

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DISI)

In collaboration with the Department of Ecoscience, Aarhus University (AU)

ARTIFICIAL INTELLIGENCE

MASTER THESIS

IN

IMAGE PROCESSING AND COMPUTER VISION

Leveraging Foundation Models to Improve Insect Localization in Field Images

SUPERVISOR:

Prof. Giuseppe Lisanti (DISI)

CO-SUPERVISORS:

Dr. Jamie Alison (AU)

Prof. Toke Thomas Høye (AU)

CANDIDATE:

Lorenzo Girotti

ACADEMIC YEAR 2025/2026

SESSION II

Abstract

The automatic detection of insects in natural environments is a critical challenge for ecological monitoring. While detection on homogeneous backgrounds is largely solved, localizing insects in “*wild*” settings with complex textures remains an open task. To address this limitation, this thesis proposes a novel zero-shot detection pipeline leveraging Grounding DINO and Segment Anything Model 3 (SAM3).

After evaluating multiple prompting strategies, a lightweight supervised binary discriminator is introduced as a post-processing step to re-score predictions based on the presence of insects. Benchmarked across diverse full-frame and close-up datasets, the proposed pipeline demonstrates exceptional out-of-distribution capabilities, consistently outperforming state-of-the-art supervised models in single flower monitoring scenarios. Finally, the limitations of this approach are discussed, particularly regarding the computational cost of foundation models inference.

Contents

1	Introduction	11
2	Background	15
2.1	Insect monitoring	15
2.1.1	Insect biodiversity tracking	15
2.1.2	Manual monitoring methodologies	16
2.1.3	Automated monitoring through computer vision	17
2.2	Object detection	21
2.2.1	Task definition	21
2.2.2	Supervised architectures	21
2.2.3	Small-instance detection challenge	22
2.3	Vision Foundation Models	24
2.3.1	Vision transformers	24
2.3.2	Transformer-based object detection	25
2.3.3	Vision-Language alignment	25
3	Related work	27
3.1	Open-vocabulary and promptable vision models	27
3.2	Visual representation learning	29
3.3	Supervised baselines	30
4	Project	35
4.1	Datasets and data preparation	35
4.1.1	Ground ECOSTACK dataset	36
4.1.2	Bjerge et al. dataset	36
4.1.3	RangeX dataset	39
4.1.4	Ştefan et al. dataset	39
4.1.5	Dataset annotations normalization	41
4.2	Inference and evaluation framework	41
4.2.1	Implementation details	43
4.2.2	Tiled inference strategy	43
4.2.3	Evaluation metrics	43

4.2.4	Prompt design	47
4.2.5	Zero-shot detectors hyperparameters	48
4.2.6	Proposed ensembling strategy	48
4.3	Expert binary discriminator	49
4.3.1	Feature space analysis and dimensionality reduction	49
4.3.2	Expert classifier training	51
4.3.3	Scoring functions	52
5	Results and discussion	55
5.1	Supervised baseline domain shift problem	55
5.2	Zero-shot detector generalization capabilities	57
5.3	Results of the proposed ensembling strategy	58
5.4	Negative prompting analysis	60
5.5	Binary discriminator results	62
5.6	Limitations and future work	65
6	Conclusions	69

List of Figures

2.1	Automated monitoring systems: Sticky Pi and AMI trap	18
2.2	Automated monitoring systems <i>in the wild</i>	20
2.3	Architectural comparison between a single-stage and a two-stage object detector [1].	22
2.4	Slicing Aided Hyper Inference (SAHI) approach for small-instance detection [2].	23
3.1	SAM3 architecture diagram [3].	28
4.1	Sample images from the Ground ECOSTACK dataset test split (crop of the original images).	37
4.2	Sample images from the Bjerger et al. dataset test split (crop of the original images).	38
4.3	Sample images from the RangeX dataset test split.	40
4.4	Sample images from the Ştefan et al. test split.	42
4.5	t-SNE visualizations for the Bjerger et al. training split using DINOv3 and BioCLIP 2.5 embedding vectors.	50
4.6	t-SNE visualizations for the Ground ECOSTACK training split using DINOv3 and BioCLIP 2.5 embedding vectors.	50

List of Tables

4.1	Distribution of annotated instances across dataset splits in the Ground ECOSTACK dataset	36
4.2	Distribution of annotated instances across the Bjerge et al. dataset splits .	37
4.3	Distribution of annotated instances across the RangeX dataset splits. . . .	40
4.4	Distribution of annotated instances across the Ştefan et al. dataset splits. .	41
5.1	Supervised models performance on the Ground ECOSTACK dataset	56
5.2	Supervised models performance on the Bjerge et al. dataset	56
5.3	Supervised models performance on the RangeX dataset	56
5.4	Supervised models performance on the Ştefan et al. dataset	57
5.5	Zero-shot performance detectors on the Ground ECOSTACK dataset. See Section 4.2.4 for detailed descriptions of the prompt configurations.	58
5.6	Zero-shot performance detectors on the Bjerge et al. dataset. See Section 4.2.4 for detailed descriptions of the prompt configurations.	58
5.7	Zero-shot performance detectors on the RangeX dataset. See Section 4.2.4 for detailed descriptions of the prompt configurations.	59
5.8	Zero-shot performance detectors on the Ştefan et al. dataset. See Section 4.2.4 for detailed descriptions of the prompt configurations.	59
5.9	Performance of the proposed ensembling strategy for zero-shot detectors on the Ground ECOSTACK dataset. See Section 4.2.4 for detailed descriptions of the prompt configurations.	60
5.10	Performance of the proposed ensembling strategy for zero-shot detectors on the Bjerge et al. dataset. See Section 4.2.4 for detailed descriptions of the prompt configurations.	60
5.11	Performance of the proposed ensembling strategy for zero-shot detectors on the RangeX dataset. See Section 4.2.4 for detailed descriptions of the prompt configurations.	61
5.12	Performance of the proposed ensembling strategy for zero-shot detectors on the Ştefan et al. dataset. See Section 4.2.4 for detailed descriptions of the prompt configurations.	61

5.13	Performance of negative prompting with Grounding DINO on the RangeX dataset. See Section 4.2.4 for detailed descriptions of the prompt configurations.	61
5.14	Performance metrics of MLP and SVM classifiers on test sets. See Section 4.2.4 for detailed descriptions of the prompt configurations.	62
5.15	Performance of embedding-based re-scoring on the RangeX dataset by using an expert discriminator (joint scoring). See Section 4.2.4 for detailed descriptions of the prompt configurations.	63
5.16	Performance of embedding-based re-scoring on the RangeX dataset by using an expert discriminator (mean scoring). See Section 4.2.4 for detailed descriptions of the prompt configurations.	64
5.17	Performance of embedding-based re-scoring on the Ştefan et al. dataset by using an expert discriminator (joint scoring). See Section 4.2.4 for detailed descriptions of the prompt configurations.	64
5.18	Performance of embedding-based re-scoring on the Ştefan et al. dataset by using an expert discriminator (mean scoring). See Section 4.2.4 for detailed descriptions of the prompt configurations.	64
5.19	Inference time comparison across different models and datasets.	65

Chapter 1

Introduction

Insects represent the most diverse animal group on Earth, comprising more than half of all known species [4]. Beyond the profound interest inspired by their broad variety and complex behaviors, they are indispensable to terrestrial ecosystems. They play vital functional roles, serving as primary pollinators, natural pest controllers, and essential agents of decomposition and nutrient cycling [5, 6].

In recent decades, however, global insect populations have suffered a dramatic decline [7]. Driven by climate change and human-induced pressures, this loss poses severe risks to ecosystem stability and human well-being through the collapse of critical ecosystem services and the unmonitored spread of agricultural pests.

Consequently, scalable monitoring has become urgent. Yet, traditional methods remain highly dependent on manual trapping and visual identification. Given the vast diversity of insects, this reliance on human labor creates a significant bottleneck that hinders both timely conservation efforts and rapid control measures for pests [8].

To overcome this bottleneck, Artificial Intelligence (AI), specifically Computer Vision, has emerged as a key technology to scale up biodiversity monitoring. Unlike manual methods, AI enables automated, non-invasive monitoring capable of processing the vast data streams generated by passive monitoring devices, such as time-lapse cameras. The continuous deployment of these devices also allows for the capture of information regarding community dynamics and insect behavior throughout the entire day, a task that is nearly impossible or very expensive to achieve with manual field monitoring. Devices such as cameras, often deployed for long periods, generate massive volumes of imagery, much of which captures only empty backgrounds. By accurately localizing insects, AI systems can efficiently filter out these non-informative images, automate individual counting, and enable further classification of natural features, such as flowers, fungi, or dung.

In the domain of insect monitoring via camera traps, numerous systems have been developed to automate data collection. These range from simple pitfall camera traps, which can be deployed in both laboratory and field settings, to sticky traps, which utilize colored plates to attract insects and a facing camera to capture them. Crucially,

these methods rely on homogeneous backgrounds, often specific colors designed to attract target species and simplify the visual scene. Thanks to this controlled environment, computer vision implementations have achieved significant success in detecting and localizing insect boundaries, such as FlatBug [9], while others have also enabled accurate species classification at high-order taxonomic levels [10].

However, a significant research gap remains in the context of localizing insects in wild settings. Unlike controlled setups, this approach abandons the use of engineered, homogeneous backgrounds in favor of orienting camera traps directly toward the natural substrate. This configuration introduces substantial challenges, including extreme background diversity, heterogeneous insect scales depending on the application, variable illumination, and the presence of natural debris (e.g., leaves, sticks) that can easily be mistaken for insects even in high-resolution imagery.

Recently, the computer vision landscape has been transformed by the emergence of powerful zero-shot foundation models, which demonstrate robust generalization capabilities across highly diverse and unseen scenarios. Novel and increasingly popular examples of this paradigm include the Segment Anything Model (SAM) family. These models offer versatile and promptable interfaces that can theoretically bypass the need for domain-specific re-training. However, deploying such massive architectures in real-world production environments presents significant bottlenecks. They are heavily resource-intensive, requiring substantial computational power and memory, which translates to high inference latency and high costs for both fine-tuning and continuous at-scale operations.

Nonetheless, if they provide sufficient out-of-the-box performance, they may still be preferable to custom alternatives, as repeatedly annotating data and training new supervised models is equally resource-intensive.

This thesis investigates fundamental trade-offs between domain-specific supervised object detection models and modern zero-shot Vision-Language models when applied to ecological monitoring.

Specifically, this work evaluates how domain-specific supervised baseline performance differs between their training distribution (In-Domain-Distribution) and datasets with distinct environmental conditions, possibly captured with different sensors (Out-Of-Distribution). To explore a more resilient alternative, this study benchmarks state-of-the-art foundation models, SAM3 [3] and Grounding DINO [11]. A core focus of the analysis is exploring the inherent prompt sensitivity of these zero-shot architectures. This challenge is tackled by evaluating simple, contextual, and taxonomic prompting strategies, alongside negative prompting and an ensembling technique to mitigate the variability of performance across different prompts.

To further enhance the performance of generalist zero-shot models, this work introduces a lightweight supervised binary discriminator used as a post-processing step, with the purpose of refining predictions obtained by foundation models. This discriminator re-scores each candidate crop based on the likelihood of it containing an insect. This

approach allows for domain-specific specialization without the need to re-train the entire detection architecture, which is computationally expensive and requires large amounts of spatially annotated data.

Results obtained on the evaluated datasets demonstrate that the proposed approach, especially in the context of single flower monitoring, consistently outperforms state-of-the-art supervised models while also showing promising results in detecting insects at smaller scales.

Chapter 2

Background

2.1 Insect monitoring

2.1.1 Insect biodiversity tracking

Insects play a foundational role in ecosystems. As the most diverse animal group on the planet, they serve vital functions as pollinators, decomposers, and a primary food source for numerous species. Consequently, monitoring insect biodiversity is essential for assessing ecosystem health and understanding the broader impacts of environmental changes.

Recent reports highlighting significant declines in insect populations [7, 12, 13], particularly among pollinators, underscore the urgent need not only to track these trends but also to investigate their extent and drivers, such as agricultural intensification, climate change, and urbanization, which severely threaten environmental stability [14]. At the same time, a comprehensive monitoring framework must also account for insects as agricultural pests and problem species. As climate change continues to alter their biology and ecology, modern prediction and tracking tools are urgently needed to devise effective management strategies that safeguard crop production [15]. Ultimately, tracking both beneficial and harmful species is vital to inform conservation strategies, protect natural ecosystems, and ensure global food security.

For these reasons, biodiversity monitoring has become a critical area of research, and continuous monitoring systems now offer an unprecedented opportunity to capture fine-scale behavioral shifts and track how climate change reshapes insect phenology and ecological interactions over time.

The following sections review manual and modern automated methodologies for insect monitoring, highlighting the intrinsic challenges of traditional approaches and the potential of computer vision techniques.

2.1.2 Manual monitoring methodologies

Traditional insect monitoring methodologies often involve manual labor, such as field sampling, physical traps, and consequent laboratory work. These methods can be time-consuming, labor-intensive, and may not provide continuous data throughout the day and growing season. Additionally, they can be invasive or lethal to insects and might not capture the full diversity of insect communities, especially in wild environments where insects can be elusive and difficult to observe.

However, traditional methods play an essential role in understanding ecological baselines and validating results from automated monitoring systems, and often capture aspects of insect diversity that cameras do not, such as morphologically cryptic species. Physical specimen collection enables a level of taxonomic precision not yet achievable with remote sensing. Moreover, field surveying remains foundational to ecological literacy, as it builds the field-based expertise necessary to interpret data from both manual and automated systems [16].

Depending on the target taxa and the specific research questions, different manual methodologies are commonly employed.

Direct observations

To build plant-insect interaction networks, ecologists often rely on direct human observations. Researchers physically monitor specific floral patches for predefined time intervals, manually recording the occurrence and duration of insect visits. While this approach provides high-quality behavioral data, it is severely constrained by human fatigue and weather conditions. Additionally, it risks introducing an observer effect, where the physical presence of a researcher alters natural wildlife behavior. Furthermore, the presence of many monitoring schemes and protocols can lead to significant variability in data collection methods and incompatibility of results across different studies.

Sweep netting

Sweep netting is an active sampling technique used to characterize vegetation-dwelling and flying insects (e.g., flies, bees, wasps). Researchers manually pass a specialized sweep net over vegetation along a linear transect. This method is highly dependent on specific environmental conditions, typically requiring dry and sunny hours, and introduces considerable human bias based on the individual's sweeping technique. The captured specimens must be manually transferred into sterile tubes with a preservative liquid for subsequent freezing and laboratory analysis. Beyond the labor-intensive post-processing, this method exhibits a strong sampling bias, as it disproportionately captures larger or aggregating species, leading to the systematic undersampling of smaller or solitary taxa [17].

Pitfall traps

Pitfall traps are the standard passive methodology for collecting ground-dwelling arthropods, such as spiders, beetles, and ants [18]. They consist of plastic containers placed at ground level, partially filled with a preservative fluid like propylene glycol, and protected by a small roof. While effective, the deployment and retrieval require significant physical labor. Traps are typically left in the field for several weeks, meaning the resulting data represents a delayed, cumulative snapshot rather than real-time monitoring. Furthermore, the collected samples contain a mixture of insects and environmental debris (soil, vegetation fragments, stones), introducing an important bottleneck during the manual sorting phase in the laboratory [19]. It is also likely that they undersample some insect groups, such as grasshoppers.

Malaise traps

Malaise traps are large, pyramidal structures designed to intercept flying insects. Insects fly into the tent wall and naturally move upward toward the light, eventually falling into a collection vessel filled with a preservative fluid. While they require less active human intervention than sweep netting, they indiscriminately capture vast quantities of biomass. Sorting, counting, and classifying the thousands of specimens generated by a single Malaise trap requires hundreds of hours of expert taxonomic labor, or the use of DNA barcoding techniques, which are, on the other hand, expensive and require additional genetics expertise [19].

DNA barcoding

Once physical specimens are collected from the aforementioned methods, visual identification under a microscope might be insufficient for distinguishing cryptic species, or too time-consuming for large sample sizes in high-diversity environments. DNA barcoding has emerged as a powerful, highly automated tool for accurate species identification. By sequencing a short, standardized section of the insect's DNA and comparing it against online reference databases, researchers can achieve highly precise classifications. However, from a continuous monitoring perspective, this remains a laboratory-based process rather than a real-time field sensor. While it offers unparalleled taxonomic resolution, it can be costly, may require the destruction of the specimen, and cannot easily provide continuous data on insect abundance, behavior, or fine-scale temporal activity patterns [19].

2.1.3 Automated monitoring through computer vision

With the advancement of computer vision and machine learning techniques, there has been a growing interest in developing automated monitoring systems for insects, aiming to complement traditional approaches by exploiting the full potential of novel AI

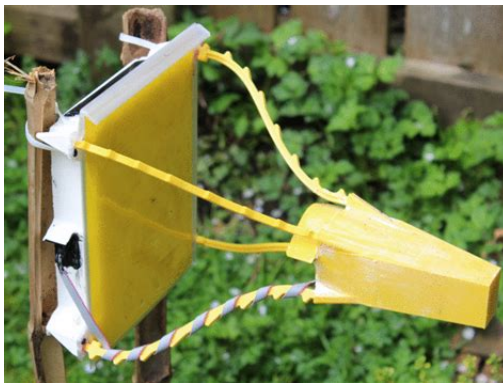
technologies. The use of cameras and subsequent image analysis allows for non-invasive and continuous monitoring of insect populations, providing high-resolution data to track phenological events, abundance, diversity, and community dynamics.

Sticky traps and fixed-background setups

The most straightforward automated monitoring setups rely on attractive mechanisms, such as sticky cards or light traps, that secure insects either temporarily or permanently within a restricted field of view. By integrating cameras into these physical traps, researchers can continuously capture images of the specimens for subsequent computer vision analysis.

Several initiatives have proposed integrated, end-to-end systems that combine custom hardware with dedicated software to automate this process. For example, the *Sticky Pi* [20] system leverages open-source hardware to record high-frequency temporal data of insects landing on adhesive surfaces, utilizing a specific built-in model to identify the captured specimens. Similarly, for nocturnal monitoring, the *Automated Monitoring of Insects (AMI)* trap uses artificial illumination to attract and take images of species, such as moths, against a uniform background [21, 22], with a further model to detect and classify the captured insects. Another example is *InsectDetect*, an open-source platform tailored for edge AI that integrates custom camera hardware with on-device models, enabling real-time insect detection directly in the field without relying on offline processing [10].

The deployment of such integrated automated setups significantly enhances the efficiency and scope of ecological surveys. Recent studies [23, 24] indicate that the data produced by these automated pipelines yield estimates of species richness, abundance, and population trends that directly rival conventional, destructive trapping methods. The inherent scalability and high temporal resolution of these platforms allow for the precise monitoring of phenological events, the generation of standardized biodiversity metrics, and a more profound understanding of community-level interactions.



(a) Sticky Pi [20]



(b) AMI trap [21]

Figure 2.1: Automated monitoring systems: Sticky Pi and AMI trap

Automated camera systems in the wild

To capture a more comprehensive and informative picture of insect diversity and behavior, many studies have explored the deployment of automated camera systems “*in the wild*”, without relying on artificial attractive mechanisms. These setups typically involve time-lapse cameras directed at natural habitats, such as specific flowers or vegetation crops, to observe insects interacting naturally with their environment and with other species.

As in the previous section, several studies have developed end-to-end systems integrating custom hardware and software to automate the monitoring process. For instance, extensive time-lapse camera networks have been deployed over flowering plants to monitor functionally important taxa like bees, hoverflies, and butterflies [25]. When used to train models like You Only Look Once (YOLO) [26], these systems achieve high accuracy across similar images in detecting and classifying dominant pollinator species.

Building on custom frameworks, more scalable open-source Automated Camera Systems (ACS) utilizing single-board computers (e.g., Raspberry Pi) paired with deep learning pollinator detectors have also been deployed. By cross-validating ACS data with direct human surveys, researchers found that automated units record significantly higher pollinator visitation rates, likely because the absence of human observers prevents the alteration of natural insect behavior [27]. Furthermore, these continuous setups capture both diurnal and nocturnal plant-pollinator interactions, successfully logging rare, low-frequency events. However, they exhibit notable limitations in reliably detecting very small-bodied instances, such as under 5 mm size.

Exploring even more accessible commercial hardware, other studies have demonstrated the feasibility of using smartphones as automated time-lapse sensors [28]. Positioned directly above target flowers, smartphones offer a cost-effective way to capture high-resolution images. This image quality is often sufficient for accurate taxonomic identification down to the genus or species level for many bees. However, aligning with the limitations previously observed in ACS, identifying smaller insects like flies and ants remains challenging due to the camera’s inability to resolve microscopic morphological features in camouflaged backgrounds.

Despite these hardware advancements, operating in the wild introduces significant computer vision challenges. Unlike sticky traps, insects in natural habitats often appear as tiny objects against highly complex and dynamically moving backgrounds, such as wind-blown vegetation. To address this, specialized algorithmic pipelines have been proposed to enhance standard object detectors. For example, motion-informed enhancement techniques have been developed to preprocess time-lapse sequences. By leveraging motion and color cues, these algorithms isolate insects from background noise before feeding the images into the network [29]. This two-step approach significantly increases performance of detectors, highlighting the necessity of context-aware preprocessing when monitoring non-constrained natural scenes.

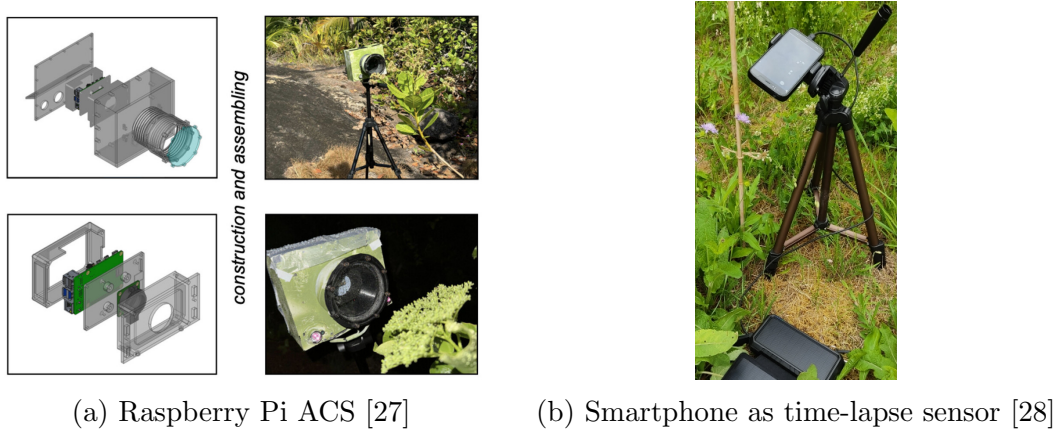


Figure 2.2: Automated monitoring systems *in the wild*

Limitations of automated monitoring

Despite the technological advances discussed in the previous sections, deploying automated monitoring systems in real-world scenarios still involves several practical and methodological challenges, as highlighted in this comprehensive review [30].

First, these systems are rarely fully autonomous, since they still require considerable manual effort for field maintenance, power management, and data collection. In addition, the absence of standardized hardware and software pipelines often compromises reproducibility and hinders large-scale adoption.

From a computer vision point of view, one of the main bottlenecks involves the classification stage. Given the enormous diversity of insect species, many of which exhibit highly similar visual traits, reliable species-level identification remains extremely challenging. This issue is further influenced by the limited camera resolution and by the inherently small size of the subjects, which also might negatively affect the detection stage. This implies that the classification could be, in the majority of the cases, limited to higher taxonomic levels (e.g., family or genus), which might not be sufficient to answer some ecological questions.

Furthermore, this classification bottleneck is severely influenced by the ecological “long tail” problem, where a few species are highly abundant but the vast majority occur very rarely. This natural imbalance makes it exceedingly difficult to gather enough examples to effectively train automated systems for rare species, a common problem in computer vision tasks.

Moreover, supervised deep learning approaches commonly adopted in these pipelines require large amounts of annotated data to achieve robust performance. However, even with abundant data, critical challenges persist: domain shift and shortcut learning significantly degrade model performance when applied to unseen environments or under variations in illumination, scale, background, and acquisition conditions [31]. These challenges are further discussed and mitigated in the following chapters.

2.2 Object detection

2.2.1 Task definition

Object detection is a fundamental computer vision task; its objective is to identify and localize, by defining a bounding box, all instances of predefined object categories present in the input image.

Formally, given an input image $I \in \mathbb{R}^{H \times W \times C}$ (where H, W and C define height, width, and color channels respectively), the goal of an object detector is to predict a set of detections $\mathcal{D} = \{d_1, d_2, \dots, d_N\}$. Each detection d_i is represented as a tuple:

$$d_i = (c_i, p_i, \mathbf{b}_i) \quad (2.1)$$

where $c_i \in \mathcal{C}$ is the predicted class label from a predefined set of classes, $p_i \in [0, 1]$ is the confidence score of the prediction, and $\mathbf{b}_i$ represents the bounding box coordinates. The bounding box is usually parameterized by its center coordinates, width, and height: $\mathbf{b}_i = (x_c, y_c, w, h)$, known as the COCO format. Another common parametrization is the so called YOLO format, where the bounding box is defined by the coordinates of its top-left and bottom-right corners: $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$, with values normalized to the image dimensions in contrast to the absolute pixel values used in COCO.

For a pixel-level understanding, instance segmentation extends object detection by predicting a binary segmentation mask $\mathbf{m}_i \in \{0, 1\}^{H \times W}$ for each detected object, separating the foreground instances, belonging to the class c_i , from the background and overlapping objects.

2.2.2 Supervised architectures

Historically, supervised object detection architectures are categorized into two main categories: two-stage and one-stage detectors.

Two-stage detectors, such as Faster R-CNN [32], process the image in two distinct phases. Firstly, a Region Proposal Network (RPN) generates class-agnostic candidate Regions of Interest (RoIs) that likely contain an object. Then, a second network extracts features from these RoIs, using operations like *RoIPool*, to perform the final classification and bounding box refinement.

The network is trained by minimizing a multi-task loss function:

$$\mathcal{L} = \mathcal{L}_{cls}(p, p^*) + \lambda \mathcal{L}_{reg}(\mathbf{b}, \mathbf{b}^*) \quad (2.2)$$

where $\mathcal{L}_{cls}$ is typically a Cross-Entropy loss for classification (comparing predicted probability p with the ground truth p^*), $\mathcal{L}_{reg}$ is a regression loss (e.g., Smooth L_1) for the bounding box coordinates $\mathbf{b}$.

Indeed, bounding box prediction is treated as a relaxation of the original localization task: the model learns to estimate the precise coordinate offsets required to adjust predefined anchor boxes, fixed-size reference rectangles of varying scales and aspect ratios that tile the image, to match the corresponding ground-truth objects.

One-stage detectors, such as the YOLO (You Only Look Once) family [26], bypass the region proposal step. They formulate detection as a single regression problem, dividing the image into a grid and directly predicting bounding boxes and class probabilities for each grid cell. While two-stage models typically offer higher localization precision, one-stage models are significantly faster, making them the standard choice for real-time applications and edge computing deployment.

Thanks to their efficiency, as well as straightforward training pipeline documentation, YOLO models have become a standard baseline for supervised object detection tasks, including insect detection. All the supervised detection models further evaluated in this thesis are based on the YOLO architecture, with various versions and training strategies.

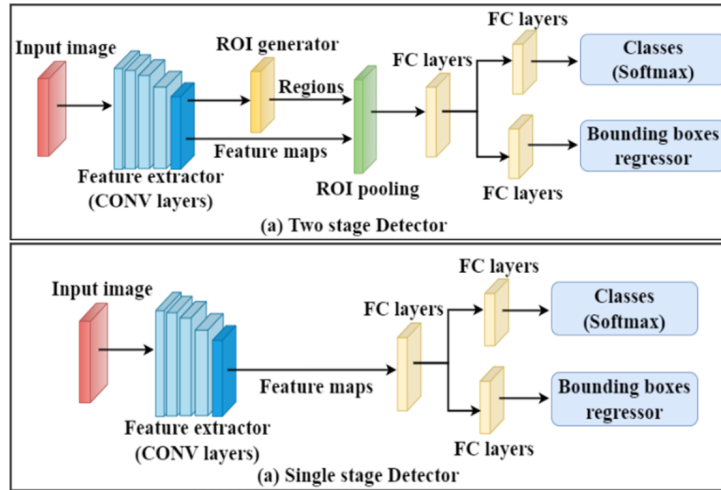


Figure 2.3: Architectural comparison between a single-stage and a two-stage object detector [1].

2.2.3 Small-instance detection challenge

Detecting insects inside images, due to their size, introduces the critical challenge of small-instance detection. In standard Convolutional Neural Networks (CNNs), the image undergoes progressive downsampling through pooling or strided convolutions to increase the receptive field and extract high-level semantic features.

If an image is downsampled by a factor of S (e.g., $S = 32$ in many CNN-based detectors), an object occupying 16×16 pixels in the original high-resolution image would

correspond to only 0.5×0.5 spatial units in the deepest feature map. As a result, the object may no longer retain sufficient spatial information for reliable localization and classification, severely degrading detection performance for small instances. A similar limitation applies to Vision Transformers (ViTs), where small targets can be completely absorbed within a single large token patch, leading to an equivalent loss of information.

Furthermore, feeding high-resolution images (e.g., $4K$ or $8K$) directly into deep architectures is computationally prohibitive due to memory constraints, especially with large foundation models. To overcome this bottleneck, slicing methodologies such as Slicing Aided Hyper Inference (SAHI) [2] have become standard practice. SAHI works by dividing the input image into a grid of overlapping patches $P_{i,j}$ of manageable dimensions (e.g., 512×512 pixels).

The detection model is applied independently to each patch, preserving the native resolution of small instances, so the inference time increases linearly with the number of patches. Finally, the predictions from all patches are mapped back to the original image coordinates and merged using Non-Maximum Suppression (NMS) algorithms to filter out redundant bounding boxes generated in the overlapping regions.

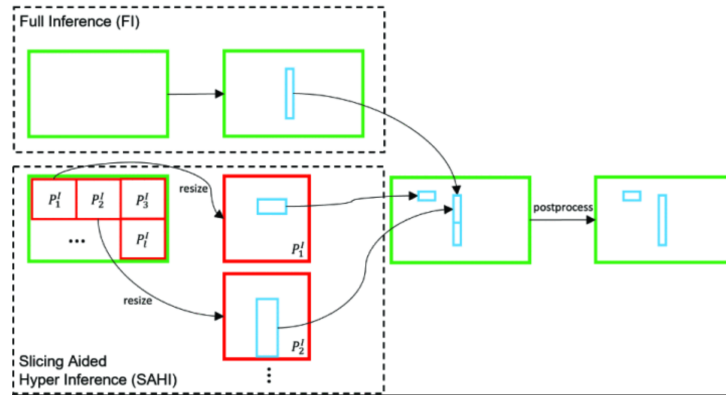


Figure 2.4: Slicing Aided Hyper Inference (SAHI) approach for small-instance detection [2].

An alternative, and often complementary, strategy to address small-instance detection is multi-scale inference. Rather than partitioning the image into fixed-size patches, this approach processes the input at several different resolutions (or scales) and aggregates the resulting predictions. Small objects that are missed or poorly localized at lower resolutions may become detectable once the image is upsampled, while larger objects remain reliably detected at coarser scales, where computational cost is lower. This is commonly implemented either as *image pyramids*, where the same image is resized multiple times before being independently fed to the detector, or as *feature pyramids* (e.g., Feature Pyramid Networks, FPN [33]), where multi-scale representations are built internally within the network by combining high-resolution, low-semantic features with low-resolution, high-semantic ones.

While multi-scale inference avoids some of the computational overhead associated with

slicing large images into many overlapping patches, it still increases inference time and memory usage compared to a single-scale forward pass, and predictions across scales must be merged using NMS to remove duplicate detections.

2.3 Vision Foundation Models

2.3.1 Vision transformers

Historically, CNNs have dominated computer vision tasks due to their inductive biases, specifically the translation equivariance property and localized receptive fields thanks to the kernel mechanism. However, the introduction of the Vision Transformer (ViT) [34] introduced a paradigm shift, adapting the purely attention-based architecture, originally designed for Natural Language Processing (NLP), to image data.

Instead of processing pixels sequentially or via sliding windows, a ViT treats an image as a sequence of tokens, a discrete representation of the visual input. Given an image $I \in \mathbb{R}^{H \times W \times C}$, it is first divided into a grid of non-overlapping patches $X_p \in \mathbb{R}^{N \times P \times P \times C}$, where $(P \times P)$ is the patch resolution and $N = \frac{H \times W}{P \times P}$ is the total number of patches, determining the effective sequence length.

These patches are flattened and mapped to a latent vector size D via a trainable linear projection, creating the patch embeddings. To retain spatial information, otherwise lost due to the permutation-invariant nature of self-attention, learnable 1D positional embeddings E_{pos} are added to the sequence:

$$\mathbf{z}_0 = [\mathbf{x}_{class}; \mathbf{x}_p^1 \mathbf{E}; \dots; \mathbf{x}_p^N \mathbf{E}] + \mathbf{E}_{pos} \quad (2.3)$$

where $\mathbf{x}_{class}$ is a specialized classification token prepended to the sequence, and $\mathbf{E}$ is the linear projection matrix.

The core of the architecture relies on the Multi-Head Self-Attention (MHSA) mechanism. For each attention head, the input is projected into Queries (Q), Keys (K), and Values (V). The attention weights are computed by taking the dot product of the query with all keys, typically scaled by the square root of the key dimension (d_k) to normalize the attention scores, and passed through a softmax function to obtain the attention distribution:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \quad (2.4)$$

Thanks to MHSA, attention is computed across different representation subspaces, allowing the model to capture various types of relationships between patches.

Unlike CNNs, which progressively enlarge the receptive field through stacked convolutional layers, self-attention enables ViTs to model long-range dependencies between patches from the very first layer.

While this global context is highly advantageous for understanding the overall image structure, it introduces significant challenges for scale-dependent tasks and small instance detection.

2.3.2 Transformer-based object detection

Standard supervised detectors, as discussed in Section 2.2.2, heavily rely on hand-designed components such as anchor box generation, spatial priors, and NMS to filter redundant predictions. The DETection TRansformer (DETR) [35] revolutionized this pipeline by framing object detection as a direct set prediction problem, eliminating the need for these heuristic components.

The key innovation of DETR lies in the decoder, which operates on a fixed set of learnable embeddings called *object queries*. During decoding, these queries first interact with each other through self-attention, allowing the model to jointly reason over multiple object hypotheses. They then attend to the encoder feature map through cross-attention, aggregating visual information from the image. Finally, each query is processed by a shared Feed-Forward Network (FFN) that independently predicts a class label (or a *no object* class) and normalized bounding box coordinates.

To train the network without NMS, DETR employs a bipartite matching loss based on the *Hungarian algorithm*. This algorithm finds the optimal one-to-one assignment between the fixed set of N predictions and the M ground-truth objects ($N > M$) by minimizing a matching cost that accounts for both class accuracy and bounding box overlap. This one-to-one assignment encourages the network to produce non-redundant predictions, removing the need for NMS.

Notably, this spatial overlap is typically evaluated using the *Generalized Intersection over Union (GIoU)* rather than standard IoU. This choice ensures a more robust loss function that provides continuous, non-zero gradients even when the predicted and target boxes do not intersect.

2.3.3 Vision-Language alignment

While traditional computer vision models are typically trained on a fixed set of predefined categorical labels, in an open-world setting this approach is impractical due to the high diversity of possible classes. *Vision-Language Models (VLMs)* address this bottleneck by mapping visual features directly into a continuous semantic text space. By aligning

images with a natural language vocabulary, these models enable zero-shot generalization to novel classes without requiring task-specific retraining.

The foundations of this paradigm were introduced by CLIP (Contrastive Language-Image Pre-training) [36]. CLIP utilizes two separate encoders: a *vision encoder* (typically a ViT) to process an image I_i , and a *text encoder* (a transformer) to process a descriptive caption T_i . Both encoders project their respective inputs into a shared, multi-modal latent space of a fixed dimension.

The architecture is trained on massive datasets of image-text pairs scraped from the internet using a *Symmetric Contrastive Loss*. Given a batch of B image-text pairs, the model computes the cosine similarity between all $B \times B$ possible pairings in the latent space. The loss function, as the NT-Xent loss [37], is designed to maximize the similarity of the B correct (positive) pairs along the diagonal of the similarity matrix, while simultaneously minimizing the similarity of the $B^2 - B$ incorrect (negative) pairs:

$$\mathcal{L}_i^{(I \rightarrow T)} = -\log \frac{\exp(\text{sim}(\mathbf{v}_i, \mathbf{t}_i)/\tau)}{\sum_{j=1}^B \exp(\text{sim}(\mathbf{v}_i, \mathbf{t}_j)/\tau)} \quad (2.5)$$

where $\mathbf{v}_i$ and $\mathbf{t}_i$ are the latent representations of the image and text, respectively, $\text{sim}(\cdot, \cdot)$ is the cosine similarity and τ is a learnable temperature parameter. A symmetric loss $\mathcal{L}^{(T \rightarrow I)}$ is computed similarly for the text-to-image direction.

By learning to associate groups of pixels with broad natural language descriptions rather than categorical IDs, these models construct a highly robust feature space. This semantic alignment is what enables zero-shot detectors to identify out-of-distribution instances through textual prompt conditioning.

Chapter 3

Related work

3.1 Open-vocabulary and promptable vision models

In recent years, computer vision has shifted toward large-scale pre-trained models that exhibit strong generalization capabilities across tasks. Within the computer vision domain, zero-shot object detection and promptable vision models have been introduced to identify or localize novel objects without requiring task-specific training. This represents a significant change from traditional supervised detectors, which are restricted to a pre-defined set of categories and often perform poorly when applied to out-of-distribution data.

To overcome these limitations in ecological monitoring, this thesis primarily adopts a zero-shot approach by leveraging open-vocabulary foundation models. Specifically, Grounding DINO and SAM3 were selected as the localization backbones of our pipeline, exploiting their text-prompting capabilities to localize targets in complex wild environments without requiring expensive domain-specific retraining.

Grounding DINO

Grounding DINO [11] is an open-set object detector developed through a grounded pre-training phase, designed to detect objects in an image using only natural language inputs, such as category names.

The architecture is fundamentally composed of two modality-specific encoders and one cross-modality decoder. The visual and linguistic inputs are processed independently by a Swin Transformer [38] and a BERT [39] model, respectively. Once the features are extracted, a feature enhancer module augments these representations by applying self-attention mechanisms to both feature spaces. Subsequently, building upon the paradigm introduced by GLIP [40], cross-attention modules are employed to align the vision and language spaces, generating fused cross-modality queries. Finally, a cross-modality decoder refines these queries to output the final bounding box and class predictions.

The model is available in two variants: *tiny* and *base*.

In the context of automated ecological monitoring, this model has already been tested, particularly for the detection of mammals [41] and bees [42] in camera trap images, and applied to beetles [43] to detect them against homogeneous backgrounds.

Note that more recent versions of the model (e.g., Grounding DINO 1.5 and DINO-X [44, 45]) have since been released, though these are only accessible through proprietary API calls rather than as open-weight models.

SAM3

SAM3 [3] is the latest major iteration of the Segment Anything Model family. Compared to previous versions, SAM3 introduces *Promptable Concept Segmentation (PCS)*, which enables the model to effectively generate segmentation masks over objects using only natural language prompts. Despite this advancement, it retains the core capabilities of its predecessors, such as generating masks from visual prompts like points, bounding boxes, or existing masks.

As illustrated in Figure 3.1, the model architecture incorporates several innovations while maintaining key components from the SAM2 [46] framework, particularly the modules dedicated to temporal object tracking across video frames. Furthermore, it integrates the Perception Encoder (PE) [47], a multi-modal architecture that is composed of several distinct modules. This encoder utilizes different modules depending on the input information: *PE Core* for processing visual data and *PE Language* for encoding text inputs.

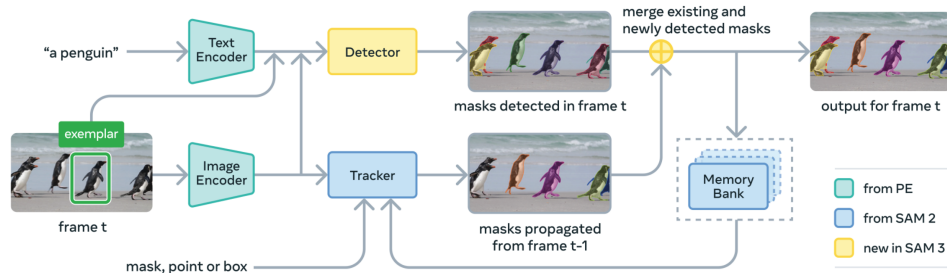


Figure 3.1: SAM3 architecture diagram [3].

The primary architectural innovation lies in the detector module and the subsequent fusion process that merges various representations to generate the output masks, with an improved tracking mechanism. An important distinction from Grounding DINO is that SAM3 is fundamentally a class-agnostic model and does not natively produce multi-class predictions in a single forward pass. Indeed, the decoder layer predicts a binary label of whether the object corresponds to the prompt or not.

In general, SAM3 achieves state-of-the-art performance on several tasks, including instance segmentation, semantic segmentation, object detection and object counting.

3.2 Visual representation learning

The advancement of self-supervised and contrastive learning has influenced the development of vision foundation models, which serve as robust feature extractors capable of mapping complex visual data into high-dimensional embedding spaces.

Two distinct architectures were selected for this project, representing different approaches to visual representation learning.

DINOv3

The DINO (DIstillation with NO labels) family of models [48, 49, 50] represents a significant breakthrough in self-supervised visual representation learning. These architectures leverage Vision Transformers (ViTs) to learn dense feature embeddings from massive amounts of unlabelled data, bypassing the dependency on manual annotations.

Building upon its predecessors, DINOv3 introduces advanced training paradigms to enhance the understanding of fine-grained semantics and spatial relationships. A relevant innovation is *Gram anchoring*, a regularization objective designed to mitigate the performance degradation on dense prediction tasks, such as segmentation, that typically occurs during extended training schedules. By enforcing patch-level consistency through the Gram matrix (the pairwise dot-product matrix of patch features), DINOv3 prevents the feature degradation that otherwise undermines dense tasks, ensuring that the model retains its spatial awareness even after millions of training iterations. This result is empirically demonstrated by the consistent performance on semantic segmentation benchmarks, such as ADE20K [51, 52] and Pascal VOC [53], where the model maintains a high mean IoU throughout extended training phases. Furthermore, this approach leverages high-resolution feature distillation to refine these spatial representations, making the model even more resilient to prolonged training protocols, still producing high-quality features.

Consequently, DINOv3 serves as a highly versatile foundation model, achieving state-of-the-art performance across diverse visual domains without the need for domain-specific fine-tuning.

For this project, the ViT-H/16 variant was selected, distilled from the ViT-7B architecture and pre-trained on the LVD-1689M dataset [50], a large collection of 1.7 billion images.

BioCLIP 2.5

In contrast to the general-purpose, purely visual approach of DINOv3, BioCLIP 2.5 [54] has been selected as a domain-specific foundation model explicitly developed for biological and ecological images. Building upon the contrastive vision-language framework of the original CLIP architecture, BioCLIP 2 [55] demonstrated that models trained at scale

on biological data exhibit remarkable emergent behaviors beyond their initial training objectives.

The latest iteration, BioCLIP 2.5, significantly scales this framework. Available in a single large variant, it utilizes a Vision Transformer backbone (ViT-H/14) pre-trained on a subset of LAION-2B [56] and fine-tuned on an expanded version of the TreeOfLife-200M dataset [55], which comprises over 233 million images of living organisms. Through large-scale contrastive vision-language training, the model maps visual features and taxonomic metadata into a shared semantic space.

An important finding from BioCLIP is the emergence of a biologically meaningful embedding space. Despite the primary objective of species distinction, the model organizes its latent space to align with functional and ecological meanings. At the inter-species level, the embeddings naturally cluster according to traits and habitats, while intra-species variations (such as life stages and sexes) are preserved in orthogonal subspaces. This structure allows the model to capture complex ecological relationships and reach state-of-the-art performance also on broader ecological tasks.

3.3 Supervised baselines

To establish a comprehensive understanding of the state-of-the-art in automated insect monitoring, it is essential to analyze the performance of supervised models specifically trained for this domain. This analysis serves to quantify the performance gap between specialized, task-specific architectures and general-purpose foundation models, highlighting the trade-offs between manual labeling requirements and model generalization capabilities.

The following sections provide a comprehensive description of the selected supervised models, ordered chronologically from their respective publication dates.

Bjerge et al.

Developed by Bjerge et al. [25], this multi-class detector utilizes the YOLOv5 architecture and was released in conjunction with a comprehensive dataset [57] for pollinator monitoring. The model’s optimal configuration was determined through a rigorous ablation study that evaluated several key architectural and training parameters, including:

- **Architecture version:** A comparison between YOLOv3 and YOLOv5
- **Class granularity:** The models were trained and evaluated using both a 6-class training subset and the full 9-class dataset

- **Input resolution:** Evaluation of training image sizes, specifically 640×640 and 1280×1280 pixels
- **Model complexity:** Testing of different model scales, specifically the *small* and *medium* variants
- **Early stopping criteria:** Comparison between F1-score and mAP as metrics to trigger early stopping after 100 epochs without performance gains

The study concluded that optimal performance was achieved by the medium-sized YOLOv5 model trained at the higher resolution (1280 pixels), utilizing the F1-score on the validation set as early stopping metric.

When evaluated on a test set containing OOD classes not present in the training data, the model correctly detected approximately 80% of all insects. This detection rate increased to 93.8% when only IDD classes were considered, highlighting the challenges posed by domain shift for specialized supervised models.

Stark et al.

Published in late 2023 [58], this study evaluated multiple YOLO architectures, specifically YOLOv5-nano, YOLOv5-small, and YOLOv7-tiny. The models were trained on a large-scale dataset comprising more than 17,000 citizen science images sourced from *iNaturalist* and *Observation.org*. These images are characterized by “wild” backgrounds, but generally exhibit higher visual quality compared to field images captured by automated camera traps.

The models were designed to detect and classify eight distinct groups of insects: *Hymenoptera*, *Diptera*, *Coleoptera*, *Lepidoptera*, *Orthoptera*, *Hemiptera*, *Araneae*, and the *Formicidae* family. The evaluation phase was conducted using standard object detection metrics, with a specific focus on performance variations across different IoU thresholds. When tested on IDD images, the recall ranged from 83% to 89%, consistent with findings in similar supervised monitoring studies.

In a subsequent study, these models were tested on more than 23,000 field images captured using smartphones [59] to assess OOD performance on a real-world dataset. The YOLOv5-small variant emerged as the best-performing model based on the F1-score, achieving 70% using an NMS IoU threshold of 0.3. Specifically, the study showed that for non-crowded scenes, adopting a lower IoU threshold of 0.1 instead of the standard 0.5, offered slight performance improvements and generated predictions of acceptable biological quality, especially by reducing false positives per image.

Marin et al.

This study [27] evaluates the efficacy of an Automated Camera System (ACS) for pollinator monitoring, comparing automated detection results with direct human observations.

During the experimental phase, six distinct YOLOv5 models were trained, on a binary insect detection task, using consistent hyperparameters but varying dataset configurations. These configurations included a combination of diurnal and nocturnal images, as well as data captured by different sensors characterized by varying image quality. In addition, the study explored the impact of a sliced inference technique (SAHI, see Section 2.2.3) on detection performance.

The optimal model, selected based on its F1-score, was trained on a heterogeneous dataset including both day/night and high/low-quality images, achieving precision, recall, and F1-score of 95%. Interestingly, in this specific setup, the application of sliced inference led to a performance degradation, reducing the F1-score by approximately 10 percentage points.

Finally, to assess the Out-of-Distribution (OOD) robustness of the best-performing model, a leave-one-out cross-validation experiment was conducted across different plant species. This evaluation revealed a significant performance gap, with an average F1-score of 0.624 ± 0.160 . Such a high standard deviation highlights the model’s difficulty in generalizing across diverse backgrounds, emphasizing the limitations of supervised approaches when faced with domain shifts.

FlatBug

FlatBug [9] is a specialized YOLOv8-seg architecture developed to detect and segment insects against homogeneous backgrounds. A key feature of the framework is its integrated *pyramid-slicing* inference mechanism, a multi-scale variant of SAHI, designed to accurately localize small insects even within crowded scenes.

Trained on a large-scale ensemble of 15 datasets comprising more than 113,000 individual instances, the model achieved a 94.2% F1-score on its IDD test set. In particular, it demonstrated robustness to domain shifts, with a performance decrease of only 7.1% when evaluated on OOD images.

Consistent with the monitoring tools mentioned previously, the default NMS IoU threshold is set at 0.2. This further underscores the emerging consensus in automated insect monitoring that lower thresholds are preferable to manage detection sensitivity in complex scenarios.

The evaluation of FlatBug on datasets involved in this project represents a substantial domain shift. Nevertheless, it might provide valuable insights into the generalization

capabilities and limitations of this domain-specific model.

Chapter 4

Project

Building upon the technological and methodological challenges discussed in the previous chapters, this project explores the potential of recent computer vision advancements for automated insect monitoring. The main purpose of this study is to determine if modern foundation models can successfully detect insects across diverse, real-world time-lapse monitoring contexts. By benchmarking state-of-the-art zero-shot architectures, specifically Grounding DINO and SAM3, this project aims to address several key research questions.

First, it investigates whether zero-shot foundation models are inherently more robust to domain shifts than custom-trained supervised alternatives. Second, it evaluates the extent to which different prompting strategies influence the models’ detection accuracy.

Finally, the study explores whether false positives can be effectively pruned using an additional supervised classification step. Within this verification stage, a critical sub-question is to determine whether domain-specific visual representations (e.g., BioCLIP 2.5) outperform general-purpose foundation models (e.g., DINOv3) in accurately discriminating insects from background crops.

4.1 Datasets and data preparation

To conduct the experiments proposed in this research, several datasets, both published and unpublished, were curated and standardized. Rather than relying on a single data source, this work uses multiple image collections, each with different characteristics, serving a specific methodological purpose. This diversity facilitates the evaluation of both supervised baselines and zero-shot object detection models on multiple Out-of-Distribution (OOD) datasets. These test sets introduce various intrinsic and extrinsic domain shifts, such as diverse background environments, different orders of insects, and variations in both image resolution and camera-to-insect distance.

4.1.1 Ground ECOSTACK dataset

This collection consists of images captured by automated camera traps monitoring pitfall traps deployed in agricultural environments, specifically rapeseed and winter wheat fields located in the UK, during June and July 2020. At the time of writing, this dataset is not publicly available.

The dataset comprises a total of 12,276 images. In the training, validation, and test partitions, 40% of the data consists of negative samples (i.e., background images that do not contain insects). Furthermore, this dataset features high-resolution imagery, primarily at 4224×2210 pixels, with a subset extending up to 6080×3420 pixels.

As detailed in Table 4.1, the labeling of insect classes is approximately at the order level, and the distribution of instances within this specific dataset is highly heterogeneous. It is heavily dominated by *Diptera* (flies) and *Hymenoptera* (specifically bees and wasps) and ground beetles such as *Carabidae* and *Staphylinidae*, with a long-tail distribution where several classes are strongly under-represented.

Taxonomic Group	Train	Validation	Test	Total
Diptera-Hymenoptera	2282	492	482	3256
Carabidae	1836	389	390	2615
Staphylinidae	673	143	146	962
Araneae	390	80	81	551
Myriapoda	227	52	48	327
Coleoptera (larvae)	101	23	22	146
Isopoda	43	9	10	62
Coccinellidae	35	9	7	51
Opiliones	33	7	8	48
Dermaptera	33	7	6	46
Cantharidae	1	1	1	3
Total	5654	1212	1201	8067

Table 4.1: Distribution of annotated instances across dataset splits in the Ground ECOSTACK dataset

Examples of images from the test set of this dataset are shown in Figure 4.1, illustrating the diversity of backgrounds and insect sizes encountered.

4.1.2 Bjerger et al. dataset

This dataset [57, 25] was originally developed to train and evaluate a multi-class YOLOv5 detection model. While initially compiled through a citizen science initiative,

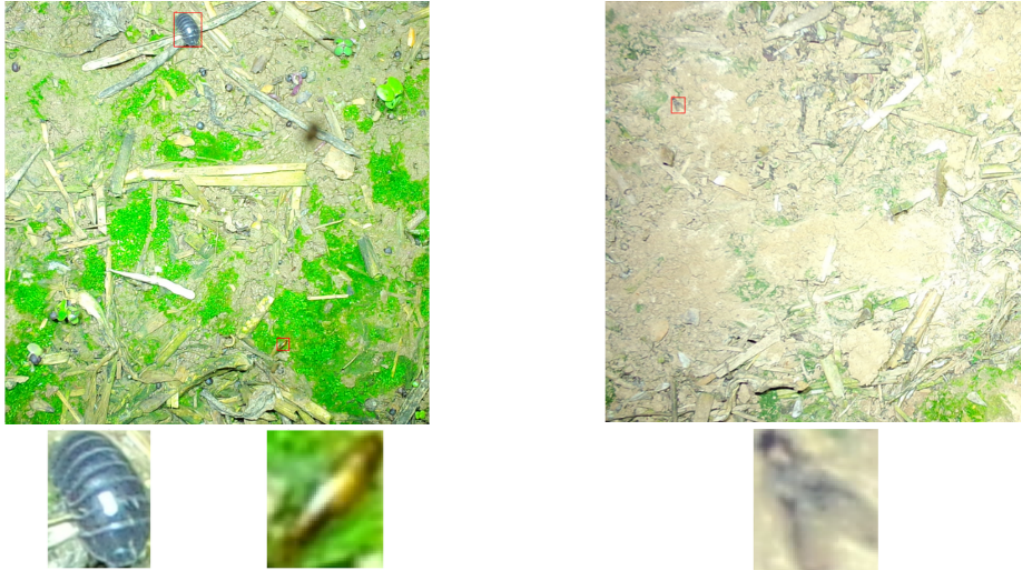


Figure 4.1: Sample images from the Ground ECOSTACK dataset test split (crop of the original images).

the dataset’s annotations were subsequently subjected to a revision process to correct labeling errors or inaccuracies, establishing a high-quality ground truth.

The image data was acquired through an array of ten automated cameras deployed on a green roof. The recording scene features dense vegetation that undergoes significant visual changes during the capturing period. This introduces a highly dynamic background that, together with a reasonable insect-camera distance, makes the dataset an ideal benchmark to evaluate the robustness of detection models against complex natural environments.

The instance distribution, detailed in Table 4.2, is heavily dominated by bees, bumblebees, and wasps (*Diptera-Hymenoptera*). Within the *Coleoptera* order, the dataset exclusively features ladybugs, contrasting with the ground beetles observed in the Ground ECOSTACK dataset. *Lepidoptera* (butterflies) are also present, although in significantly lower numbers.

Taxonomic group	Train	Validation	Test	Total
Diptera-Hymenoptera	17398	2931	6967	27296
Coleoptera	6344	299	399	7042
Lepidoptera	286	27	31	344
Other insects	0	0	1502	1502
Total	24028	3257	8899	36184

Table 4.2: Distribution of annotated instances across the Bjerge et al. dataset splits

Sample images from the test set of this dataset are shown in Figure 4.2, illustrating the dynamic background and the presence of multiple insects in the same scene.



Figure 4.2: Sample images from the Bjerge et al. dataset test split (crop of the original images).

4.1.3 RangeX dataset

This dataset comprises still images captured by time-lapse cameras focused on individual flowers to monitor pollinator behavior. Unlike the aforementioned datasets, the insects appear larger due to a center-crop acquisition strategy applied to the original frames. Despite this increased effective resolution, the inclusion of minute species, such as ants, set against highly variable backgrounds, ensures that the dataset remains a significant challenge for small-object localization. The collection consists of 2,561 images, with background-only frames accounting for approximately 50% of the samples. All images are provided at a 1024×1024 pixel resolution. At the time of writing, this dataset is not publicly available.

A particular trait of this dataset is the presence of rich metadata, including several attributes relevant to computer vision task evaluation, such as:

- *flash*: a boolean indicating if the camera’s flash was active during capture
- *flower_species*: the taxonomic abbreviated name of the monitored flower
- *focus*: a manually annotated attribute with values `clear`, `acceptable`, or `blur`
- *scalingFactor*: a manually annotated float that scales with the average dimensions of a bumblebee in that scene; it serves as a proxy for the insect-camera distance
- *lifeStage*: a string that annotates the life stage of the detected insect

To provide a robust evaluation, the collection of images was partitioned into validation (calibration) and test splits, reserving 20% of the data for validation. A stratified split was performed based on the `scene_id` attribute, ensuring that every monitoring setup is adequately represented in both the threshold calibration and evaluation phases. This design distributes all scenes and environmental conditions evenly across both splits, reflecting an in-domain deployment scenario. Sample images from the test set are shown in Figure 4.3.

4.1.4 Ştefan et al. dataset

This collection of images contains time-lapse captures of flower-visiting pollinators, automatically recorded using smartphones from July to September 2021 in urban green spaces in Germany. The detailed methodology of data collection is provided in Ştefan et al. [60]. For these observations, smartphones were positioned above selected open flowers of 33 plant species. The smartphones captured time-lapse images at an average rate of approximately one frame every second for an average session duration of approximately

Taxonomic group	Validation	Test	Total
Bumblebee	92	367	459
Syrphid	85	313	398
Diptera	63	250	313
Lepidoptera	47	166	213
Bee	14	78	92
Ant	11	49	60
Coleoptera	11	29	40
Hymenoptera	2	19	21
Total	325	1271	1596

Table 4.3: Distribution of annotated instances across the RangeX dataset splits.



Figure 4.3: Sample images from the RangeX dataset test split.

one hour on a targeted flower after which the smartphones were relocated to different flowers.

The version of this dataset used in this project contains 23,899 images and 24,838 bounding boxes; images are also cropped at the Region of Interest (ROI) to include only the flower area, thus relaxing the problem onto a smaller field of view. Because the dataset comes without any split, a random 20% of the data was extracted to form a validation set, ensuring that the species distribution is preserved across the splits. Because of the ROI cropping, images in the dataset have heterogeneous resolutions. The dataset is publicly available. Sample images from the test set of this dataset are shown in Figure 4.4.

Taxonomic group	Validation	Test	Total
Hymenoptera	2724	10688	13412
Diptera	986	4039	5025
Coleoptera	543	2238	2781
Hymenoptera (Formicidae)	400	1580	1980
Araneae	214	822	1036
Hemiptera	111	492	603
Lepidoptera	0	1	1
Total	4978	19860	24838

Table 4.4: Distribution of annotated instances across the Ștefan et al. dataset splits.

4.1.5 Dataset annotations normalization

Given that the datasets utilized in this study originated from independent research projects, the initial annotations were provided in heterogeneous formats, ranging from custom CSV schemas (as in the RangeX dataset) to YOLO-style text files (as in the Bjerger et al. dataset). To ensure interoperability across the entire evaluation pipeline, a comprehensive normalization process, different for each dataset, was implemented.

All annotations were converted to the Common Objects in Context (COCO) format [61]. This standard was selected to streamline downstream analysis, as it is fully compatible with the `pycocotools` library [62]. The use of the COCO evaluation API provides a unified and robust framework for computing standard metrics, such as Average Precision (AP) and Average Recall (AR), ensuring consistency across all evaluated architectures and baseline models.

4.2 Inference and evaluation framework

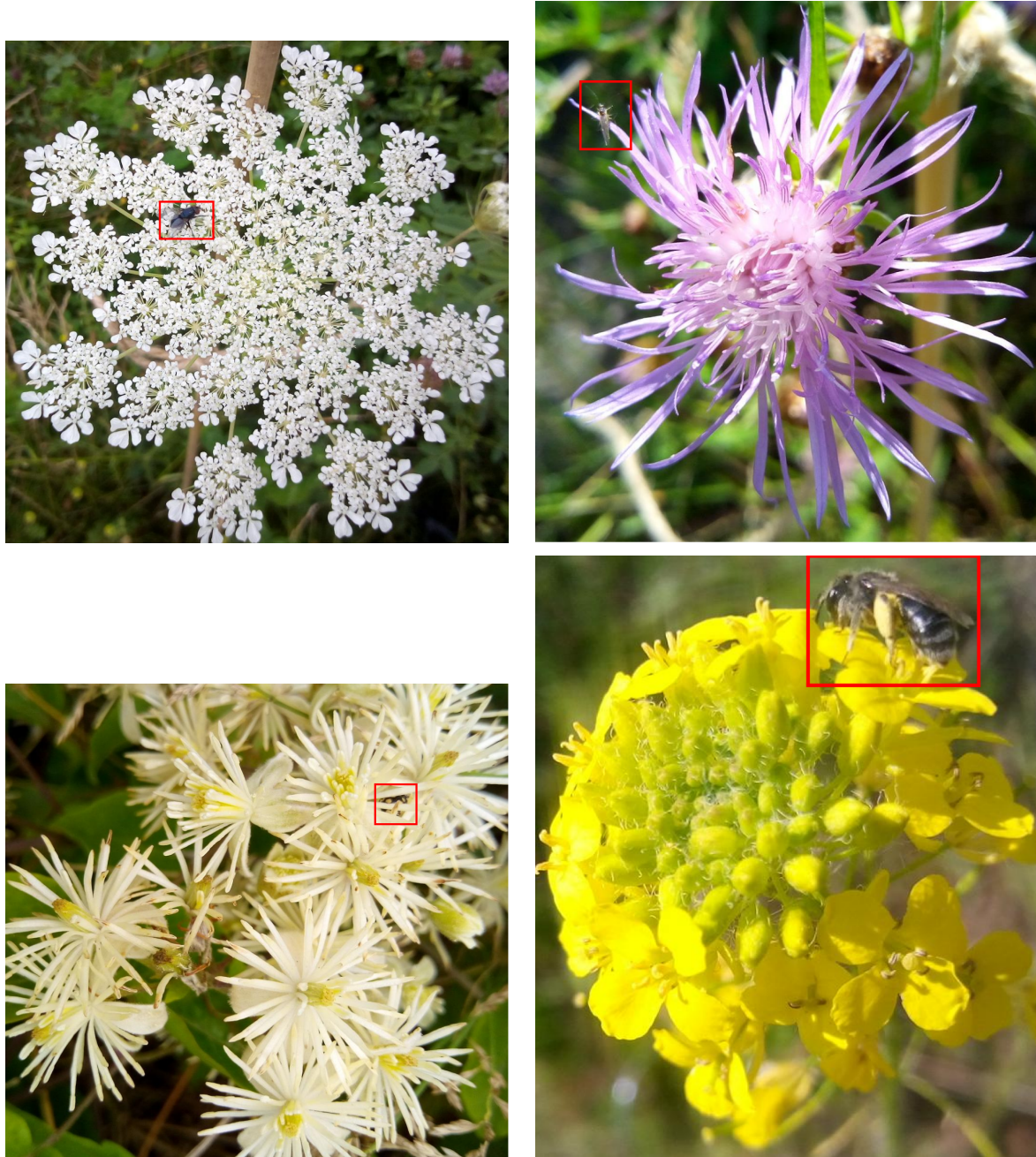


Figure 4.4: Sample images from the Ștefan et al. test split.

4.2.1 Implementation details

All experiments were executed on a dedicated workstation running Ubuntu 24.04.4 LTS. The hardware configuration includes an Intel Core Ultra 9 285K processor accompanied by 128 GB of system memory. To handle the computational load of the vision foundation models, the system is equipped with a single NVIDIA GeForce RTX 5090 GPU featuring 32 GB of VRAM. The software environment relies on the CUDA version 13.1 for GPU acceleration. All models and processing scripts were implemented in `Python`, using a `venv` virtual environment to manage library dependencies and ensure the reproducibility of the experimental setup.

4.2.2 Tiled inference strategy

Given the high resolution of the source images and the small size of the insects compared to the total area of the image pixels, a custom tiled inference strategy was essential to maintain detection sensitivity. This approach ensures that small instances are not downsampled to a point where their morphological features become indistinguishable for the model’s feature extractor. For instance, in the Ground ECOSTACK dataset, down-sampling a 6K image to a 1024×1024 pixel square would result in a significant loss of information, making the model unable to distinguish small insects from complex backgrounds characterized by grasses, sticks, and camouflaged textures.

Furthermore, this strategy is designed to exploit the full potential of the selected architectures by aligning the input tiles with their respective native training resolutions, which span from 640×640 to 1024×1024 pixels, avoiding significant upscaling or downscaling of the original image.

The implementation follows a logic similar to the SAHI framework (described in Section 2.2.3). The original image is partitioned into overlapping tiles, where the tile size is dynamically adjusted to match the training resolution of the specific model being evaluated. To prevent truncation of insects located over the tile boundaries, an overlap ratio of 20% was maintained across both axes.

The final step involves a standardizing NMS operation in order to merge redundant detections from overlapping regions. Considering the biological nature of these datasets, where insects are typically spatially isolated rather than clustered in dense crowds, a common IoU threshold of 0.1 was selected. This aggressive low threshold effectively suppresses duplicate bounding boxes while minimizing the risk of merging distinct individuals.

4.2.3 Evaluation metrics

Evaluating an object detection model is inherently more complex than evaluating a standard image classifier, as the network must be assessed on two simultaneous tasks: the accuracy of the predicted class and the spatial precision of the predicted bounding box. This section formalizes the core metrics used in object detection and details the specific experimental setup adopted in this study.

Spatial overlap and matching

The principal metric for evaluating spatial localization is the *Intersection over Union* (*IoU*), also known as the Jaccard index (or Jaccard similarity). Given a predicted bounding box B_{pred} and a ground-truth bounding box B_{gt} , the IoU measures the ratio of their overlap to their combined total area:

$$\text{IoU} = \frac{\text{Area}(B_{pred} \cap B_{gt})}{\text{Area}(B_{pred} \cup B_{gt})} \quad (4.1)$$

To determine whether a prediction is correct, depending on the specific application, a predefined IoU threshold (τ_{IoU}) is established (commonly 0.50). Based on this threshold and the predicted class label, each detection is categorized into one of three states:

- **True Positive (TP):** The predicted class matches the ground-truth class, and $\text{IoU} \geq \tau_{\text{IoU}}$.
- **False Positive (FP):** A detection with $\text{IoU} < \tau_{\text{IoU}}$, a duplicate detection for the same ground-truth object, or a detection with an incorrect class label.
- **False Negative (FN):** A ground-truth object that the model failed to detect, or one that was detected with an IoU strictly below the threshold.

Note that in object detection, there is no explicit concept of a *True Negative (TN)*, as the background space consists of infinite potential bounding boxes that the model correctly ignores.

Precision, Recall, and F1-score

Using the accumulation of TPs, FPs, and FNs across an entire evaluation dataset, we can define the standard classification metrics:

Precision (P) measures the exactness of the model. It represents the proportion of predictions that are actually correct over all predictions made:

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4.2)$$

A high precision indicates that when the model detects an object, it is highly likely to be a real object and not a background crop.

Recall (R) measures the completeness of the model. It represents the proportion of actual ground-truth objects that were successfully detected:

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4.3)$$

To provide a single metric that balances both precision and recall, the **F1-score** is computed as the harmonic mean of the two:

$$\text{F1} = 2 \cdot \frac{P \cdot R}{P + R} \quad (4.4)$$

Average Precision (AP) and Average Recall (AR)

While precision and recall are useful at a specific confidence threshold, object detectors output a continuous confidence score $p_i \in [0, 1]$ for each prediction. By sorting all predictions in descending order of confidence and incrementally lowering the acceptance threshold, we can compute precision and recall at every possible rank. Plotting these values generates a *Precision-Recall (PR) Curve*.

The *Average Precision (AP)* summarizes the shape of this curve by computing the Area Under the Curve (AUC). Theoretically, this is defined as the integral of precision P as a function of recall R :

$$\text{AP} = \int_0^1 P(R) dR \quad (4.5)$$

However, because the PR curve is generated from discrete predictions and is inherently stepwise, standard evaluation protocols (such as COCO) compute AP using an interpolation method. The recall range $[0, 1]$ is divided into a set of evenly spaced points (e.g., 101 points in the COCO protocol), and the AP is calculated as the discrete average of the interpolated maximum precision at each recall level r :

$$P_{\text{interp}}(r) = \max_{\hat{r} \geq r} P(\hat{r}) \quad (4.6)$$

$$\text{AP} = \frac{1}{101} \sum_{r \in \{0, 0.01, \dots, 1\}} P_{\text{interp}}(r) \quad (4.7)$$

This interpolation ensures that the metric is robust to small variations in the ranking of predictions.

When evaluating a multi-class dataset, the AP is calculated independently for each class $c \in \mathcal{C}$, and the results are averaged. Historically termed *mean Average Precision*

(mAP), in a single-class detection scenario, AP will be equivalent to mAP , as the average is taken over a single class.

To capture a more rigorous evaluation of spatial accuracy, the standard COCO metrics compute the AP across 10 linearly spaced IoU thresholds ranging from 0.50 to 0.95 (denoted as $AP_{0.50:0.95}$). Specific benchmarks like AP_{50} (IoU = 0.50) evaluate loose localization, while AP_{95} evaluates strict spatial overlap.

Similarly, *Average Recall (AR)* measures the maximum recall achievable when restricting the detector to a fixed number of predictions per image (e.g., $AR^{\max=10}$ or $AR^{\max=100}$). Following the COCO protocol, recall is computed at multiple IoU thresholds ranging from 0.50 to 0.95 and then averaged. AR therefore provides a robust estimate of the detector’s ability to find objects in the image, independently of confidence calibration.

Experimental setup

As discussed in Section 4.1.5, the evaluation phase was conducted leveraging the COCO annotation format in conjunction with the standard `pycocotools` API. To provide a broader assessment of the experimental results, Mean Average Precision (mAP) and Mean Average Recall (mAR) were calculated at two different IoU thresholds: 0.1 and 0.5.

The selection of a low IoU threshold of 0.1 is intended to evaluate the detection capabilities of the models, specifically their ability to correctly identify the presence of insects within the frame, even if the predicted bounding boxes do not perfectly align with the ground-truth. This is particularly relevant for zero-shot vision foundation models, which are not explicitly trained on this particular domain and might exhibit slight localization offsets, for example skipping legs or antennae of the insects when the camera is further from the insect, still retaining high semantic confidence. It also reflects the real life priority of the task, which is to find small objects in large volumes of imagery, not to set perfect bounding boxes.

On the other hand, the higher threshold of 0.5 provides a more stringent measure of effective localization and boundary precision, although it may penalize true positives that fail to meet this stricter box overlap requirement.

Analyzing the performance at both thresholds allows for a comprehensive understanding of the models’ capabilities. A significant performance drop from 0.1 to 0.5 indicates that while the model can detect the presence of insects, it struggles with precise localization, which is a common challenge in zero-shot detection scenarios.

In addition to the standard COCO metrics, the *F1-score*, along with its relative *precision* and *recall*, was computed for each IoU threshold. To ensure a fair and robust evaluation, the confidence threshold for these metrics was determined by identifying the value that maximized the F1-score on the validation set for each model. In particular, the F1-score calculated at an IoU of 0.1 was established as the primary metric to identify the best model across the different experimental configurations.

4.2.4 Prompt design

To ensure a robust assessment and fair comparison between the two architectures, an ablation study was conducted using different prompting strategies. These strategies were designed to evaluate how semantic specificity and contextual information influence zero-shot detection performance.

The following main prompt configurations were tested:

- **Basic prompt** (P1), a generic “*insect*” query used as a baseline to evaluate the most basic zero-shot localization capabilities.
- **Contextual prompt** (P2, SAM3 exclusive), specific descriptive phrases such as “*an insect on a natural background, an insect on a flower, an insect on the ground*”. This was tested only for SAM3 to leverage its contextual perception, whereas Grounding DINO’s architecture is primarily optimized for discrete category labels.
- **Taxonomic prompt** (P3), a comprehensive list of target categories: “*beetle, fly, bee, ant, wasp, butterfly, moth, spider, bug, dragonfly, earwig, millipede, pillbug*”. This prompt has been constrained to respect the 32-token context limit inherent to the SAM3 architecture, while ensuring coverage of the vast majority of insects present in the evaluated datasets.

By using the same taxonomic list for both models, a standardized benchmark is established to quantify the gap between class-aware (Grounding DINO) and class-agnostic (SAM3) localization.

Negative prompting

Negative prompting was evaluated, specifically for Grounding DINO, to mitigate false positives in datasets where a non-target object consistently dominates the foreground. This technique leverages the model’s class-aware output to identify and subsequently discard detections belonging to unwanted categories.

In this study, negative prompting was applied primarily to the *RangeX* dataset. Given that these images are characterized by a close-up view where a flower occupies a significant portion of the frame alongside the insect, “*flower*” was added as an additional category to both the simple and taxonomic prompts. During post-processing, any bounding box predicted with the “*flower*” label was automatically discarded.

For the remaining datasets, this technique was not implemented due to the larger camera-to-subject distance, which results in a more heterogeneous background with multiple diverse objects. Indeed, negative prompting requires a custom-tuned prompt based on the specific visual distractions of the domain.

4.2.5 Zero-shot detectors hyperparameters

During the inference phase, the zero-shot detectors were evaluated using a permissive confidence threshold of 0.1. This setting was chosen to ensure adequate recall, ensuring that the vast majority of potential insect detections are captured, even with low confidence scores prior to subsequent pruning/refinement stages. This approach is particularly advantageous given that standard COCO evaluation metrics compute performance by considering also the confidence scores, thus allowing for a more comprehensive analysis.

Specifically for Grounding DINO, the *text confidence threshold* was lowered to 0.1 from the default 0.25. This allows for broader and less restrictive matches between the textual prompts and the image features. This adjustment has been chosen after preliminary empirical explorations on the validation sets, which indicated that the default threshold was too stringent for the zero-shot nature of the task, leading to a significant number of false negatives.

4.2.6 Proposed ensembling strategy

In addition to the individual evaluation of each prompt configuration, an ensembling strategy was proposed to further enhance detection performance and mitigate the potential sensitivity or weaknesses of specific prompts. This can be potentially useful in production environments, where the best prompt configuration may not be known a priori, and might vary across different deployments.

This approach involves an offline aggregation of spatial predictions across the N different prompt configurations from the same model. The ensembling process relies on a greedy clustering algorithm to group detections that refer to the same physical object. Specifically, predictions are sorted in descending order based on their confidence scores. Iteratively, the highest-scoring unassigned bounding box acts as the *reference box* for a cluster, and any other box demonstrating a spatial overlap, measured via Intersection over Union (IoU), greater than a predefined threshold ($\text{IoU} \geq 0.1$) is assigned to its cluster, denoted as C .

Once a cluster C is formed, the predictions are fused to produce a single, more robust detection. Let $K = |C|$ be the number of overlapping boxes in the cluster (where $1 \leq K \leq N$). For each individual detection $i \in C$, let B_i represent its bounding box coordinates and S_i its confidence score. The ensembled bounding box coordinates $\hat{B}$ are computed as the spatial arithmetic mean of the constituent boxes:

$$\hat{B} = \frac{1}{K} \sum_{i=1}^K B_i \quad (4.8)$$

To calculate the fused confidence score $\hat{S}$, the sum of the cluster’s scores is averaged over the total number of prompt configurations N , rather than just the number of matches K :

$$\hat{S} = \frac{1}{N} \sum_{i=1}^K S_i \quad (4.9)$$

This formulation introduces a voting penalization mechanism. If a predicted object is highly consistent and detected by all prompt configurations ($K = N$), its final confidence score remains high, effectively rewarding stable detections. However, if a bounding box is generated by only a few prompts ($K < N$), the missing predictions implicitly contribute a score of zero, penalizing the final ensembled score.

4.3 Expert binary discriminator

In the context of automated ecological monitoring, a recent study [63] has demonstrated that the latent representations extracted by such self-supervised foundation models are quite separable even in complex domains. Within this project, DINOv3 and BioCLIP 2.5 are deployed as the visual backbones for the binary verification stage. By extracting high-dimensional embeddings from the bounding box crops generated by the zero-shot detectors, these backbones enable the downstream classifiers to effectively differentiate between the features of true insects and the textures of the natural background, providing a robust mechanism for false-positive pruning.

A similar approach has already been explored [64]. In that work, a training-free binary verification module is proposed, using a Vision Language Model (VLM) to verify that the output bounding boxes of a zero-shot detector are consistent with the textual prompt, allowing the model to answer only with a binary yes/no answer. In this project, however, the use of a domain-specific expert binary discriminator is explored, trained on specific insect and background crops.

4.3.1 Feature space analysis and dimensionality reduction

The embeddings extracted by DINOv3 and BioCLIP 2.5 reside in high-dimensional manifold spaces, which, even if rich in semantic and morphological information, are inherently difficult to interpret directly. To assess the quality of these representations, t-Distributed Stochastic Neighbor Embedding (t-SNE) was applied as a dimensionality reduction technique. This method serves to visualize high-dimensional clusters in a human-interpretable 2D space, and is useful for validating the hypothesis regarding the

linear separability between insect and background detections.

The t-SNE visualizations, illustrated in Figures 4.5 and 4.6, reveal the underlying structure of the feature space across different datasets. The emergence of distinct, well-defined clusters for the *Insect* and *Background* classes confirms the effectiveness of these foundation models in capturing discriminative features. Interestingly, the presence of different sub-clusters within the main categories, particularly within the insect class, indicates that the embeddings capture more than just a binary distinction, likely reflecting diverse taxonomic orders. This granularity further supports the robustness of these representations for a possible downstream classification task.

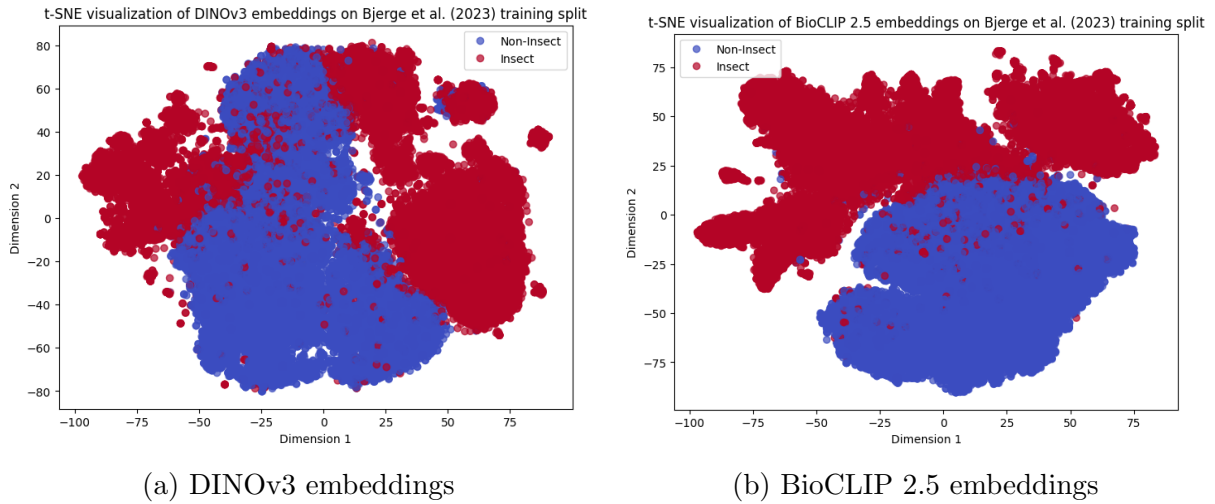


Figure 4.5: t-SNE visualizations for the Bjerge et al. training split using DINOv3 and BioCLIP 2.5 embedding vectors.

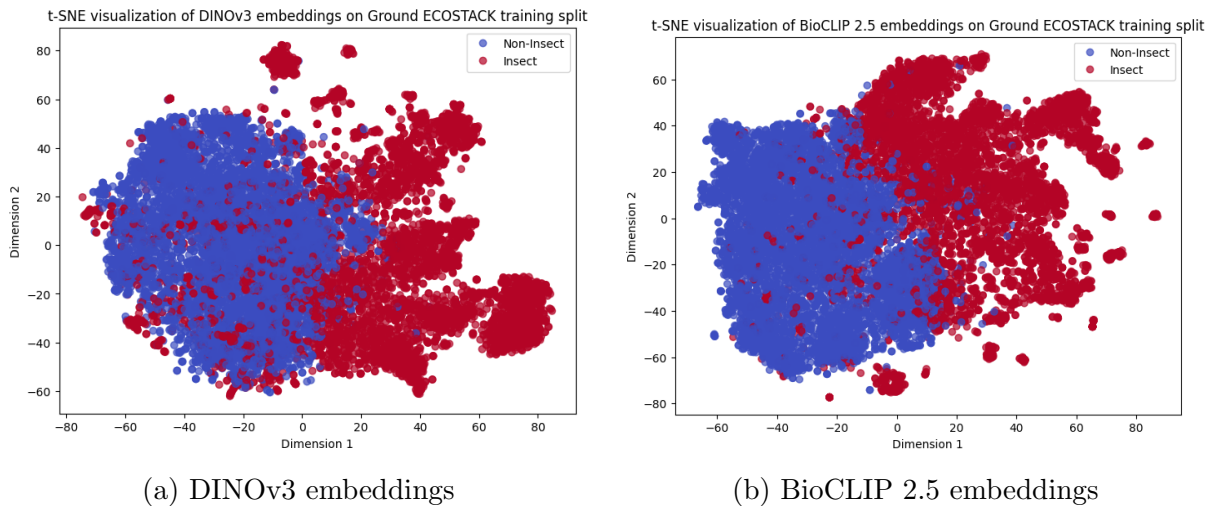


Figure 4.6: t-SNE visualizations for the Ground ECOSTACK training split using DINOv3 and BioCLIP 2.5 embedding vectors.

However, it is important to highlight that dimensionality reduction is utilized exclusively for preliminary data analysis. The final binary classifiers are trained on the original,

high-dimensional embeddings. This approach ensures that the discriminators can leverage the full set of latent information and subtle feature correlations that might be lost during the projection into a lower-dimensional space.

4.3.2 Expert classifier training

The expert binary classifiers are trained using the high-dimensional embeddings extracted from the labeled insect crops within the training sets of both the Bjerger et al. (Section 4.1.2) and Ground ECOSTACK (Section 4.1.1) datasets. Regarding the background class, during the crop extraction process, for each insect, a random, non-overlapping background crop of identical dimensions is extracted from the same image. This ensures that the background samples are representative of the natural environment in which insects are detected, and that a balanced dataset is created for the binary classification task.

During the crop and extraction process, labels and embeddings are stored in *npz* files, enabling an efficient storage and retrieval of the high-dimensional information during the training phase.

Overall, the extraction process yields the following distribution of samples, mapped to their respective source datasets:

- *Training set*: 59,358 samples (29,679 insect, 29,679 background), extracted from the training splits of the Ground ECOSTACK and Bjerger et al. datasets.
- *Validation set*: 8,938 samples (4,469 insect, 4,469 background), extracted from the validation splits of the Ground ECOSTACK and Bjerger et al. datasets.
- *Test set (In-Distribution)*: 20,200 samples (10,100 insect, 10,100 background), extracted from the test splits of the Ground ECOSTACK and Bjerger et al. datasets.
- *Test set (Out-of-Distribution)*: 50,563 samples (24,130 insect, 26,433 background), extracted from the test splits of the completely unseen RangeX and Stefan et al. datasets.

For this binary classification task, two architectures are explored: a simple Multi-Layer Perceptron (MLP) and a Support Vector Machine (SVM).

Multi-Layer Perceptron (MLP)

The MLP architecture comprises three hidden layers, each followed by Batch Normalization and ReLU activation. The network employs a bottleneck design, where the input dimensionality is reduced to 512 in the first layer, and subsequently halved in each successive hidden layer. Dropout ($p = 0.5$) is applied after the first hidden layer for regularization. The model is trained for 50 epochs with a batch size of 64 using the AdamW

optimizer (learning rate $1e^{-3}$, weight decay $1e^{-4}$). To account for class imbalance within the training set, Binary Cross Entropy with Logits loss function (`BCEWithLogitsLoss`) is used, weighted by the ratio of negative to positive samples. Even if the dataset is balanced, this choice is made to ensure that the model does not develop a bias towards one class during training if the distribution of samples changes in future iterations of the dataset or in real-world applications.

Support Vector Machine (SVM)

For this SVM implementation, a linear kernel was selected, following the hypothesis that the embeddings extracted by DINOv3 and BioCLIP 2.5 are already linearly separable. To ensure the robustness of the model, the following architectural choices were made: feature scaling is integrated into the pipeline through a `StandardScaler` to normalize the high-dimensional embeddings, preventing features with larger scales from dominating the optimization process. Given the 50/50 split between insects and background crops in the training dataset, a `balanced` class weight strategy is applied.

Since standard Linear SVMs do not inherently provide probability estimates, a `CalibratedClassifierCV` is employed using a *sigmoid* method, with calibration performed through a 5-fold cross-validation scheme to output confidence scores rather than hard labels. Finally, the model is trained with a maximum of 3000 iterations to ensure convergence in the high-dimensional latent space.

4.3.3 Scoring functions

The pipeline integrates zero-shot detection with expert binary verification, requiring a scoring function to aggregate confidence scores from both components. The detector provides an initial confidence score for each candidate region, while the classifier (MLP or SVM) outputs a probability representing the likelihood that the crop contains an insect rather than background. Two different aggregation strategies have been evaluated: a joint (multiplicative) scoring function and a mean (additive) scoring function.

Joint scoring function

$$\text{Score}_{\text{joint}} = \text{Score}_{\text{det}} \cdot \text{Score}_{\text{cls}} \quad (4.10)$$

This function leverages the strengths of both models by requiring high confidence from both components. By multiplying the scores, only detections that are both identified as potential candidates by the detector and classified as positive by the expert discriminator receive a high final confidence. This approach effectively suppresses false positives, however, it tends to be highly conservative, potentially penalizing the overall recall.

Mean scoring function

To mitigate the restrictive nature of the joint scoring function, we also consider an arithmetic mean:

$$\text{Score}_{\text{mean}} = \frac{\text{Score}_{\text{det}} + \text{Score}_{\text{cls}}}{2} \quad (4.11)$$

Empirical observation suggests that the expert classifier often assigns higher confidence scores to positive instances compared to the zero-shot detector. Consequently, the mean scoring function exhibits a less penalizing effect on recall. This approach retains the discriminator’s ability to refine predictions, boosting the confidence of true positives when detector scores are low, and suppressing false positives when the expert classifier identifies a detection as background noise.

Chapter 5

Results and discussion

5.1 Supervised baseline domain shift problem

The primary challenge encountered in automated insect monitoring is the performance degradation of supervised models when deployed in unseen environments. This section analyzes the impact of the domain shift by grouping results based on the camera-to-subject distance.

Full-frame monitoring

The Ground ECOSTACK and Bjerge et al. datasets contain natural scenes where insects occupy only a small fraction of the total pixel area. Tables 5.1 and 5.2 summarize the performance of the supervised models in these complex scenarios.

On the Ground ECOSTACK dataset, all models struggle to generalize, highlighting this as the most challenging scenario presented in this study. The presence of many obstacles in the background, combined with the minute size of the target instances, renders the detection task particularly difficult.

In the Bjerge et al. dataset, only the model trained on the corresponding training set achieves high performance. Consistent with the Ground ECOSTACK results, all the other models trained on different datasets, even those with similar camera-to-subject distances, such as the Marin et al. model, perform poorly in Out-of-Distribution (OOD) settings. This clearly confirms the severity of the domain shift problem.

Interestingly, FlatBug exhibits a unique behavior in the Ground ECOSTACK scenario. Particularly, it demonstrates a higher recall compared to other models. This suggests that the relatively homogeneous background of the Ground ECOSTACK dataset, despite the extremely small insect size, favors the FlatBug model.

Experiment	IoU = 0.10					IoU = 0.50				
	AP	AR	P	R	$F1_{\max}$ (Conf)	AP	AR	P	R	$F1_{\max}$ (Conf)
FlatBug (OOD)	0.144	0.299	0.254	0.176	0.208 (0.39)	0.125	0.246	0.247	0.160	0.194 (0.40)
Marin et al. (OOD)	0.086	0.253	0.221	0.141	0.172 (0.49)	0.053	0.173	0.164	0.105	0.128 (0.49)
Stark et al. (OOD)	0.095	0.267	0.234	0.114	0.153 (0.48)	0.083	0.202	0.281	0.097	0.145 (0.52)
Bjerge et al. (OOD)	0.003	0.123	0.024	0.081	0.037 (0.57)	0.002	0.080	0.019	0.053	0.027 (0.63)

Table 5.1: Supervised models performance on the Ground ECOSTACK dataset

Experiment	IoU = 0.10					IoU = 0.50				
	AP	AR	P	R	$F1_{\max}$ (Conf)	AP	AR	P	R	$F1_{\max}$ (Conf)
FlatBug (OOD)	0.005	0.062	0.044	0.061	0.051 (0.10)	0.002	0.039	0.033	0.032	0.033 (0.16)
Marin et al. (OOD)	0.176	0.249	0.493	0.199	0.284 (0.26)	0.080	0.146	0.345	0.112	0.169 (0.34)
Stark et al. (OOD)	0.084	0.163	0.243	0.125	0.165 (0.19)	0.035	0.088	0.146	0.070	0.094 (0.20)
Bjerge et al. (IDD)	0.884	0.909	0.949	0.865	<i>0.905</i> (0.39)	0.858	0.887	0.938	0.816	<i>0.872</i> (0.51)

Table 5.2: Supervised models performance on the Bjerge et al. dataset

Single flower monitoring

In the RangeX and Ştefan et al. datasets, the camera is positioned closer to the subject, resulting in insects occupying a larger portion of the image. This setup theoretically allows models to leverage more discriminative features useful for detection.

The results in Tables 5.3 and 5.4 indicate that, although the task is simplified, the domain shift problem remains significant. Notably, the Marin et al. and Stark et al. models achieve a consistently high precision, suggesting that their detections are highly reliable when successful.

In contrast to full-frame monitoring, these models exhibit a more pronounced decline in $F1$ -score as the IoU threshold increases. This indicates a limitation not only in classification accuracy but also in precise spatial localization.

While the insects appear larger, the models fail to generalize to the variability of new backgrounds and varying insect scales. This generalization failure results primarily in a low recall, which drastically degrades the overall $F1$ -score. These findings suggest that even when insect details are clearer, background variability and scale variance remain primary bottlenecks for detection generalization in automated insect monitoring.

Experiment	IoU = 0.10					IoU = 0.50				
	AP	AR	P	R	$F1_{\max}$ (Conf)	AP	AR	P	R	$F1_{\max}$ (Conf)
FlatBug (OOD)	0.192	0.330	0.385	0.327	0.354 (0.10)	0.101	0.204	0.249	0.202	0.223 (0.11)
Marin et al. (OOD)	0.479	0.507	0.813	0.461	0.588 (0.21)	0.326	0.399	0.658	0.371	0.475 (0.21)
Stark et al. (OOD)	0.411	0.501	0.660	0.398	0.497 (0.23)	0.274	0.371	0.587	0.302	0.399 (0.28)
Bjerge et al. (OOD)	0.034	0.101	0.215	0.098	0.134 (0.11)	0.008	0.040	0.0091	0.040	0.056 (0.11)

Table 5.3: Supervised models performance on the RangeX dataset

Experiment	IoU = 0.10					IoU = 0.50				
	AP	AR	P	R	$F1_{\max}$ (Conf)	AP	AR	P	R	$F1_{\max}$ (Conf)
FlatBug (OOD)	0.458	0.532	0.645	0.495	0.560 (0.17)	0.322	0.401	0.548	0.360	0.435 (0.25)
Marin et al. (OOD)	0.701	0.733	0.848	0.677	0.753 (0.37)	0.478	0.563	0.716	0.492	0.584 (0.54)
Stark et al. (OOD)	0.643	0.650	0.897	0.634	0.743 (0.14)	0.607	0.623	0.865	0.611	0.716 (0.14)
Bjerge et al. (OOD)	0.214	0.364	0.350	0.350	0.350 (0.12)	0.014	0.088	0.117	0.069	0.087 (0.31)

Table 5.4: Supervised models performance on the Ştefan et al. dataset

5.2 Zero-shot detector generalization capabilities

After observing the evident domain shift problem in supervised models, the following step is to evaluate the generalization capabilities of zero-shot object detectors, which, by design and by their size, should be more robust to OOD scenarios.

Full-frame monitoring

Tables 5.5 and 5.6 demonstrate that zero-shot detectors exhibit significant improvements over supervised models.

On the Ground ECOSTACK dataset, both SAM3 and Grounding DINO achieve a substantial increase in recall. Notably, Average Recall (AR) is particularly high when using the basic prompt (93% for SAM3 and 89.2% for Grounding DINO). This indicates that the models successfully localize the vast majority of insects in the scene. However, both struggle to discriminate them from background distractors, leading to numerous false positives (e.g., sticks or leaves misclassified as insects) and a consequently low Average Precision (AP).

Performance on the Bjerge et al. dataset is more heterogeneous. SAM3 achieves strong results with the basic prompt, while Grounding DINO demonstrates remarkable precision (95.8%) using the taxonomic prompt, although at the cost of lower recall (27.4%). Furthermore, the discrepancy in $F1$ -score across different IoU thresholds indicates that both models face challenges in precise bounding box placement.

Finally, it is interesting to note that Grounding DINO achieves its best precision using a very low confidence threshold (0.1), suggesting a conservative detection strategy that, in this case, minimizes false positives. In contrast, SAM3 maintains a higher operating threshold (0.57), indicating that, even at higher confidence levels, the model can still produce a significant number of false positives.

Single flower monitoring

As shown in Tables 5.7 and 5.8, zero-shot detectors outperform supervised models in the single flower monitoring setup, enabling both architectures to achieve remarkably high $F1$ -score values in their best configurations.

Experiment	IoU = 0.10					IoU = 0.50				
	AP	AR	P	R	$F1_{\max}$ (Conf)	AP	AR	P	R	$F1_{\max}$ (Conf)
SAM3 P1 (Basic)	0.215	0.930	0.296	0.347	0.320 (0.82)	0.136	0.584	0.286	0.263	0.274 (0.84)
SAM3 P2 (Contextual)	0.273	0.632	0.350	0.345	0.348 (0.30)	0.168	0.429	0.261	0.257	0.259 (0.30)
SAM3 P3 (Taxonomic)	0.315	0.755	0.417	0.376	0.396 (0.46)	0.184	0.494	0.307	0.290	0.298 (0.45)
Grounding DINO P1 (Basic)	0.223	0.892	0.363	0.254	0.299 (0.58)	0.204	0.674	0.344	0.240	0.283 (0.58)
Grounding DINO P3 (Taxonomic)	0.296	0.559	0.684	0.234	0.348 (0.33)	0.269	0.505	0.726	0.207	0.322 (0.35)

Table 5.5: Zero-shot performance detectors on the Ground ECOSTACK dataset. See Section 4.2.4 for detailed descriptions of the prompt configurations.

Experiment	IoU = 0.10					IoU = 0.50				
	AP	AR	P	R	$F1_{\max}$ (Conf)	AP	AR	P	R	$F1_{\max}$ (Conf)
SAM3 P1 (Basic)	0.753	0.848	0.878	0.675	0.763 (0.57)	0.171	0.292	0.423	0.230	0.298 (0.72)
SAM3 P2 (Contextual)	0.427	0.470	0.751	0.438	0.554 (0.12)	0.113	0.222	0.365	0.208	0.265 (0.12)
SAM3 P3 (Taxonomic)	0.568	0.642	0.787	0.548	0.646 (0.16)	0.065	0.220	0.257	0.197	0.223 (0.15)
Grounding DINO P1 (Basic)	0.235	0.541	0.446	0.238	0.310 (0.43)	0.159	0.348	0.411	0.189	0.259 (0.45)
Grounding DINO P3 (Taxonomic)	0.273	0.274	0.958	0.274	0.426 (0.10)	0.231	0.246	0.860	0.246	0.382 (0.10)

Table 5.6: Zero-shot performance detectors on the Bjerger et al. dataset. See Section 4.2.4 for detailed descriptions of the prompt configurations.

On the RangeX dataset, the models exhibit the same behavioral patterns observed in full-frame monitoring: SAM3 performs better when guided by the basic prompt, while Grounding DINO with a taxonomic one. However, the performance gap between these prompting strategies is less pronounced in this scenario, likely because the reduced camera-to-subject distance inherently simplifies the detection task.

Analyzing the Ștefan et al. dataset, SAM3 demonstrates impressive OOD generalization, reaching an $F1$ -score of 0.91 when guided by the basic prompt. Grounding DINO also yields strong results with the taxonomic prompt, maintaining a precision above 90%, although with lower recall than SAM3. This overall high performance is heavily favored by the nature of the dataset itself. Indeed, images are tightly cropped around individual flowers, so the target insects occupy a larger portion of the frame and background heterogeneity is minimized, mostly restricted to the flower itself.

Across both datasets, Grounding DINO generates more precise bounding boxes, as evidenced by its minimal $F1$ -score degradation at the higher IoU threshold of 0.5. In contrast, SAM3 struggles with exact spatial localization, often failing to capture very small insect details, such as legs, antennae, or wings. The omission of these fine features significantly penalizes the intersection area, leading to a drastic drop in the final IoU calculation.

5.3 Results of the proposed ensembling strategy

The ensembling strategy (described in Section 4.2.6) was designed to mitigate the sensitivity of zero-shot detectors to specific prompt formulations by aggregating predictions

Experiment	IoU = 0.10					IoU = 0.50				
	AP	AR	P	R	$F1_{\max}$ (Conf)	AP	AR	P	R	$F1_{\max}$ (Conf)
SAM3 P1 (Basic)	0.858	0.970	0.815	0.824	0.819 (0.64)	0.469	0.637	0.566	0.571	0.569 (0.65)
SAM3 P2 (Contextual)	0.633	0.752	0.639	0.658	0.648 (0.15)	0.337	0.517	0.537	0.403	0.460 (0.21)
SAM3 P3 (Taxonomic)	0.652	0.846	0.665	0.643	0.654 (0.23)	0.275	0.563	0.435	0.421	0.428 (0.23)
Grounding DINO P1 (Basic)	0.690	0.923	0.529	0.737	0.616 (0.37)	0.488	0.703	0.563	0.506	0.533 (0.44)
Grounding DINO P3 (Taxonomic)	0.739	0.756	0.861	0.744	0.798 (0.12)	0.619	0.670	0.831	0.628	0.715 (0.17)

Table 5.7: Zero-shot performance detectors on the RangeX dataset. See Section 4.2.4 for detailed descriptions of the prompt configurations.

Experiment	IoU = 0.10					IoU = 0.50				
	AP	AR	P	R	$F1_{\max}$ (Conf)	AP	AR	P	R	$F1_{\max}$ (Conf)
SAM3 P1 (Basic)	0.948	0.981	0.936	0.884	0.910 (0.61)	0.419	0.519	0.607	0.474	0.532 (0.75)
SAM3 P2 (Contextual)	0.721	0.775	0.814	0.682	0.742 (0.15)	0.341	0.509	0.567	0.445	0.499 (0.17)
SAM3 P3 (Taxonomic)	0.768	0.869	0.735	0.770	0.752 (0.15)	0.209	0.476	0.393	0.412	0.402 (0.15)
Grounding DINO P1 (Basic)	0.834	0.926	0.822	0.843	0.832 (0.35)	0.610	0.717	0.724	0.656	0.688 (0.40)
Grounding DINO P3 (Taxonomic)	0.813	0.825	0.906	0.806	0.853 (0.12)	0.662	0.700	0.867	0.648	0.742 (0.17)

Table 5.8: Zero-shot performance detectors on the Ştefan et al. dataset. See Section 4.2.4 for detailed descriptions of the prompt configurations.

from multiple configurations. The results, summarized in Tables 5.9 through 5.12, reveal a strong dependence on the complexity of the monitoring scenario.

In challenging environments, such as the Ground ECOSTACK dataset, the ensemble acts as a regularizer. For SAM3, the ensemble achieves an $F1$ -score (0.425 at IoU = 0.1) that outperforms all individual prompt configurations (P1, P2, and P3). By requiring spatial consensus across different prompts, this strategy effectively filters out hallucinations, false positives that appear under one prompt formulation but not others, thereby stabilizing performance in scenarios where individual configurations exhibit highly diverse results.

In contrast, in simpler or highly controlled scenarios, such as the Ştefan et al. dataset, where insects occupy a larger portion of the frame, the ensemble tends to slightly suppress the peak performance of the best-performing individual prompt. For instance, the near-optimal $F1$ -score of 0.910 achieved by SAM3 P1 is reduced by 3.1 percentage points after the ensemble strategy was applied. This degradation occurs because highly accurate detections from the optimal configuration are penalized if ignored or assigned a low score by the alternative prompts.

However, this trend does not uniformly apply to all architectures. On the RangeX and Ştefan et al. datasets, the Grounding DINO ensemble successfully outperforms its respective single-prompt configurations, achieving peak $F1$ -scores of respectively 0.810 and 0.856 at IoU = 0.1.

In conclusion, while the ensemble may not always match the absolute maximum performance of a perfectly tuned single prompt in every specific scenario, it provides an important fail-safe mechanism. For real-world ecological monitoring systems, where the

optimal prompt cannot be determined a priori for unseen environments, the ensemble strategy helps ensure reliability across diverse datasets.

Experiment	IoU = 0.10					IoU = 0.50				
	AP	AR	P	R	$F1_{\max}$ (Conf)	AP	AR	P	R	$F1_{\max}$ (Conf)
SAM3 P1 (Basic)	0.215	0.930	0.296	0.347	0.320 (0.82)	0.136	0.584	0.286	0.263	0.274 (0.84)
SAM3 P2 (Contextual)	0.273	0.632	0.350	0.345	0.348 (0.30)	0.168	0.429	0.261	0.257	0.259 (0.30)
SAM3 P3 (Taxonomic)	0.315	0.755	0.417	0.376	0.396 (0.46)	0.184	0.494	0.307	0.290	0.298 (0.45)
SAM3 Ensemble	0.362	0.932	0.453	0.401	0.425 (0.50)	0.214	0.588	0.345	0.308	0.325 (0.50)
Grounding DINO P1 (Basic)	0.223	0.892	0.363	0.254	0.299 (0.58)	0.204	0.674	0.344	0.240	0.283 (0.58)
Grounding DINO P3 (Taxonomic)	0.296	0.559	0.684	0.234	0.348 (0.33)	0.269	0.505	0.726	0.207	0.322 (0.35)
Grounding DINO Ensemble	0.308	0.893	0.558	0.253	0.348 (0.44)	0.273	0.681	0.561	0.232	0.328 (0.44)

Table 5.9: Performance of the proposed ensembling strategy for zero-shot detectors on the Ground ECOSTACK dataset. See Section 4.2.4 for detailed descriptions of the prompt configurations.

Experiment	IoU = 0.10					IoU = 0.50				
	AP	AR	P	R	$F1_{\max}$ (Conf)	AP	AR	P	R	$F1_{\max}$ (Conf)
SAM3 P1 (Basic)	0.753	0.848	0.878	0.675	0.763 (0.57)	0.171	0.292	0.423	0.230	0.298 (0.72)
SAM3 P2 (Contextual)	0.427	0.470	0.751	0.438	0.554 (0.12)	0.113	0.222	0.365	0.208	0.265 (0.12)
SAM3 P3 (Taxonomic)	0.568	0.642	0.787	0.548	0.646 (0.16)	0.065	0.220	0.257	0.197	0.223 (0.15)
SAM3 Ensemble	0.728	0.848	0.839	0.647	0.731 (0.24)	0.127	0.297	0.369	0.246	0.295 (0.28)
Grounding DINO P1 (Basic)	0.235	0.541	0.446	0.238	0.310 (0.43)	0.159	0.348	0.411	0.189	0.259 (0.45)
Grounding DINO P3 (Taxonomic)	0.273	0.274	0.958	0.274	0.426 (0.10)	0.231	0.246	0.860	0.246	0.382 (0.10)
Grounding DINO Ensemble	0.320	0.542	0.756	0.254	0.380 (0.26)	0.243	0.363	0.708	0.222	0.338 (0.26)

Table 5.10: Performance of the proposed ensembling strategy for zero-shot detectors on the Bjerge et al. dataset. See Section 4.2.4 for detailed descriptions of the prompt configurations.

5.4 Negative prompting analysis

To further investigate the flexibility of zero-shot detectors, the use of negative prompting was explored. This strategy involves explicitly instructing the model to identify separately background entities that frequently co-occur with the targets and can be confused with the target class. This analysis was conducted exclusively using Grounding DINO, as its architecture provides natively class-aware predictions.

The RangeX dataset was selected as the primary benchmark for this experiment. It represents a unique scenario among the evaluated datasets because it features a clear and consistent foreground entity, the flower on which the insect could be positioned, that can be explicitly defined in the prompt. This approach was not extended to the Ștefan et al. dataset, as the flower always occupies the entire background of the crop.

As shown in Table 5.13, the introduction of a negative prompt, explicitly defining “flower” as a class to be further ignored, yields significant performance improvements.

Experiment	IoU = 0.10					IoU = 0.50				
	AP	AR	P	R	$F1_{\max}$ (Conf)	AP	AR	P	R	$F1_{\max}$ (Conf)
SAM3 P1 (Basic)	0.858	0.970	0.815	0.824	0.819 (0.64)	0.469	0.637	0.566	0.571	0.569 (0.65)
SAM3 P2 (Contextual)	0.633	0.752	0.639	0.658	0.648 (0.15)	0.337	0.517	0.537	0.403	0.460 (0.21)
SAM3 P3 (Taxonomic)	0.652	0.846	0.665	0.643	0.654 (0.23)	0.275	0.563	0.435	0.421	0.428 (0.23)
SAM3 Ensemble	0.818	0.972	0.798	0.780	0.789 (0.34)	0.397	0.649	0.556	0.544	0.550 (0.34)
Grounding DINO P1 (Basic)	0.690	0.923	0.529	0.737	0.616 (0.37)	0.488	0.703	0.563	0.506	0.533 (0.44)
Grounding DINO P3 (Taxonomic)	0.739	0.756	0.861	0.744	0.798 (0.12)	0.619	0.670	0.831	0.628	0.715 (0.17)
Grounding DINO Ensemble	0.811	0.923	0.945	0.710	0.810 (0.28)	0.617	0.725	0.857	0.599	0.705 (0.31)

Table 5.11: Performance of the proposed ensembling strategy for zero-shot detectors on the RangeX dataset. See Section 4.2.4 for detailed descriptions of the prompt configurations.

Experiment	IoU = 0.10					IoU = 0.50				
	AP	AR	P	R	$F1_{\max}$ (Conf)	AP	AR	P	R	$F1_{\max}$ (Conf)
SAM3 P1 (Basic)	0.948	0.981	0.936	0.884	0.910 (0.61)	0.419	0.519	0.607	0.474	0.532 (0.75)
SAM3 P2 (Contextual)	0.721	0.775	0.814	0.682	0.742 (0.15)	0.341	0.509	0.567	0.445	0.499 (0.17)
SAM3 P3 (Taxonomic)	0.768	0.869	0.735	0.770	0.752 (0.15)	0.209	0.476	0.393	0.412	0.402 (0.15)
SAM3 Ensemble	0.926	0.982	0.895	0.864	0.879 (0.30)	0.327	0.541	0.541	0.512	0.526 (0.31)
Grounding DINO P1 (Basic)	0.834	0.926	0.822	0.843	0.832 (0.35)	0.610	0.717	0.724	0.656	0.688 (0.40)
Grounding DINO P3 (Taxonomic)	0.813	0.825	0.906	0.806	0.853 (0.12)	0.662	0.700	0.867	0.648	0.742 (0.17)
Grounding DINO Ensemble	0.882	0.926	0.902	0.815	0.856 (0.25)	0.675	0.734	0.883	0.652	0.750 (0.33)

Table 5.12: Performance of the proposed ensembling strategy for zero-shot detectors on the Ştefan et al. dataset. See Section 4.2.4 for detailed descriptions of the prompt configurations.

In particular, observing the basic prompt configuration, the precision at IoU equal to 0.1 increases from 0.529 to 0.909 with the introduction of the negative prompt, only with a slightly penalizing effect on the recall. This indicates that the model effectively leverages the negative cues to better differentiate the insect from the flower.

Observing the taxonomic prompt configuration, the performance improvement is less pronounced. This might be due to the already rich information provided by that prompt, that allows the model to correctly discriminate insects from the background.

Due to the nature of the results, also the ensembling strategy applied to the negative prompting configurations was evaluated, where the ensemble of negative prompts outperforms the single-prompt configurations, achieving the best $F1$ -score at IoU = 0.1.

Experiment	IoU = 0.10					IoU = 0.50				
	AP	AR	P	R	$F1_{\max}$ (Conf)	AP	AR	P	R	$F1_{\max}$ (Conf)
Grounding DINO P1 (Basic)	0.690	0.923	0.529	0.737	0.616 (0.37)	0.488	0.703	0.563	0.506	0.533 (0.44)
Grounding DINO P3 (Taxonomic)	0.739	0.756	0.861	0.744	0.798 (0.12)	0.619	0.670	0.831	0.628	0.715 (0.17)
Grounding DINO Ensemble	0.811	0.923	0.945	0.710	0.810 (0.28)	0.617	0.725	0.857	0.599	0.705 (0.31)
Grounding DINO P1 (Basic) Neg.	0.803	0.870	0.909	0.709	0.797 (0.35)	0.573	0.673	0.810	0.587	0.681 (0.38)
Grounding DINO P3 (Taxonomic) Neg.	0.754	0.767	0.869	0.757	0.809 (0.13)	0.637	0.684	0.865	0.638	0.734 (0.20)
Grounding DINO Ensemble Neg.	0.819	0.873	0.921	0.749	0.826 (0.22)	0.617	0.696	0.843	0.637	0.726 (0.27)

Table 5.13: Performance of negative prompting with Grounding DINO on the RangeX dataset. See Section 4.2.4 for detailed descriptions of the prompt configurations.

5.5 Binary discriminator results

Intrinsic binary classification performance

After training, the performance of both the MLP and SVM classifiers is evaluated on in-distribution and out-of-distribution test sets. Particular emphasis is placed on the out-of-distribution samples, which feature insects and backgrounds completely unseen during training, to rigorously assess the models’ generalization capabilities. The evaluation metrics include accuracy, precision, recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC).

Observing the results in Table 5.14, it is evident that both classifiers perform exceptionally well on the In-Domain-Distribution (IDD) test set, achieving near-perfect scores across all metrics. However, the true test of their robustness lies in the OOD performance. Here, the MLP classifier on BioCLIP 2.5 embeddings demonstrates superior generalization, achieving a higher $F1$ -score of 0.89 and an AUC of 0.96, indicating a strong ability to discriminate between insect and background detections even in unseen conditions.

Test set	In-Domain-Distribution (IDD)					Out-of-Distribution (OOD)				
Classifier	Acc	Prec	Rec	F1	AUC	Acc	Prec	Rec	F1	AUC
MLP + DINOv3	0.97	0.97	0.97	0.97	0.99	0.83	0.85	0.83	0.83	0.92
MLP + BioCLIP 2.5	0.97	0.97	0.97	0.97	1.00	0.89	0.89	0.89	0.89	0.96
SVM + DINOv3	0.96	0.96	0.96	0.96	0.99	0.85	0.85	0.85	0.85	0.92
SVM + BioCLIP 2.5	0.97	0.97	0.97	0.97	1.00	0.86	0.86	0.86	0.86	0.94

Table 5.14: Performance metrics of MLP and SVM classifiers on test sets. See Section 4.2.4 for detailed descriptions of the prompt configurations.

Impact of expert re-scoring

In this stage, the binary classifiers are integrated into the detection pipeline to act as *expert discriminators*. The goal is to refine the outputs of the SAM3 zero-shot detector with a basic prompt (P1), identified as the best-performing configuration across different datasets, by re-scoring each candidate crop based on its embedding representation.

Analyzing the RangeX dataset (Tables 5.15 and 5.16), almost all configurations successfully improve precision across both joint and mean scoring strategies, with gains ranging from 2.9 to 9.2 percentage points. The only exception is the SVM combined with BioCLIP 2.5 under joint scoring. Notably, recall only improves in the optimal configurations where the overall $F1$ -score also increases. Specifically, the MLP paired with DINOv3

yields a recall increase of 1.6 points with joint scoring and 0.6 points with mean scoring. This configuration achieves the best overall performance, boosting the $F1$ by +2.3 and +3.7 points for joint and mean scoring, respectively.

A similar trend regarding precision is observed in the Ştefan et al. dataset (Tables 5.17 and 5.18). In the best-performing configuration, which for this dataset is the MLP with BioCLIP 2.5, precision increases by 1.5 points using joint scoring and 2.6 points using mean scoring. However, this is accompanied by a consistent decline in recall, which consequently leads to a reduction in the overall $F1$ -score across all configurations. This drop might be justified by the fact that the baseline performance of SAM3 was already high, leaving a very narrow margin for improvement, particularly considering the fully OOD nature of this pipeline.

Overall, when the baseline performance of SAM3 leaves room for improvement, integrating an expert discriminator significantly enhances precision, although with a necessary trade-off in recall. In this context, the mean scoring technique proves more effective at preserving recall. It effectively mitigates the strict penalization inherent in joint scoring. By averaging the predictions, a high confidence score from the expert discriminator can elevate an initially low-confidence true positive, providing a recovery mechanism that prevents valid detections from falling below the acceptance threshold.

Regarding the optimal foundation model for feature extraction, further investigation is required to establish a definitive standard approach. DINOv3 proved most effective for the RangeX dataset, while BioCLIP 2.5 was the best one on the Ştefan et al. dataset. Interestingly, both achieved their best results when paired with an MLP classifier, suggesting that the SVM might be a less robust choice, potentially constrained by the complexity of the latent space representations.

Experiment	IoU = 0.10					IoU = 0.50				
	AP	AR	P	R	$F1_{\max}$ (Conf)	AP	AR	P	R	$F1_{\max}$ (Conf)
SAM3 P1 (Basic)	0.858	0.970	0.815	0.824	0.819 (0.64)	0.469	0.637	0.566	0.571	0.569 (0.65)
SAM3 P1 MLP D3	0.876	0.970	0.844	0.840	0.842 +2.3 (0.55)	0.475	0.637	0.616	0.570	0.592 (0.62)
SAM3 P1 MLP BC2.5	0.857	0.970	0.882	0.800	0.839+2.0 (0.57)	0.470	0.637	0.648	0.540	0.589 (0.65)
SAM3 P1 SVM D3	0.863	0.970	0.887	0.784	0.833+1.4 (0.53)	0.463	0.637	0.658	0.523	0.583 (0.62)
SAM3 P1 SVM BC2.5	0.801	0.970	0.788	0.735	0.760-5.9 (0.43)	0.414	0.637	0.618	0.450	0.521 (0.59)

Table 5.15: Performance of embedding-based re-scoring on the RangeX dataset by using an expert discriminator (joint scoring). See Section 4.2.4 for detailed descriptions of the prompt configurations.

Time complexity analysis

Table 5.19 reports inference times of the primary detectors analyzed in this work. The additional time required by the expert discriminators strictly depends on the number of candidate crops generated by the zero-shot detectors, which is in turn influenced by the

Experiment	IoU = 0.10					IoU = 0.50				
	AP	AR	P	R	$F1_{\max}$ (Conf)	AP	AR	P	R	$F1_{\max}$ (Conf)
SAM3 P1 (Basic)	0.858	0.970	0.815	0.824	0.819 (0.64)	0.469	0.637	0.566	0.571	0.569 (0.65)
SAM3 P1 MLP D3	0.876	0.970	0.884	0.830	0.856 ^{+3.7} (0.80)	0.475	0.637	0.641	0.560	0.598 (0.83)
SAM3 P1 MLP BC2.5	0.857	0.970	0.881	0.800	0.838 ^{+1.9} (0.78)	0.470	0.637	0.651	0.540	0.590 (0.82)
SAM3 P1 SVM D3	0.879	0.970	0.907	0.800	0.850 ^{+3.1} (0.78)	0.472	0.637	0.631	0.560	0.593 (0.77)
SAM3 P1 SVM BC2.5	0.813	0.970	0.846	0.720	0.778 ^{-4.1} (0.74)	0.436	0.637	0.600	0.500	0.545 (0.75)

Table 5.16: Performance of embedding-based re-scoring on the RangeX dataset by using an expert discriminator (mean scoring). See Section 4.2.4 for detailed descriptions of the prompt configurations.

Experiment	IoU = 0.10					IoU = 0.50				
	AP	AR	P	R	$F1_{\max}$ (Conf)	AP	AR	P	R	$F1_{\max}$ (Conf)
SAM3 P1 (Basic)	0.948	0.981	0.936	0.884	0.910 (0.61)	0.419	0.519	0.607	0.474	0.532 (0.75)
SAM3 P1 MLP D3	0.919	0.981	0.927	0.830	0.876 ^{-3.4} (0.53)	0.395	0.519	0.622	0.420	0.502 (0.78)
SAM3 P1 MLP BC2.5	0.922	0.981	0.951	0.830	0.887 ^{-2.3} (0.59)	0.412	0.519	0.634	0.450	0.526 (0.78)
SAM3 P1 SVM D3	0.924	0.981	0.912	0.826	0.867 ^{-4.3} (0.34)	0.339	0.519	0.469	0.460	0.465 (0.29)
SAM3 P1 SVM BC2.5	0.801	0.981	0.929	0.818	0.870 ^{-4.0} (0.48)	0.396	0.519	0.605	0.428	0.501 (0.69)

Table 5.17: Performance of embedding-based re-scoring on the Ştefan et al. dataset by using an expert discriminator (joint scoring). See Section 4.2.4 for detailed descriptions of the prompt configurations.

Experiment	IoU = 0.10					IoU = 0.50				
	AP	AR	P	R	$F1_{\max}$ (Conf)	AP	AR	P	R	$F1_{\max}$ (Conf)
SAM3 P1 (Basic)	0.948	0.981	0.936	0.884	0.910 (0.61)	0.419	0.519	0.607	0.474	0.532 (0.75)
SAM3 P1 MLP D3	0.917	0.981	0.924	0.830	0.875 ^{-3.5} (0.76)	0.394	0.519	0.621	0.420	0.501 (0.89)
SAM3 P1 MLP BC2.5	0.920	0.981	0.962	0.820	0.886 ^{-2.4} (0.81)	0.411	0.519	0.634	0.450	0.526 (0.89)
SAM3 P1 SVM D3	0.923	0.981	0.895	0.830	0.861 ^{-4.9} (0.63)	0.338	0.519	0.481	0.450	0.465 (0.63)
SAM3 P1 SVM BC2.5	0.857	0.970	0.881	0.800	0.838 ^{-7.2} (0.78)	0.470	0.637	0.651	0.540	0.590 (0.82)

Table 5.18: Performance of embedding-based re-scoring on the Ştefan et al. dataset by using an expert discriminator (mean scoring). See Section 4.2.4 for detailed descriptions of the prompt configurations.

confidence threshold applied to the initial predictions. However, as a general reference, the feature extraction process remains the most computationally intensive additional step, scaling with the size of the chosen image encoder. In contrast, the final binary classification step is relatively negligible in terms of latency, given the lightweight nature of the 3-layer MLP and SVM architectures.

Note that only the inference time of the Marin et al. YOLO model is reported, as the inference times of all other supervised baselines are nearly identical to it. Furthermore, for configurations where a sliced inference strategy is applied, a *SAHI* tag is included, indicating that the final outputs are obtained through multiple forward passes of the model over image patches.

Dataset (Image size ↓) (Insect size ↑)	SAM3 (800M)	Grounding DINO (341M)	Marin et al. (YOLO) (21M)
Ground ECOSTACK	5.17 sec/img (<i>SAHI</i>)	2.05 sec/img (<i>SAHI</i>)	0.14 sec/img (<i>SAHI</i>)
Bjerge et al.	2.58 sec/img (<i>SAHI</i>)	0.28 sec/img (<i>SAHI</i>)	0.03 sec/img (<i>SAHI</i>)
RangeX	0.47 sec/img	0.08 sec/img	0.01 sec/img
Ştefan et al.	0.18 sec/img	0.04 sec/img	0.01 sec/img

Table 5.19: Inference time comparison across different models and datasets.

5.6 Limitations and future work

The proposed pipeline, while demonstrating significant advantages over state-of-the-art supervised baselines in terms of generalization, still possesses limitations that should be addressed in future research:

- **Inference latency:** The current architecture, which relies on heavy foundation models and ensembles of different prompts, is computationally intensive. While suitable for offline ecological analysis, this high inference latency currently precludes deployment on edge AI devices for real-time field monitoring. Future work should explore model distillation or pruning techniques to maintain zero-shot capabilities within lighter, more efficient architectures [65, 66]. This would enable faster inference, broader applicability in resource-constrained environments and potentially allow for on-device inference.

The additional computational cost introduced by the expert discriminator increases the latency issue, as it currently requires a separate forward pass for each candidate crop. Future research should investigate more efficient integration strategies. For instance, exploiting features already computed by the zero-shot detector would leave only the binary classification step, adding minimal overhead to the overall inference time.

Furthermore, the proposed ensembling technique introduces a computational overhead by requiring a distinct forward pass for each prompt configuration. Similar to the aforementioned optimization, the features extracted once by the vision encoder could be reused for the text similarity computation. This would restrict the additional cost to a single vision encoder forward pass and a set of text encoder forward passes equal to the number of prompts, extracted only once. A comparable optimization could also be beneficial when applied to streamline the sliced inference strategy.

- **Ensemble weighting mechanism:** The currently adopted ensembling strategy treats all prompt configurations as equally reliable. However, empirical results indicate that certain prompts (e.g., basic and taxonomic) consistently outperform others in specific scenarios and models. A potential development involves implementing a dynamic weighting mechanism that assigns higher importance to the most reliable prompts based on the visual context of the image or on the precision achieved on a validation set specific to a certain deployment.
- **Bounding box precision:** Zero-shot detectors, particularly SAM3, exhibited a significant performance drop when moving from a loose ($\text{IoU} = 0.1$) to a strict ($\text{IoU} = 0.5$) threshold. This may be less critical for a subsequent classification stage, where the FixRes [67] technique and robust classifier training can help mitigate the impact of imprecise bounding boxes. Nonetheless, it still represents a limitation of the current pipeline.
- **Binary discriminator implementation:** This pipeline introduces a novel strategy of training a binary discriminator to particularly prune false positives from zero-shot detector results. However, the current implementation relies on relatively simple MLP or SVM architectures. Future work should explore more sophisticated expert models designed to be robust to OOD samples while minimizing the rejection of true positives. Furthermore, alternative re-scoring strategies and integration methods for the expert model’s output could be investigated, with the purpose of increasing precision, at least without altering the already obtained recall.

Additionally, the current discriminator training set was built using random background crops. It might be more beneficial to implement hard negative mining by selecting background samples typically confused with the target class (e.g., sticks, flowers, or other objects with insect-like shapes). These hard negatives could be extracted by collecting the false positives generated when zero-shot detectors are operated at low confidence thresholds, requiring only minimal human supervision.

Another possible future work involves investigating dataset-specific development of lightweight SVM models. Specifically, it would be interesting to determine the minimum number of annotated samples required to train a reliable discriminator,

potentially enabling rapid adaptation to new monitoring sites with minimal annotation effort.

Chapter 6

Conclusions

This thesis addresses a common challenge in automated ecological monitoring: the poor generalization of standard supervised architectures, such as YOLO models, under domain shift. While these models perform remarkably well on in-distribution data or in easier tasks like localizing insects over homogeneous backgrounds, this work demonstrates their failure when encountering heterogeneous backgrounds, shifting object scales, and unfamiliar taxa. Since many of the tested baseline models were originally designed specifically for pollinators, the decline of these models on novel taxa, such as woodlice (*Isopoda*), is not surprising, though it remains a crucial factor to quantify and consider depending on the specific use case. To overcome this limitation, this work proposes a zero-shot detection pipeline paired with an efficiently-trained expert verification step.

Zero-shot foundation models, specifically Grounding DINO and SAM3, demonstrated excellent out-of-distribution capabilities despite not being specialized for ecological tasks; in several cases, SAM3 alone already achieved strong results without any additional training. To further bridge this lack of specialization where needed, expert discriminators requiring minimal supervision were introduced. Training these specialized classifiers proved to be significantly more efficient than training an entire object detection network from scratch. This approach offers a highly promising solution for increasing precision and refining detectors' predictions, ultimately obtaining reliable results on single flower monitoring datasets and consistently outperforming state-of-the-art supervised models.

Beyond benchmarking, this thesis provides actionable guidelines for the ecology and entomology communities. It outlines practical strategies for model selection, targeted prompting (including negative prompts), confidence calibration, and IoU or ensemble-based filtering across various insect detection use cases. By demonstrating that zero-shot models can perform effectively even without additional training, this work offers a highly accessible tool for practitioners who lack the programming expertise typically required to train and deploy custom deep learning architectures.

However, the proposed pipeline is not without limitations. The foundational architectures benchmarked in this study are computationally heavy, making real-time field

deployment impossible and presenting scalability challenges for large-scale monitoring projects without access to high-performance hardware. Regarding performance, foundation models in this context are highly capable of semantically localizing objects but struggle to draw precise boundaries. Indeed, results at common spatial overlap thresholds (e.g., $IoU = 0.5$) remain significantly lower than those obtained with supervised models. Nevertheless, since the primary goal of these monitoring efforts is to accurately count instances rather than measure them precisely, the lack of perfectly tight bounding boxes does not compromise the overall value of the models for ecological applications.

Future research must address these operational constraints by exploring established model compression, pruning, and knowledge distillation techniques to maintain zero-shot capabilities within lighter, more efficient architectures.

Finally, this work introduces promising new research directions for automated ecological monitoring. The community’s focus might beneficially shift from training narrow supervised detection models to efficiently leveraging existing large-scale foundation models, possibly extending the application of targeted expert discriminators to implicitly inject the necessary domain specialization. Since training these discriminators is comparatively fast and scalable, even though inference with large foundation models remains resource-intensive, this approach could directly enable advanced behavioral studies, help in image annotation, facilitate the efficient monitoring of critical pollinators and agricultural pests, and play a vital role in understanding and combating global insect declines.

Acknowledgments

I would like to express my deepest gratitude to my co-supervisors, Dr. Jamie Alison and Prof. Toke Thomas Høye, for their support and for the invaluable opportunity to work on this project at Aarhus University, which has been a truly enriching experience thanks to all the wonderful people at the Høye Lab. I am also grateful to my supervisor, Prof. Giuseppe Lisanti, for having accepted to supervise this thesis and for his guidance. Finally, I would like to thank my family and friends for their constant support and encouragement throughout this journey.

Bibliography

- [1] A. Balasubramaniam and S. Pasricha, “Object detection in autonomous vehicles: Status and open challenges,” *arXiv preprint arXiv:2201.07706*, 2022.
- [2] F. C. Akyon, S. O. Altinuc, and A. Temizel, “Slicing aided hyper inference and fine-tuning for small object detection,” in *2022 IEEE international conference on image processing (ICIP)*, pp. 966–970, IEEE, 2022.
- [3] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang, *et al.*, “Sam 3: Segment anything with concepts,” *arXiv preprint arXiv:2511.16719*, 2025.
- [4] N. E. Stork, “How many species of insects and other terrestrial arthropods are there on earth?,” *Annual review of entomology*, vol. 63, no. 2018, pp. 31–45, 2018.
- [5] Y. M. Bar-On, R. Phillips, and R. Milo, “The biomass distribution on earth,” *Proceedings of the National Academy of Sciences*, vol. 115, no. 25, pp. 6506–6511, 2018.
- [6] J. P. van der Sluijs, “Insect decline, an emerging global environmental risk,” *Current Opinion in Environmental Sustainability*, vol. 46, pp. 39–42, 2020. Reflections on advances in health and environment research in the context of the COVID-19 pandemic.
- [7] C. A. Hallmann, M. Sorg, E. Jongejans, H. Siepel, N. Hofland, H. Schwan, W. Stenmans, A. Müller, H. Sumser, T. Hörren, D. Goulson, and H. de Kroon, “More than 75 percent decline over 27 years in total flying insect biomass in protected areas,” *PLOS ONE*, vol. 12, pp. 1–21, 10 2017.
- [8] R. Van Klink, T. August, Y. Bas, P. Bodesheim, A. Bonn, F. Fossøy, T. T. Høye, E. Jongejans, M. H. Menz, A. Miraldo, *et al.*, “Emerging technologies revolutionise insect ecology and monitoring,” *Trends in ecology & evolution*, vol. 37, no. 10, pp. 872–885, 2022.
- [9] A. Svenning, G. Mougeot, J. Alison, D. Chevalier, N. C. Molina, S. Ong, K. Bjerger, J. Carrillo, T. T. Høye, and Q. Geissmann, “A general method for detection and segmentation of terrestrial arthropods in images,” *Methods in Ecology and Evolution*, vol. 17, p. 727–739, Feb. 2026.

- [10] M. Sittinger, J. Uhler, M. Pink, and A. Herz, “Insect detect: An open-source diy camera trap for automated insect monitoring,” *Plos one*, vol. 19, no. 4, p. e0295474, 2024.
- [11] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, *et al.*, “Grounding dino: Marrying dino with grounded pre-training for open-set object detection,” in *European conference on computer vision*, pp. 38–55, Springer, 2024.
- [12] D. Goulson, “The insect apocalypse, and why it matters,” *Current Biology*, vol. 29, p. R967–R971, Oct. 2019.
- [13] M. Artamendi, P. A. Martin, I. Bartomeus, and A. Magrach, “Loss of pollinator diversity consistently reduces reproductive success for wild and cultivated plants,” *Nature Ecology & Evolution*, vol. 9, no. 2, pp. 296–313, 2025.
- [14] Q. Rumohr, C. U. Baden, M. Bergtold, M. T. Marx, J. Oellers, M. Schade, A. Toschki, and C. Maus, “Drivers and pressures behind insect decline in central and western europe based on long-term monitoring data,” *PLoS One*, vol. 18, no. 8, p. e0289565, 2023.
- [15] B. Subedi, A. Poudel, and S. Aryal, “The impact of climate change on insect pest biology and ecology: Implications for pest management strategies, crop production, and food security,” *Journal of Agriculture and Food Research*, vol. 14, p. 100733, 2023.
- [16] L. A. Dyer, A. M. Smilanich, Z. Gompert, and M. L. Forister, “Insect conservation, technological traps, and the fading arts of natural history and field ecology,” *Current Opinion in Insect Science*, vol. 66, p. 101261, 2024.
- [17] S. Kirby, C. C. Fortuin, N. S. Minaev, C. J. O’Neal, D. W. Held, C. J. Wang, G. M. Henry, J. W. Harris, I. G. P. de Souza, J. G. Hill, *et al.*, “Effectiveness of passive trapping and sweep netting for insect pollinator sampling in turf communities,” *Journal of Insect Conservation*, vol. 30, no. 3, p. 37, 2026.
- [18] R. R. Hohbein and C. J. Conway, “Pitfall traps: A review of methods for estimating arthropod abundance,” *Wildlife Society Bulletin*, vol. 42, no. 4, pp. 597–606, 2018.
- [19] G. A. Montgomery, M. W. Belitz, R. P. Guralnick, and M. W. Tingley, “Standards and best practices for monitoring and benchmarking insects,” *Frontiers in Ecology and Evolution*, vol. Volume 8 - 2020, 2021.
- [20] Q. Geissmann, P. K. Abram, D. Wu, C. H. Haney, and J. Carrillo, “Sticky pi is a high-frequency smart trap that enables the study of insect circadian activity under natural conditions,” *PLoS biology*, vol. 20, no. 7, p. e3001689, 2022.

- [21] K. Bjerger, J. B. Nielsen, M. V. Sepstrup, F. Helsing-Nielsen, and T. T. Høye, “An automated light trap to monitor moths (lepidoptera) using computer vision-based tracking and deep learning,” *Sensors*, vol. 21, no. 2, 2021.
- [22] W. Lord, S. Teagle, J. Alton, G. Bannister, J. Beuchert, K. Bjerger, D. Carbone, A. Gomez Segura, T. Howson, T. T. Høye, J. Lawson, A. Ravivarma, D. B. Roy, D. Rylett, G. Skinner, A. Warwick, and T. August, “Automated monitoring of insects system: Building a non-lethal and scalable solution for monitoring of nocturnal insects,” *bioRxiv*, 2026.
- [23] M. Gillespie, K. Bjerger, J. Alison, H. S. Gerlich, G. Mougeot, A. Svenning, S. Wogram, and T. Høye, “Automated insect monitoring with camera traps is transforming ecological understanding,” *EcoEvoRxiv*, Oct. 2025.
- [24] V. Holzhauer, D. Böttger, P. Bodesheim, and G. Brehm, “Do camera light traps for moths provide similar data as conventional funnel light traps?,” *Insect Conservation and Diversity*, vol. 19, no. 3, pp. 498–510, 2026.
- [25] K. Bjerger, J. Alison, M. Dyrmann, C. E. Frigaard, H. M. R. Mann, and T. T. Høye, “Accurate detection and identification of insects from camera trap images with deep learning,” *PLOS Sustainability and Transformation*, vol. 2, p. e0000051, Mar. 2023.
- [26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.
- [27] P. E. Serra-Marin, A. Solé-Ribalta, A. Lana, J. Borge-Holthoefer, S. Hervías-Parejo, and A. Traveset, “Comparative assessment of automated and manual monitoring in comprehensive plant–pollinator communities,” *Methods in Ecology and Evolution*, vol. 16, no. 12, pp. 2960–2978, 2025.
- [28] V. Ştefan, A. Workman, J. C. Cobain, D. Rakosy, and T. M. Knight, “Utilising affordable smartphones and open-source time-lapse photography for pollinator image collection and annotation,” *Journal of Pollination Ecology*, vol. 38, p. 1–21, Jan. 2025.
- [29] K. Bjerger, C. E. Frigaard, and H. Karstoft, “Object detection of small insects in time-lapse camera recordings,” *Sensors*, vol. 23, no. 16, 2023.
- [30] J. Alison, L. Pegoraro, J. Blair, Y. Cohen, B. Haest, J. Idec, J. Kamminga, J. Lawson, M. Li, L. Do Nascimento, C. Outhwaite, B. Rutschmann, M. Sittinger, M. Abarca, T. August, Q. Geissmann, R. Guralnick, G. Mougeot, V. Ştefan, K. Bjerger, M. Brydegaard, T. Høye, E. Knop, W. Kunin, M. Lürig, D. Roy, A. Svenning, and R. van Klink, “Insect monitoring without pitfalls: seven steps for robust insect sensing systems,” *EcoEvoRxiv*, Feb. 2026.

- [31] W. He, C. Wu, and S. Bensalem, “Box-based monitor approach for out-of-distribution detection in yolo: an exploratory study,” in *International Conference on Runtime Verification*, pp. 229–239, Springer, 2024.
- [32] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems*, vol. 28, 2015.
- [33] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.
- [34] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.
- [35] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *European conference on computer vision*, pp. 213–229, Springer, 2020.
- [36] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*, pp. 8748–8763, PmLR, 2021.
- [37] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *International conference on machine learning*, pp. 1597–1607, PmLR, 2020.
- [38] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.
- [39] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (J. Burstein, C. Doran, and T. Solorio, eds.), (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.
- [40] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, *et al.*, “Grounded language-image pre-training,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10965–10975, 2022.

- [41] J. F. Meier and T. Lüddecke, “Zero-shot multi-animal tracking in the wild,” *arXiv preprint arXiv:2511.02591*, 2025.
- [42] J. Devlin, F. MacFarlane, A. Karley, F. Manfredini, and D. Williams, “Detecting bees in cherry flowers using timelapse images and foundational models,” *Journal of Pollination Ecology*, 2026.
- [43] F. Liu, S. Rayeed, S. Stevens, A. East, C. H. Chiang, C. Lee, D. Yi, J. Yang, T. Naik, Z. Wang, *et al.*, “Beetleflow: An integrative deep learning pipeline for beetle image processing,” *arXiv preprint arXiv:2511.00255*, 2025.
- [44] T. Ren, Q. Jiang, S. Liu, Z. Zeng, W. Liu, H. Gao, H. Huang, Z. Ma, X. Jiang, Y. Chen, *et al.*, “Grounding dino 1.5: Advance the” edge” of open-set object detection,” *arXiv preprint arXiv:2405.10300*, 2024.
- [45] T. Ren, Y. Chen, Q. Jiang, Z. Zeng, Y. Xiong, W. Liu, Z. Ma, J. Shen, Y. Gao, X. Jiang, *et al.*, “Dino-x: A unified vision model for open-world object detection and understanding,” *arXiv preprint arXiv:2411.14347*, 2024.
- [46] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, *et al.*, “Sam 2: Segment anything in images and videos,” *arXiv preprint arXiv:2408.00714*, 2024.
- [47] D. Bolya, P.-Y. Huang, P. Sun, J. H. Cho, A. Madotto, C. Wei, T. Ma, J. Zhi, J. Rajasegaran, H. Rasheed, *et al.*, “Perception encoder: The best visual embeddings are not at the output of the network,” *arXiv preprint arXiv:2504.13181*, 2025.
- [48] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, “Emerging properties in self-supervised vision transformers,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.
- [49] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, “Dinov2: Learning robust visual features without supervision,” *arXiv preprint arXiv:2304.07193*, 2023.
- [50] O. Siméoni, H. V. Vo, M. Seitzer, F. Baldassarre, M. Oquab, C. Jose, V. Khalidov, M. Szafraniec, S. Yi, M. Ramamonjisoa, *et al.*, “Dinov3,” *arXiv preprint arXiv:2508.10104*, 2025.
- [51] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, “Scene parsing through ade20k dataset,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 633–641, 2017.
- [52] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, “Semantic understanding of scenes through the ade20k dataset,” *International journal of computer vision*, vol. 127, no. 3, pp. 302–321, 2019.

- [53] M. Everingham, L. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [54] J. Gu, S. Stevens, E. G. Campolongo, M. J. Thompson, N. Zhang, J. Wu, A. Kopanев, Z. Mai, A. E. White, J. Balhoff, W. M. Dahdul, D. Rubenstein, H. Lapp, T. Berger-Wolf, W.-L. Chao, and Y. Su, “BioCLIP 2.5 Huge,” 2026.
- [55] J. Gu, S. Stevens, E. G. Campolongo, M. J. Thompson, N. Zhang, J. Wu, A. Kopanев, Z. Mai, A. E. White, J. Balhoff, *et al.*, “Bioclip 2: Emergent properties from scaling hierarchical contrastive learning,” *arXiv preprint arXiv:2505.23883*, 2025.
- [56] C. Schuhmann, R. Beaumont, R. Vencu, C. W. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. R. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev, “LAION-5b: An open large-scale dataset for training next generation image-text models,” in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.
- [57] K. Bjerge, J. Alison, M. Dyrmann, C. E. Frigaard, H. M. R. Mann, and T. T. Høye, “Benchmark datasets for detection and identification of insects from camera trap images with deep learning,” 2022.
- [58] T. Stark, V. Ştefan, M. Wurm, R. Spanier, H. Taubenböck, and T. M. Knight, “Yolo object detection models can locate and classify broad groups of flower-visiting arthropods in images,” *Scientific Reports*, vol. 13, Sept. 2023.
- [59] V. Ştefan, T. Stark, M. Wurm, H. Taubenböck, and T. M. Knight, “Successes and limitations of pretrained yolo detectors applied to unseen time-lapse images for automated pollinator monitoring,” *Scientific Reports*, vol. 15, Aug. 2025.
- [60] V. Ştefan, A. Workman, J. C. Cobain, D. Rakosy, B. Wild Stoykova, E. Cyranka, R. Urrego Álvarez, and T. Knight, “Dataset of arthropod flower visits captured via smartphone time-lapse photography,” Mar. 2025.
- [61] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 740–755, Springer International Publishing, 2014.
- [62] Y. Wu, “cocoapi: Fast and correct COCO API for Python.” <https://github.com/ppwwyyxx/cocoapi>, 2019. Accessed: 2026-04-09.

- [63] H. Markoff, S. H. Bengtson, and M. Ørsted, “Vision transformers for zero-shot clustering of animal images: A comparative benchmarking study,” *arXiv preprint arXiv:2602.03894*, 2026.
- [64] R. Hu and J. Liu, “Binary verification for zero-shot vision,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11498–11506, 2026.
- [65] C. Zeng, Y. Jiang, and A. Zhang, “Efficientsam3: Progressive hierarchical distillation for video concept segmentation from sam1, 2, and 3,” *arXiv preprint arXiv:2511.15833*, 2025.
- [66] C. Zeng, Y. Jiang, G. Gao, S. Wang, D. Danier, B. Zhu, S. Rudinac, D. Bull, and F. Zhang, “Sam3-litetext: An anatomical study of the sam3 text encoder for efficient vision-language segmentation,” *arXiv preprint arXiv:2602.12173*, 2026.
- [67] H. Touvron, A. Vedaldi, M. Douze, and H. Jégou, “Fixing the train-test resolution discrepancy,” *Advances in neural information processing systems*, vol. 32, 2019.