many small solitary bees. The tracking algorithm also struggles when small insects cross paths, which can result in mixed or fragmented tracks. However, for video recordings, we have also experimented with YOLO11 ByteTrack [55] or BoT-SORT [56] tracking algorithms in the initial processing stage. In this approach, insect detection and tracking are combined in a single step, followed subsequently by taxonomic classification. However, preliminary testing on eight video sequences indicated inferior performance compared to our approach (results not shown). In particular, occlusions and missed detections often caused a single insect to be represented by multiple fragmented tracks, reducing overall tracking accuracy.

For custom insect camera systems based on Raspberry Pi 4 or 5 platforms [11, 32, 33], our pipeline is capable of processing time-lapse recordings in near real time for image recording intervals down to 2 seconds. The complete pipeline, including tracking, can therefore operate directly on a Raspberry Pi device and could be extended to store only images containing detected insects. The confidence threshold may also be reduced to increase recall, allowing the detection of more than 90% of insects present in the recorded footage 6.

As an alternative to deploying the complete pipeline on a camera trap equipped with a Raspberry Pi, and to address the computational constraints of edge deployment, we recommend a two-stage processing strategy as proposed by Smith et al. [12]. In the first stage, insect detection using the YOLO11s NCNN model is performed in real time on the camera device. This can be achieved on a Raspberry Pi 5 for time-lapse intervals as short as 2 seconds, or on a Raspberry Pi 4 for intervals of 10–60 seconds. Only images containing detected insects are stored locally and subsequently processed offline using the full *InsectDCT* pipeline for taxonomic classification and tracking. This approach exploits the relatively low computational cost of optimized YOLO-based detection on edge hardware while offloading the classification and tracking tasks to systems with greater flexibility and processing capacity. Consequently, power consumption, storage requirements, and data transmission needs in the field can be substantially reduced without compromising taxonomic resolution or tracking performance.

Users wishing to deploy *InsectDCT* in a new ecological context, should follow a structured workflow that spans image acquisition, annotation, model training, and validation. In the field acquisition stage, cameras should be positioned at a distance of 30 cm with a wide-angle lens and 40–50 cm with a 35mm focal lens (Pi HQ Camera) of the floral patch and any angle if the insect is within a detectable size range; based on the

performance analysis presented in this study, a minimum crop size of approximately 50×50 pixels in HD resolution is advisable for reliable classification. Time-lapse intervals of 60 seconds or less are required for the motion-enhanced image pipeline. For video-based deployments, frame rates above 0.33 fps are required to maintain high performance of the tracking of individuals.

Images should be collected throughout the entire phenological season, under varying illumination conditions and at multiple locations, to ensure that the resulting data set reflects the natural variability of the target environment. For detection, the existing trained YOLO11 model can be applied directly in a first pass; images flagged as containing insects should be manually verified to remove false positives and recover missed detections. This reviewed set should ideally be incrementally added to the training corpus for iterative re-training following the procedure described in Section "Dataset for insect detection".

For taxonomic classification, new taxa can be incorporated by supplementing the data described in Tables 3 and 4 with additional labels L1, L2 and L3, provided that a minimum of approximately 100–300 annotated crops per class are available; classes with fewer samples can be supplemented with images recorded with smartphones (e.g., GBIF images), keeping in mind the domain gap between such images and camera-trap crops discussed above. Finally, tracking parameters, particularly the cost threshold and the minimum number of detections per valid track, may require adjustment depending on the frame rate and typical insect movement patterns in the new recording setup. *InsectDCT* architecture, in which detection, classification, and tracking operate as independent stages with standardized inputs and outputs, is specifically designed to facilitate these adaptations, allowing researchers to replace or retrain individual components without modifying the rest of the pipeline.

The pipeline has the potential to support the extraction of ecological and behavioral foraging data for insects, allowing studies of e.g. insect flower visitation rates, flower handling times, and movement patterns among inflorescences that would be difficult or labor-intensive to obtain through manual observation alone. By linking consecutive detections into tracks, *InsectDCT* provides a temporal dimension to camera-trap data that goes beyond simple presence/absence or abundance estimates. This information is relevant for studies of plant-pollinator interaction networks, where species identity, visitation frequency, and movement patterns could be used to automatically infer the functional contribution of each pollinator species.

Despite its strong performance under diverse field conditions, *InsectDCT* has several limitations that should be acknowledged. The taxonomic classifier is currently trained on Northern and Central European pollinator taxa, with a large part of the dataset focused on vegetation dominated by plant species of the *Sedum* genus, which implies that the pipeline will require re-training for direct transferability to other geographic regions or habitat and vegetation types. Researchers deploying the pipeline in new contexts are

encouraged to supplement the existing dataset with locally collected and annotated images, following the incremental re-training workflow described above, which has been specifically designed to accommodate this kind of extension. The classes with few training samples, such as rare solitary bees and some folded-wing Lepidoptera, show reduced classification accuracy, and this imbalance could be mitigated in future versions by re-training the algorithm by users who require improved performance in those insect groups. Regarding the tracking algorithm, it can produce fragmented or merged trajectories when small insects cross paths or are temporarily occluded, and tracks should therefore be interpreted as activity proxies rather than records of unique individuals over extended periods.

Taken together, these limitations do not reduce the broader utility of this pipeline, but instead define clear pathways for extension. Because *InsectDCT* combines open datasets, modular detection and classification models, anomaly handling, and track-level output, it can serve both as an immediately usable monitoring tool and as a foundation for project-specific adaptation. By lowering the amount of annotation and algorithm development required before camera-based insect monitoring can be applied in new ecological contexts, *InsectDCT* can help make high-resolution, non-invasive pollinator monitoring more scalable, comparable, and accessible across studies.

Acknowledgments

This research was funded by the European Union's Horizon Europe Research and Innovation programme, under Grant Agreement No. 101060639 (**MAMBO**). The authors would like to thank all participants in the **MAMBO** project for their valuable contributions to the collection of image recordings throughout Europe. We also thank Hjalte Ro Poulsen from the Department of Geosciences and Natural Resource Management, University of Copenhagen, for providing images from the **Heather** project.

References

1. Zhang ZQ. Animal biodiversity: An introduction to higher-level classification and taxonomic richness. *Zootaxa*. 2011;3148(1):7-12. doi:10.11646/zootaxa.3148.1.3.
2. Stork NE. How Many Species of Insects and Other Terrestrial Arthropods Are There on Earth? *Annual Review of Entomology*. 2018;63:31-45. doi:10.1146/annurev-ento-020117-043348.
3. Losey JE, Vaughan M. The Economic Value of Ecological Services Provided by Insects. *BioScience*. 2006;56(4):311-23. doi:10.1641/0006-3568(2006)56[311:TEVOES]2.0.CO;2.

4. Potts SG, Biesmeijer JC, Kremen C, Neumann P, Schweiger O, Kunin WE. Global pollinator declines: Trends, impacts and drivers; 2010. doi:10.1016/j.tree.2010.01.007.
5. Cardoso P, Barton PS, Birkhofer K, Chichorro F, Deacon C, Fartmann T, et al. Scientists' warning to humanity on insect extinctions. *Biological Conservation*. 2020;242:108426. doi:10.1016/j.biocon.2020.108426.
6. Wagner DL, Grames EM, Forister ML, Berenbaum MR, Stopak D. Insect decline in the Anthropocene: Death by a thousand cuts. *Proceedings of the National Academy of Sciences of the United States of America*. 2021;118(2). doi:10.1073/PNAS.2023989118.
7. Montgomery GA, Belitz MW, Guralnick RP, Tingley MW. Standards and Best Practices for Monitoring and Benchmarking Insects. *Frontiers in Ecology and Evolution*. 2021;8. doi:10.3389/fevo.2020.579193.
8. Didham RK, Basset Y, Collins CM, Leather SR, Littlewood NA, Menz MHM, et al. Interpreting insect declines: seven challenges and a way forward. *Insect Conservation and Diversity*. 2020;13(2). doi:10.1111/icad.12408.
9. van Klink R, August T, Bas Y, Bodesheim P, Bonn A, Fossøy F, et al. Emerging technologies revolutionise insect ecology and monitoring. *Trends in Ecology and Evolution*. 2022;37(10):872-85. doi:10.1016/j.tree.2022.06.001.
10. Jain A, Cunha F, Bunsen MJ, Cañas JS, Pasi L, Pinoy N, et al. Insect Identification in the Wild: The AMI Dataset. In: Leonardis A, Ricci E, Roth S, Russakovsky O, Sattler T, Varol G, editors. *Computer Vision – ECCV 2024*. Cham: Springer Nature Switzerland; 2025. p. 55-73.
11. Sittinger M, Uhler J, Pink M, Herz A. Insect detect: An open-source DIY camera trap for automated insect monitoring. *PLoS ONE*. 2024;19(4 April):1-28. Available from: <http://dx.doi.org/10.1371/journal.pone.0295474>. doi:10.1371/journal.pone.0295474.
12. Smith MAY, Bernauer OM, Melone G, Graham B, Wiessing J, Salas R, et al. AutoPollS: A tool for automated monitoring of pollinators using deep learning. *Methods in Ecology and Evolution*. 2026;00(1-11). Available from: <https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/2041-210x.70304>. arXiv:<https://besjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/2041-210x.70304>. doi:<https://doi.org/10.1111/2041-210x.70304>.

13. Bjerger K, Nielsen JB, Sepstrup MV, Helsing-Nielsen F, Høye TT. An automated light trap to monitor moths (Lepidoptera) using computer vision-based tracking and deep learning. *Sensors (Switzerland)*. 2021;21(2). doi:10.3390/s21020343.
14. Alison J, Alexander JM, Zeugin ND, Dupont YL, Iseli E, Mann HMR, et al. Moths complement bumblebee pollination of red clover: A case for day-And-night insect surveillance. *Biology Letters*. 2022;18. doi:10.1098/rsbl.2022.0187.
15. Geissmann Q, Abram PK, Wu D, Haney CH, Carrillo J. Sticky Pi is a high-frequency smart trap that enables the study of insect circadian activity under natural conditions. *PLoS Biology*. 2022;20(7):1-26. doi:10.1371/journal.pbio.3001689.
16. Barlow SE, O'Neill MA. Technological advances in field studies of pollinator ecology and the future of e-ecology. *Current Opinion in Insect Science*. 2020;38. doi:10.1016/j.cois.2020.01.008.
17. Høye TT, Årje J, Bjerger K, Hansen OLP, Iosifidis A, Leese F, et al. Deep learning and computer vision will transform entomology. *Proceedings of the National Academy of Sciences*. 2021;118(2). doi:10.1073/pnas.2002545117.
18. Besson M, Alison J, Bjerger K, Gorochowski TE, Høye TT, Jucker T, et al. Towards the fully automated monitoring of ecological communities. *Ecology Letters*. 2022;25(12). doi:10.1111/ele.14123.
19. Gillespie MAK, Bjerger K, Alison J, Gerlich HS, Helsing-Nielsen F, Mougeot G, et al. Automated insect monitoring with camera traps is transforming ecological understanding. *EcoRxiv*. 2025. doi:<https://doi.org/10.32942/X2MW86>.
20. Droissart V, Azandi L, Onguene ER, Savignac M, Smith TB, Deblauwe V. PICT: A low-cost, modular, open-source camera trap system to study plant–insect interactions. *Methods in Ecology and Evolution*. 2021;12(8):1389-96. Available from: <https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/2041-210X.13618>. arXiv:<https://besjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/2041-210X.13618>. doi:<https://doi.org/10.1111/2041-210X.13618>.
21. Bjerger K, Alison J, Dyrmann M, Frigaard CE, Mann HMR, Høye TT. Accurate detection and identification of insects from camera trap images with deep learning. *PLOS Sustainability and Transformation*. 2023;2(3). Available from: <https://doi.org/10.1371/journal.pstr.0000051>. doi:<https://doi.org/10.1371/journal.pstr.0000051>.

22. Bjerger K, Karstoft H, Mann HMR, Høye TT. A deep learning pipeline for time-lapse camera monitoring of insects and their floral environments. *Ecological Informatics*. 2024;84. doi:10.1016/j.ecoinf.2024.102861.
23. Roosjen PPJ, Kellenberger B, Kooistra L, Green DR, Fahrentrapp J. Deep learning for automated detection of *Drosophila suzukii*: potential for UAV-based monitoring. *Pest Management Science*. 2020;76(9):2994-3002. doi:10.1002/ps.5845.
24. Molina-Rotger M, Morán A, Miranda MA, Alorda-Ladaria B. Remote fruit fly detection using computer vision and machine learning-based electronic trap. *Frontiers in Plant Science*. 2023;14:1241576. doi:10.3389/fpls.2023.1241576.
25. Serra-Marin PE, Solé-Ribalta A, Lana A, Borge-Holthoefer J, Hervías-Parejo S, Traveset A. Comparative assessment of automated and manual monitoring in comprehensive plant–pollinator communities. *Methods in Ecology and Evolution*. 2025;16. doi:10.1111/2041-210X.70165.
26. Wang B, Jägermeyr J, O’Leary GJ, Wallach D, Ruane AC, Feng P, et al. Pathways to identify and reduce uncertainties in agricultural climate impact assessments. *Nature Food*. 2024;5. doi:10.1038/s43016-024-01014-w.
27. Bjerger K, Geissmann Q, Alison J, Mann HMR, Høye TT, Dyrmann M. Hierarchical Classification of Insects with Multitask Learning and Anomaly Detection. *Ecological Informatics*. 2023;77(102278). doi:https://doi.org/10.1016/j.ecoinf.2023.102278.
28. Beery S, Van Horn G, Perona P. Recognition in Terra Incognita. In: *Lecture Notes in Computer Science* (including subseries *Lecture Notes in Artificial Intelligence* and *Lecture Notes in Bioinformatics*). vol. 11220 LNCS; 2018. doi:10.1007/978-3-030-01270-028.
29. Svenning A, Mougeot G, Alison J, Chevalier D, Molina NC, Ong SQ, et al. A general method for detection and segmentation of terrestrial arthropods in images. *Methods in Ecology and Evolution*. 2026;17(3):727-39. Available from: <https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/2041-210x.70249>. arXiv:https://besjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/2041-210x.70249. doi:https://doi.org/10.1111/2041-210x.70249.
30. Oliver RY, Iannarilli F, Ahumada J, Fegraus E, Flores N, Kays R, et al. Camera trapping expands the view into global biodiversity and its change. *Philosophical Transactions of the Royal Society B: Biological Sciences*. 2023;378(1881). doi:10.1098/rstb.2022.0232.

31. Bjerger K, Frigaard CE, Karstoft H. Object Detection of Small Insects in Time-Lapse Camera Recordings. *Sensors*. 2023;23(16). Available from: <https://www.mdpi.com/1424-8220/23/16/7242>. doi:10.3390/s23167242.
32. Ratnayake MN, Dyer AG, Dorin A. Tracking individual honeybees among wildflower clusters with computer vision-facilitated pollinator monitoring. *PLOS ONE*. 2021 02;16(2):1-20. Available from: <https://doi.org/10.1371/journal.pone.0239504>. doi:10.1371/journal.pone.0239504.
33. Preti M, Verheggen F, Angeli S. Insect pest monitoring with camera-equipped traps: strengths and limitations. *Journal of Pest Science*. 2021 Mar;94(2):203-17. Available from: <https://doi.org/10.1007/s10340-020-01309-4>. doi:10.1007/s10340-020-01309-4.
34. Varga-Szilay Z, Szövényi G, Pozsgai G. Flower Visitation through the Lens: Exploring the Foraging Behaviour of *Bombus terrestris* with a Computer Vision-Based Application. *Insects*. 2024;15(9). Available from: <https://www.mdpi.com/2075-4450/15/9/729>. doi:10.3390/insects15090729.
35. Bjerger K, Mann HMR, Høye TT. Real-time insect tracking and monitoring with computer vision and deep learning. *Remote Sensing in Ecology and Conservation*. 2021;8(6). doi:10.1002/rse2.245.
36. Bjerger K, Karstoft H, Høye TT. Towards edge processing of images from insect camera traps. *Remote Sensing in Ecology and Conservation*. 2025;11. doi:10.1002/rse2.70007.
37. Khanam R, Hussain M. YOLOv11: An Overview of the Key Architectural Enhancements; 2024. Available from: <https://arxiv.org/abs/2410.17725>. arXiv:2410.17725.
38. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*; 2016. arXiv:1506.02640. doi:10.1109/CVPR.2016.91.
39. Glenn Jocher. You Only Look Once Ver. 5 (YOLOv5) on Github; 2020. <https://github.com/ultralytics/yolov5> [Accessed: 21.05.2024].
40. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. vol. 2016-Decem; 2016. doi:10.1109/CVPR.2016.90.
41. Tan M, Le QV. EfficientNetV2: Smaller Models and Faster Training; 2021. Available from: <https://arxiv.org/abs/2104.00298>. arXiv:2104.00298.

42. Todi A, Narula N, Sharma M, Gupta U. ConvNext: A Contemporary Architecture for Convolutional Neural Networks for Image Classification. In: Proceedings - 2023 3rd International Conference on Innovative Sustainable Computational Technologies, CISCT 2023; 2023.
doi:10.1109/CISCT57197.2023.10351320.
43. Jodelet Q, Liu X, Murata T. Balanced softmax cross-entropy for incremental learning with and without memory. *Computer Vision and Image Understanding*. 2022;225:103582. Available from: <https://www.sciencedirect.com/science/article/pii/S1077314222001606>.
doi:<https://doi.org/10.1016/j.cviu.2022.103582>.
44. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*. 2015;115(3).
doi:10.1007/s11263-015-0816-y.
45. Bulusu S, Kailkhura B, Li B, Varshney PK, Song D. Anomalous Example Detection in Deep Learning: A Survey. *IEEE Access*. 2020;8. doi:10.1109/ACCESS.2020.3010274.
46. Høye TT, Montagna M, Oteman B, Roy DB. Emerging technologies for pollinator monitoring. *Current Opinion in Insect Science*. 2025;69:101367.
47. Corbet SA, Fussell M, Ake R, Fraser A, Gunson C, Savage A, et al. Temperature and the pollinating activity of social bees. *Ecological Entomology*. 1993;18(1):17-30.
doi:10.1111/j.1365-2311.1993.tb01075.x.
48. Gilbert FS. Diurnal activity patterns in hoverflies (Diptera, Syrphidae). *Ecological Entomology*. 1985;10(4):385-92. doi:10.1111/j.1365-2311.1985.tb00736.x.
49. Zoller L, Bennett JM, Knight TM. Diel-scale temporal dynamics in the abundance and composition of pollinators in the Arctic summer. *Scientific Reports*. 2020;10:21187. doi:10.1038/s41598-020-78165-w.
50. Mahon N, Hodge S. Evaluating the effects of observation period, floral density, and weather conditions on the consistency and accuracy of timed pollinator counts. *Journal of Pollination Ecology*. 2022;32:124-38. doi:10.26786/1920-7603(2022)699.
51. Naqvi Q, Wolff PJ, Molano-Flores B, Sperry JH. Camera traps are an effective tool for monitoring insect–plant interactions. *Ecology and Evolution*. 2022;12(6). doi:10.1002/ece3.8962.

52. Rowcliffe JM, Field J, Turvey ST, Carbone C. Estimating animal density using camera traps without the need for individual recognition. *Journal of Applied Ecology*. 2008;45(4):1228-36.
doi:10.1111/j.1365-2664.2008.01473.x.
53. Rowcliffe JM, Kays R, Kranstauber B, Carbone C, Jansen PA. Quantifying levels of animal activity using camera trap data. *Methods in Ecology and Evolution*. 2014;5(11):1170-9.
doi:10.1111/2041-210X.12278.
54. MacKenzie DI, Nichols JD, Lachman GB, Droege S, Royle JA, Langtimm CA. Estimating site occupancy rates when detection probabilities are less than one. *Ecology*. 2002;83(8):2248-55.
doi:10.1890/0012-9658(2002)083[2248:ESORWD]2.0.CO;2.
55. Zhang Y, Sun P, Jiang Y, Yu D, Weng F, Yuan Z, et al.. ByteTrack: Multi-Object Tracking by Associating Every Detection Box; 2022. Available from: <https://arxiv.org/abs/2110.06864>.
arXiv:2110.06864.
56. Aharon N, Orfaig R, Bobrovsky BZ. BoT-SORT: Robust Associations Multi-Pedestrian Tracking; 2022. Available from: <https://arxiv.org/abs/2206.14651>. arXiv:2206.14651.

Supporting information

Figures S1–S6 present example images from the detection dataset, illustrating the diversity of plant backgrounds in several projects to train a robust and generalized insect detector.

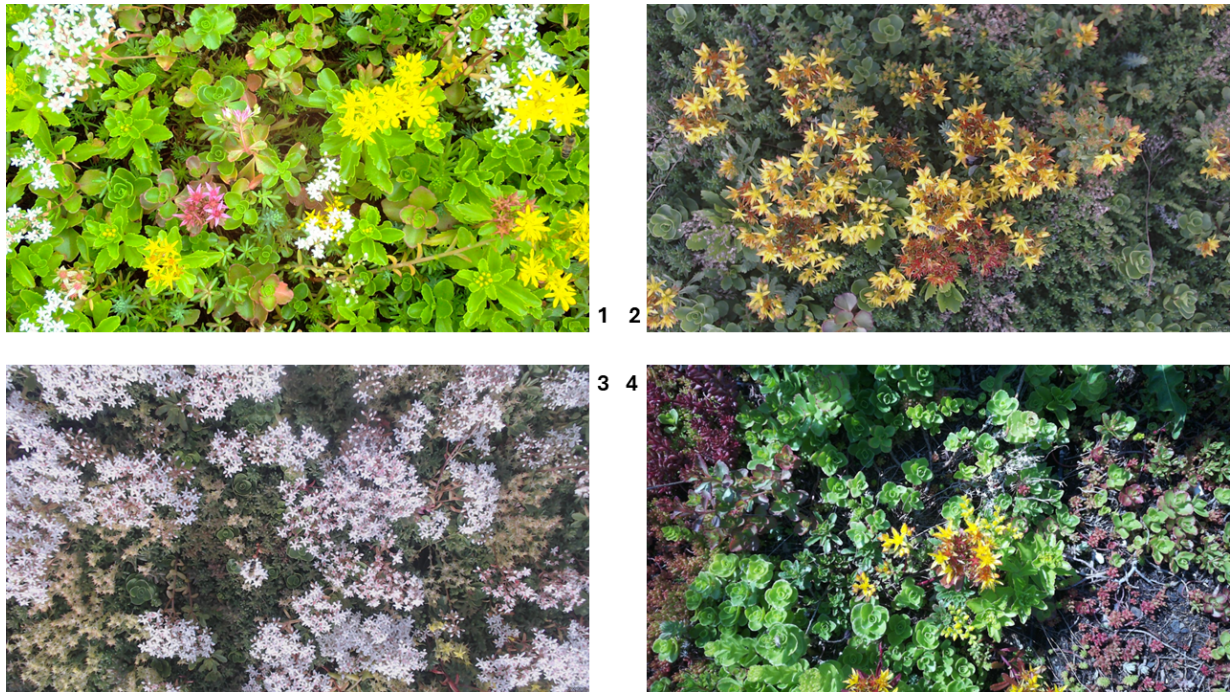


Fig. S1. Examples of *Sedum* plants are presented from the **GreenRoof** project (1–3) and the **Orchard** project (4).

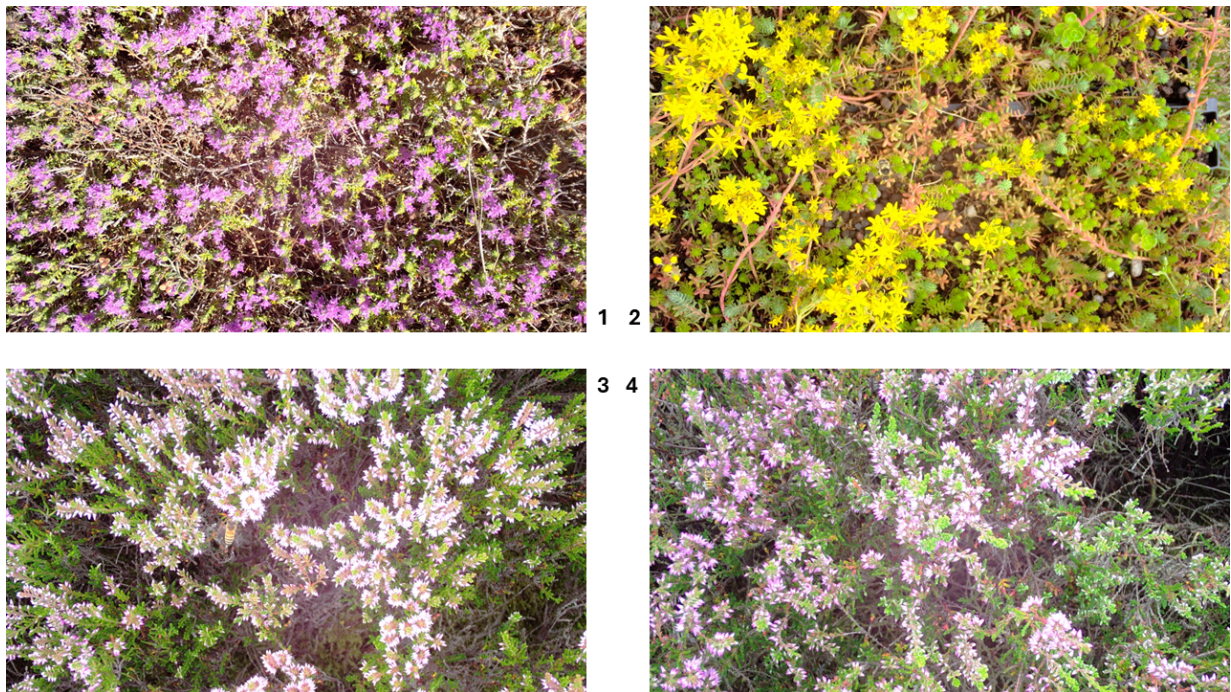


Fig. S2. Examples of Herbs, *Sedum* and Heather plants are presented from the **MAMBO** project (1,2) and the **Heather** project (3,4).

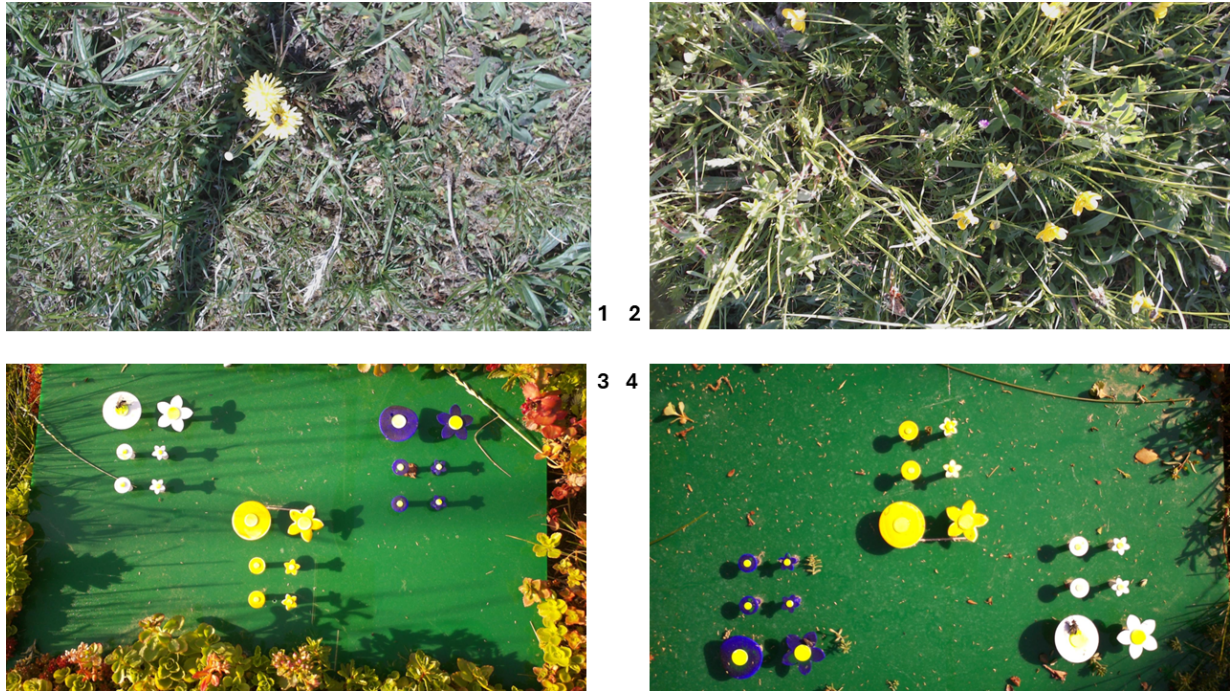


Fig. S3. Examples of mixed plants and artificial flowers are presented from the **PollWatch** (1,2) and **Orchard** (3,4) projects.

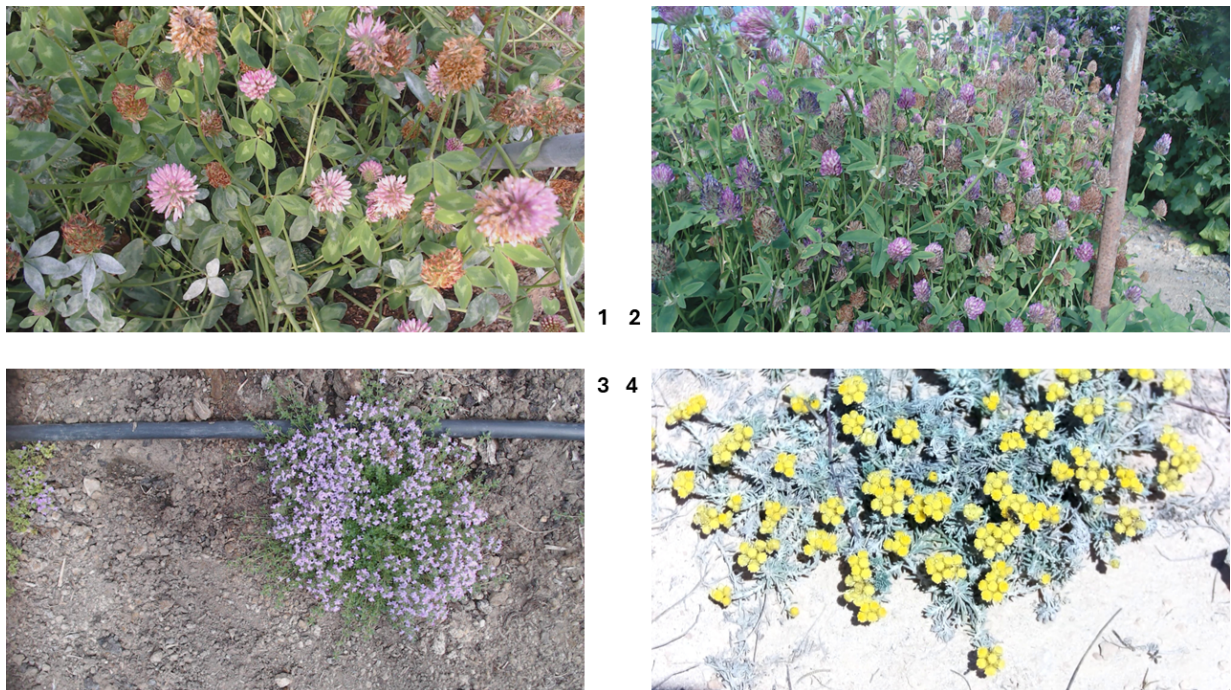


Fig. S4. Examples of Red Clover, Sea Rocket plants are presented from the **GreenHouse** (1-3) and **ACSHQ** (4) projects.

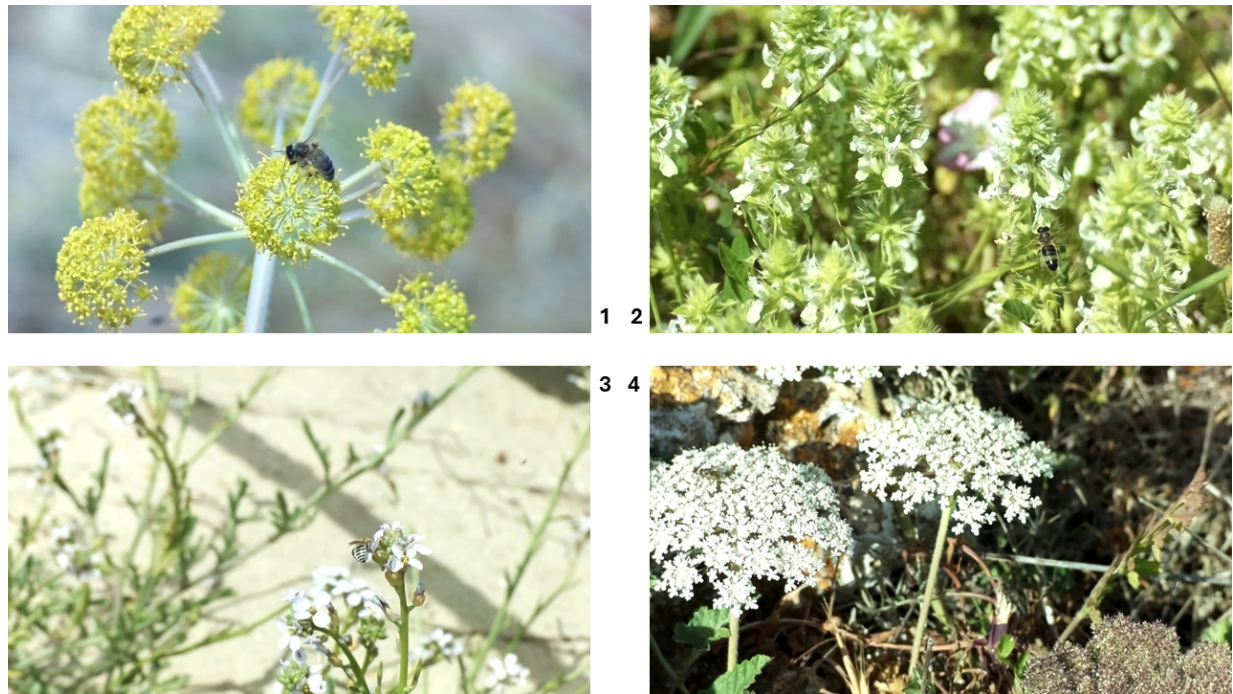


Fig. S5. Examples of different plants with insects are presented from the ACSHQ project.



Fig. S6. Examples of different plants with insects are presented from the ACSHQ project.