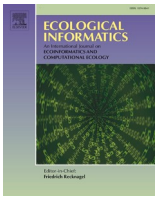
Ecological Informatics 78 (2023) 102384

Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf



A deep learning-based pipeline for whitefly pest abundance estimation on chromotropic sticky traps

Luca Ciampi [a,*], Valeria Zeni [b], Luca Incrocci [b], Angelo Canale [b], Giovanni Benelli [b], Fabrizio Falchi [a], Giuseppe Amato [a], Stefano Chessa [c]

[a] Institute of Information Science and Technologies of the National Research Council of Italy (ISTI-CNR), Pisa, Italy
[b] Department of Agriculture, Food and Environment of the University of Pisa, Pisa, Italy
[c] Department of Computer Science of the University of Pisa, Pisa, Italy

ARTICLE INFO

Keywords:
Smart agriculture
Smart farming
Integrated pest management
Computer vision
Object counting
Visual counting

ABSTRACT

Integrated Pest Management (IPM) is an essential approach used in smart agriculture to manage pest populations and sustainably optimize crop production. One of the cornerstones underlying IPM solutions is pest monitoring, a practice often performed by farm owners by using chromotropic sticky traps placed on insect hot spots to gauge pest population densities. In this paper, we propose a modular model-agnostic deep learning-based counting pipeline for estimating the number of insects present in pictures of chromotropic sticky traps, thus reducing the need for manual trap inspections and minimizing human effort. Additionally, our solution generates a set of raw positions of the counted insects and confidence scores expressing their reliability, allowing practitioners to filter out unreliable predictions. We train and assess our technique by exploiting PST - Pest Sticky Traps, a new collection of dot-annotated images we created on purpose and we publicly release, suitable for counting whiteflies. Experimental evaluation shows that our proposed counting strategy can be a valuable Artificial Intelligence-based tool to help farm owners to control pest outbreaks and prevent crop damages effectively. Specifically, our solution achieves an average counting error of approximately 9% compared to human capabilities requiring a matter of seconds, a large improvement respecting the time-intensive process of manual human inspections, which often take hours or even days.

1. Introduction

Digitalization and automatization have become customary practices in various domains, encompassing industries, entertainment, environment, and overall human well-being. The agricultural and the ecological sectors are not excluded from this trend. Smart agriculture, also called smart farming or digital agriculture in the literature, represents the advancement of precision agriculture (Kamilaris et al., 2016) and holds the potential to revolutionize food management and production by addressing the strategic challenges of this area concerning productivity, food security, environmental impact, and sustainability. Specifically, it aims to convert traditional agriculture techniques into innovative solutions that leverage Information and Communication Technologies (ICT) such as machine learning, automated image processing, unmanned aerial vehicles, unmanned ground vehicles, and wireless sensor networks (Kamilaris et al., 2016; Mahmud et al., 2023; Moazzam et al., 2023; Tian et al., 2020).

An essential strategy in farm management is Integrated Pest Management (IPM), which aims to control pest populations effectively. An effective IPM tool can prevent crop damage (Lamichhane et al., 2016) and suggest corrective measures to avoid pests from causing significant problems, keeping the use of pesticides only to levels that are ecologically justified,[1] and consequently minimizing risks or hazards to humans and the environment. On the other hand, recent trends are revisiting the IPM paradigm, focusing more on ecological aspects and putting the environment as a pivotal element (Dara, 2019), keeping and supporting the natural stability of the agro-ecosystem (Tshernyshev, 1995), e.g., by increasing biodiversity and creating habitat for natural enemies, without using chemical compounds. One of the pillars underpinning IPM solutions is monitoring insects, mites, and other living organisms near the crops, identifying them accurately to ensure adequate control decisions and actions. In practice, chromotropic sticky traps are among the most

* Corresponding author.
E-mail address: luca.ciampi@isti.cnr.it (L. Ciampi).

[1] https://food.ec.europa.eu/plants/pesticides/sustainable-use-pesticides/integrated-pest-management-ipm_en

https://doi.org/10.1016/j.ecoinf.2023.102384
Received 1 August 2023; Received in revised form 16 October 2023; Accepted 16 November 2023
Available online 25 November 2023

common methods utilized by farm owners to assess pest population densities. However, trap inspections are labor-intensive, subjective, costly, and prone to high error rates due to human susceptibility to factors like fatigue, visual illusions, and boredom (Barbedo, 2014). This opens the door for automated solutions within the realm of smart agriculture.

This paper proposes an automated counting pipeline based on data-driven Artificial Intelligence (AI), specifically Deep Learning (DL), for estimating the number of pests in images of sticky chromotropic traps. Our approach follows a modular paradigm and is model-agnostic: differently from most existing works that employ specific object detectors, the module responsible for counting can be implemented with recent SOTA methodologies, not only detection-based but also relying on regression. Its output is then fed into downstream modules that produce unified outputs expressing localization and confidence scores of the counted insects. The required data was collected by taking digital camera pictures of the traps placed in insect hot spot locations at the University of Pisa (Italy). Subsequently, images were annotated by putting dots over the centroids of the trapped insects of interest; dotting emulates the natural human technique for counting objects (at least when the number of objects is greater than the subitizing range), and it represents the golden standard concerning the labels needed for the supervised training of deep learning models for the counting task (Lempitsky and Zisserman, 2010). We name this collection of images PST - Pest Sticky Traps and publicly release it (Ciampi et al., 2023). To the best of our knowledge, it is the first publicly available dot-annotated dataset specifically tailored for counting pests on sticky trap images, and it represents another novel contribution of our work. In this setting, we experiment with several approaches: our best-performing solution achieves an average counting error of approximately 9% compared to human capabilities while requiring mere seconds for computation, in contrast to the hours or days needed for manual human inspections.

In summary, the main contributions of this work are listed in the following.

- We propose a deep learning-based pipeline to automatically estimate pest populations present on chromotropic sticky trap pictures. The proposed solution is modular and model-agnostic, allowing for seamless integration of subsequent models from the literature into the proposed pipeline, and therefore capable of encompassing all counting strategies; furthermore, it works in the wild, i.e., any pre-processing strategy of the pictures is needed, such as preliminary segmenting the region outlining the traps from the background.
- Our proposed pipeline, despite the upstream counting approach, can also produce a set of localization providing the raw positions of the counted insects and associated confidence scores expressing their reliability, thus permitting practitioners to filter out unreliable predictions.
- We introduce PST - Pest Sticky Traps, a collection of dot-labeled images depicting sticky traps where pests have been trapped; it represents the first dataset specifically tailored for counting insects freely available to the scientific community (Ciampi et al., 2023), suitable for training and testing deep learning models in recognizing and counting whiteflies.
- We conduct an extensive experimental evaluation of the proposed pipeline considering several counting approaches, ranging from detection-based techniques to regression methodologies; differently from most existing works that assessed models' performance only in terms of detection, we exploit pure counting metrics and also some hybrid evaluators that simultaneously consider the object count and the estimated raw locations of them, showing that our counting strategy can be a reliable AI-based IPM tool.

We organize the rest of the study as follows. We report related work in Section 2. We illustrate the proposed dataset in Section 3. In Section 4, we describe the methodology, while in Section 5, we outline the performed experiments discussing the obtained results. Finally, Section 6 concludes the research. The code and other resources are publicly available at https://ciampluca.github.io/sticky_trap_pest_counting/.

2. Related works

We report a comprehensive review of previous studies that are relevant to our research. Specifically, we explore works related to the counting task in a broad sense, as well as approaches designed explicitly for detecting and estimating the number of insects from images.

2.1. Counting objects in images

Object counting is one of the more challenging tasks in computer vision, offering a plethora of practical applications. Some examples range from counting vehicles (Ciampi et al., 2022b; Guerrero-Gómez-Olmedo et al., 2015) or counting cells (Ciampi et al., 2022a; Xie et al., 2016) to estimating the number of trees (Putra et al., 2022), plant leaves (Bhagat et al., 2022) or people (Benedetto et al., 2022; Liu et al., 2019). State-of-the-art performances are achieved by exploiting supervised deep learning approaches that basically follow two main strategies: counting through detection and counting through regression. The former, exemplified by works like (Amato et al., 2019; Hsieh et al., 2017), mandates preliminary identification of individual object instances. In contrast, regression-based techniques (Arteta et al., 2016) learn a mapping between image feature representations and the number of objects present in the scene, circumventing explicit localization of object instances. This can be achieved directly or through estimating a target map, such as a density or segmentation map representing a real or integer-valued (non-linear) function, respectively. Regression methods have demonstrated particular efficacy in densely populated scenarios where occlusions often obstruct object visibility. However, a limitation is their failure to furnish precise localization of objects within the scene.

2.2. Insect detection and counting

Recently, AI and, specifically, DL techniques have supplanted traditional computer vision approaches for pest detection, as seen in works such as (Costa et al., 2023; Kasinathan et al., 2021; Sun et al., 2017). Notable deep learning solutions designed specifically for detecting pests in images of sticky traps include (Wang et al., 2021), (Nieuwenhuizen et al., 2018), (Khalid et al., 2023), and (Li et al., 2021a). These works presented three approaches based on popular object detectors like YOLOv4 (Bochkovskiy et al., 2020), YOLOv8 (Jocher et al., 2023), and Faster R-CNN (Ren et al., 2017), where the authors experimented with datasets of sticky traps containing bounding box annotations for whiteflies and other arthropods. For comprehensive surveys on pest classification and detection in images, references (Li et al., 2021b; Lima et al., 2020) can be consulted.

Additionally, within the existing literature, numerous deep learning-based approaches claim to address the specific challenge of pest counting. For instance, in (Ding and Taylor, 2016), the authors proposed an automatic detection framework for localizing and counting pests from pictures of field traps via a classifier applied to local windows at different positions within the entire image. The experimental evaluation has been conducted using a commercial codling moth dataset. Similarly, (Sun et al., 2018) introduced a network based on the RetinaNet object detector (Lin et al., 2017) for detecting and counting red turpentine beetles in images of pheromone traps. The results are obtained by testing

the model on a bounding box-annotated dataset with a relatively small number of insects per image (approximately six on average). In another work (Partel et al., 2019), Partel et al. proposed an automated system utilizing an AI-based approach running on an Nvidia TX2 board to detect, classify, and count the Asian citrus psyllid, *Diaphorina citri* Kuwayama (Hemiptera: Liviidae), through image acquisition and processing in a citrus grove. Similarly, (Lins et al., 2020) introduced a method for detecting and classifying *Rhopalosiphum padi* (L.) (Hemiptera: Aphididae) using hand-crafted computer vision techniques for localization and deep learning for classification. Additionally, Li et al. (Li et al., 2019) proposed a pipeline based on the Faster R-CNN detector (Ren et al., 2017) for detecting and counting agricultural pests in images, while (Wang et al., 2022) employed a similar architecture for counting apple pests on sticky boards. Finally, in (She et al., 2022), a YOLO-based object detector is exploited for counting fruit fly pests in orchards on trap bottles. However, all these methods fall under the counting-by-detection strategy, relying on object detectors. It is important to note that while these works evaluate detection performance, they do not extensively analyze counting metrics, despite counting being the primary task for which they were designed.

Only a few works directly address counting analysis or employ alternative approaches not based on detectors. For example, Zhong et al. (Zhong et al., 2018) and Rustia et al. (Rustia et al., 2020) proposed cascade approaches using the YOLOv1 (Redmon et al., 2016) and YOLOv3 (Redmon and Farhadi, 2018) object detectors, followed by a support vector machine and a multi-class convolutional neural network-based classifier, respectively, to localize and count various types of insects automatically. Their methodologies are evaluated in terms of mean counting accuracy using non-publicly available collections of sticky trap images. Finally, (Bereciartua-Pérez et al., 2022; Bereciartua-Pérez et al., 2023) presents a regression-based approach using a Fully Convolutional Regression Network (FCRN) inspired by (Xie et al., 2016) to count whiteflies in eggplant leaves through density map estimation. The evaluation is performed using counting metrics, again, on a non-publicly available dataset.

In this work, we propose a counting pipeline that estimates the number of whiteflies by analyzing images of sticky traps. Differently from most existing works (Ding and Taylor, 2016; Li et al., 2019; Li et al., 2021a; Lins et al., 2020; Nieuwenhuizen et al., 2018; Sun et al., 2018; Wang et al., 2021), we provide a detailed analysis in terms of counting metrics. We leverage counting golden standard evaluators and hybrid metrics that consider both the number of insects and their estimated locations within the images. Additionally, our approach includes a model-agnostic module that can be implemented using various state-of-the-art deep learning algorithms. We evaluate several methods within the counting-by-detection strategy, encompassing different object detectors used in previous works (Bochkovskiy et al., 2020; Lin et al., 2017; Redmon et al., 2016; Redmon and Farhadi, 2018; Ren et al., 2017). Moreover, we assess regression-based techniques, which have been less explored in previous studies (Bereciartua-Pérez et al., 2022; Bereciartua-Pérez et al., 2023), for estimating pest populations. To evaluate our solution, we present a novel and challenging dataset that we publicly provide (Ciampi et al., 2023), contributing to the scientific community by establishing a baseline for reproducibility and comparison of results.

3. The dataset

Whiteflies (Hemiptera: Aleyrodidae) are notorious pests that pose a significant threat to numerous agricultural crops. They are recognized as a primary contributing factor to the decline in various greenhouse-grown vegetable crops (Farina et al., 2022). Specifically, they diminish plant health and production by directly collecting plant sap with their sucking stinging mouthparts and triggering the spread of sooty mold fungi by releasing sticky honeydew on fruits (Sekine et al., 2022). Furthermore, they are not only responsible for direct damaging plants but also acting as vectors for several tomato viruses, such as the tomato yellow leaf curl virus (TYLCV) (Ramasamy and Ravishankar, 2018; Tan et al., 2017). Hence, it is not surprising that we are witnessing an increasing interest in designing automated IPM solutions to monitor the population of these living organisms so that appropriate control decisions can be made.

Table 1
Dataset Statistics. We report some numerical characteristics of the proposed PST - Pest Sticky Traps dataset.

#images	image size	#insects	#insects/image		
			avg	min	max
28	4288 × 2848	17,005	607	1	3686

In this work, we collect and publicly release PST - Pest Sticky Traps (Ciampi et al., 2023), a novel dot-annotated dataset containing pictures of yellow chromotropic sticky traps specifically designed for training/testing deep learning models to automatically count insects and estimate pest populations. Other collections of images depicting pest traps are already present in the literature (Bereciartua-Pérez et al., 2022; Bereciartua-Pérez et al., 2023; Li et al., 2021a; Nieuwenhuizen et al., 2018; Sun et al., 2017; Wang et al., 2020; Wang et al., 2021; Zhong et al., 2018). However, they are often not publicly available and are all specifically tailored for the pest detection task, i.e., they provide bounding box labels. Although it is possible to cast these data as suitable for counting, they inherit some limitations due to their original purpose. Indeed, the manual process of annotating bounding boxes demands more human effort than dotting, which is a more intuitive method to count objects (Lempitsky and Zisserman, 2010). As a consequence, most existing datasets exhibit just a tiny number of insects per-image (Sun et al., 2018; Wang et al., 2020), or they include tiles of the original images obtained after data pre-processing because pest annotations were more readily performed using smaller regions of interest, i.e., tiles contained lower numbers of insects than the full images (Li et al., 2021a; Nieuwenhuizen et al., 2018; Wang et al., 2021). On the other hand, the PST dataset is tailored for pest counting. It includes images of full sticky trap pictures that deep learning models can use in the wild, i.e., without needing any pre-processing step. Data is annotated with dots rawly localizing the insects, making possible the presence of single challenging images containing thousands of insects. In the following, we describe the data collection and curation phases.

Data Collection. The traps have been preliminarily placed by some experts in several insect hotspot locations at the University of Pisa (Italy) during periods of time for which it was expected the presence of several species of whiteflies. Then, pictures were acquired using a Nikon D5300 digital camera in natural light conditions at different times of the day to have heterogeneous scenarios in terms of different illuminations. Ultimately, the dataset includes about 30 images with sizes of 4288 × 2848.

Data Curation. Images were manually annotated by some experts of the Department of Agriculture, Food and Environment of the University of Pisa (Italy) by putting a dot over each identified insect using the freely available *LabelMe* annotation tool.[2] Specifically, we labeled insects as belonging to the category "whitefly" considering two different species, i.e., the sweet potato whitefly (*Bemisia tabaci*) (Gennadius) and the greenhouse whitefly (*Trialeurodes vaporariorum*) (Westwood). Among all the images, the number of insects dramatically varies from a few to some thousand per image, for a total of more than 17,000 labeled insects. We report detailed statistics about the dataset in Table 1, while, in Fig. 1, we provide some samples of it, together with zoomed areas of a single sticky trap showing different densities of captured insects. We divided the dataset into two splits: a training subset of 20 images for the model

[2] https://github.com/wkentaro/labelme

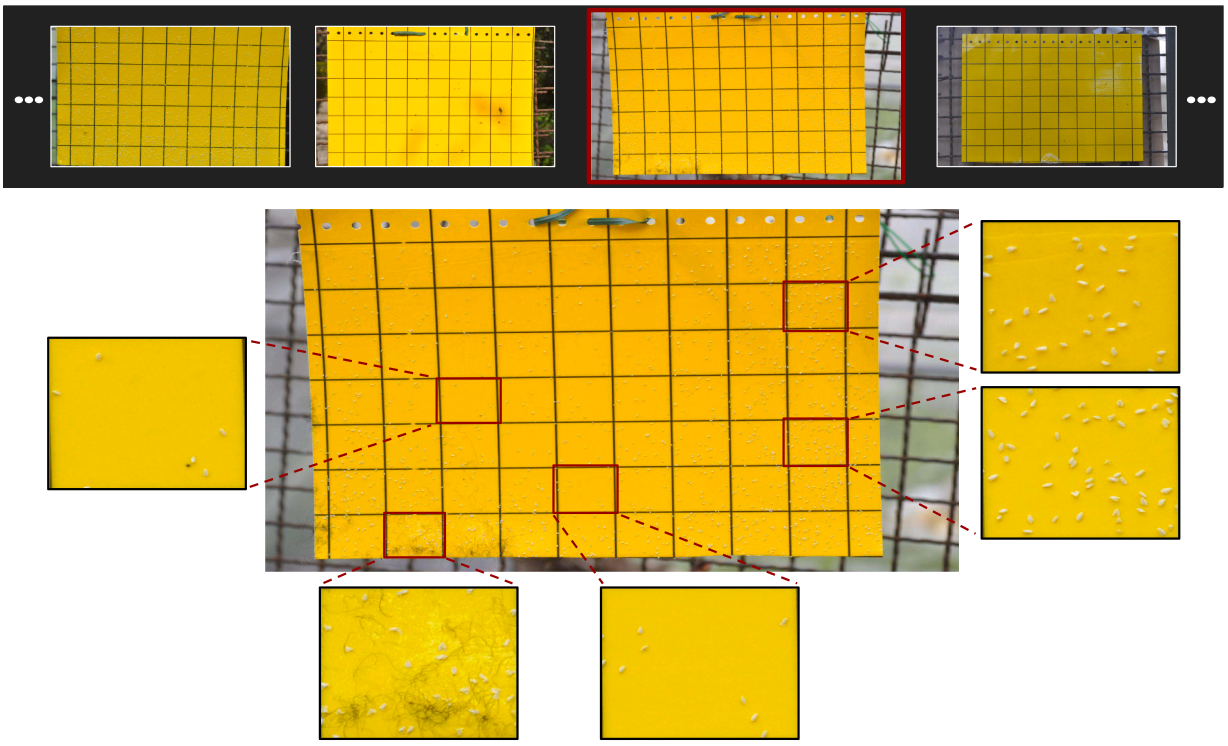


Fig. 1. Dataset Samples. We report some samples of our PST - Pest Sticky Traps dataset, showing pictures gathered at different illuminations. We also depict in detail one sample, zooming some areas of it having different densities of captured insects and other contaminants accidentally glued, such as fluff and dust. Best viewed in electronic format.

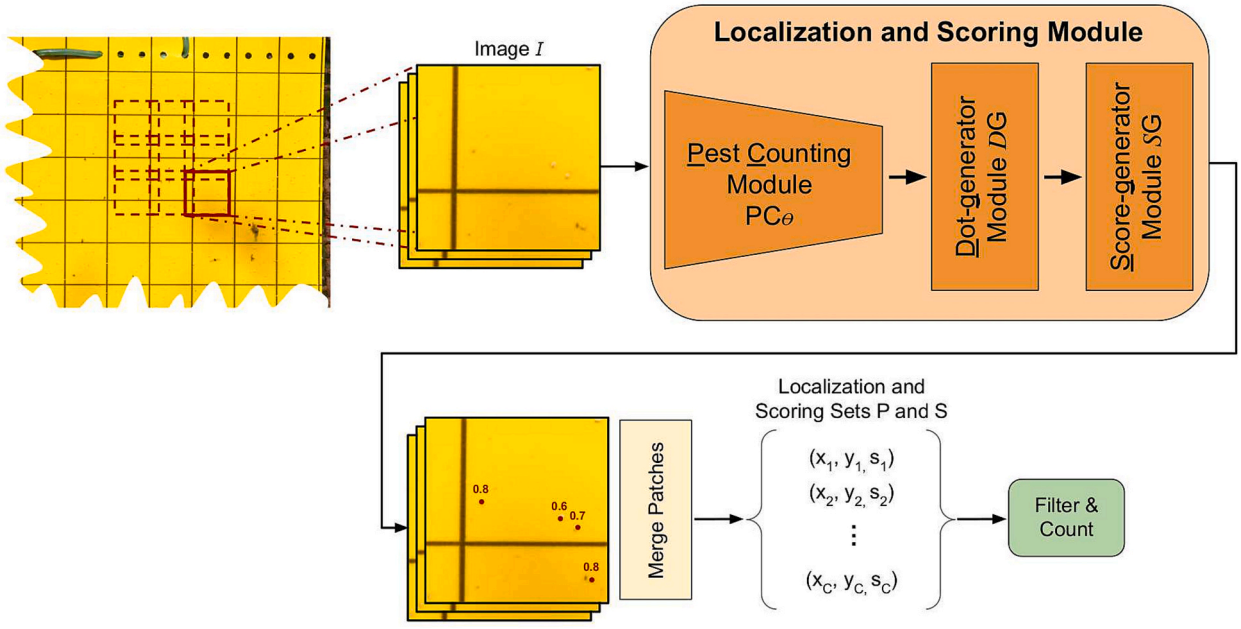


Fig. 2. The proposed counting pipeline. We design a *Localization and Scoring Module* that, from an input image I, produces (i) a set of 2D dot coordinates P, roughly localizing the detected insects, and (ii) a set of confidence scores S assigned to each of them. Specifically, it comprises a pest counting module PC_θ based on deep learning techniques, a dot-generator module DG, and a score-generator module SG. We design, implement, and test the *Localization and Scoring Module* in various ways, aligning with the primary counting approaches found in the existing literature, namely counting through detection and counting through regression. By adjusting a customizable threshold, practitioners can effectively filter out unreliable predictions and obtain the final count of the detected insects.

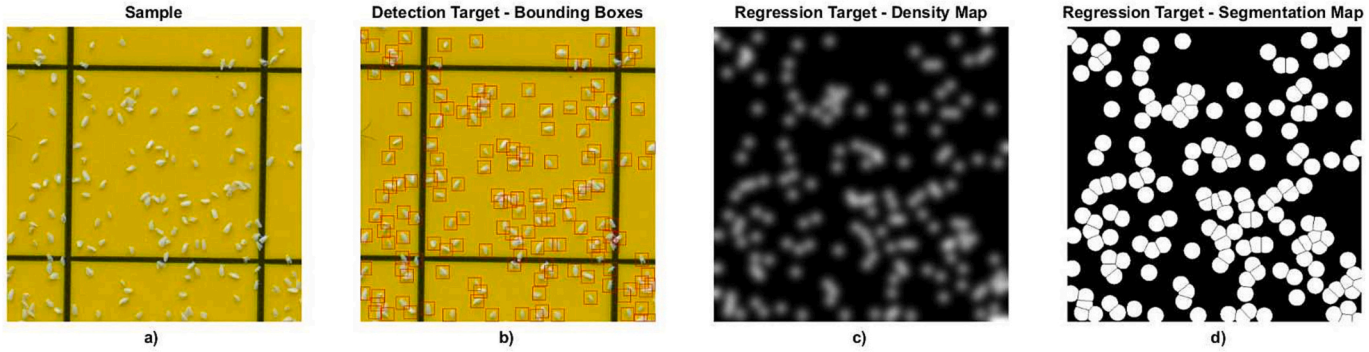


Fig. 3. Generated ground-truth. Ground-truth required for the training phase is derived from dot labels by leveraging distinct procedures depending on the specific deep learning model PC_θ being employed:: (i) bounding boxes are generated by creating squares centered over the dots with a fixed side length s, (ii) density maps are produced by overlaying Gaussian kernels G_σ centered at the dot positions, and (iii) segmentation maps are generated by superimposing dot-centered discs having radius r and splitting overlapping discs using a background ridge. The parameters s, σ, and r are constants that are established based on the standard object sizes prevalent in the images. Best viewed in electronic format.

training step and a test split with the remaining eight images for the model evaluation phase. Finally, it is worth noting that, even if we annotated only the pests belonging to the class "whitefly", other insects were accidentally stucked in the traps, which represents an additional challenge - deep learning models should not count the latter insect species. Specifically, we found insects belonging to different orders, such as Lepidoptera, Hymenoptera, and Diptera (about 21, 7, and 113, respectively); furthermore, among hemipterans we also found insects belonging to families of Psyllidae, Aphidae, and Miridae (about 4, 14, and 22, respectively). Besides, about 80% of the images contain contaminants, such as dust, fluff, or leaves, accidentally glued over the sticky traps. Some samples of bycatches insects and contaminants can be found in Fig. 1 and Fig. 4.

4. A deep learning-based pipeline for pest population estimation

4.1. Overview

We make the assumption to have a collection of N annotated images of chromotropic sticky traps denoted as $\mathscr{X} = (I_1, \widehat{P}_1), \ldots, (I_N, \widehat{P}_N)$, where I_i indicates an image and $\widehat{P}_i \subset \mathbb{R}^2$ the associated set of annotations, i.e., the ground truth. Here, following the standards for the counting task, the latter corresponds to 2D point coordinates roughly localizing the insects to be counted within the image.

Fig. 2 depicts a graphical overview of the overall proposed counting pipeline. Specifically, it comprises a *Localization and Scoring Module* that is fed with an image I_i, and it outputs a set $P_i = \{p_1, \ldots, p_C\} \in \mathbb{R}^2$ of 2D point coordinates roughly localizing the detected insects together with a set $S_i = \{s_1, \ldots, s_C\} \in \mathbb{R}$ of confidence scores expressing their reliability. More in detail, this module comprises three components: (i) a pest counting module PC_θ based on deep learning techniques, (ii) a dot-generator module DG, and (iii) a score-generator module SG.

The pest counting module PC_θ generates an estimation of the insect count within a given Region of Interest (RoI). It utilizes a deep learning technique that is trained by leveraging the data $\mathscr{X}$. In this work, we implement PC_θ using multiple networks that belong to the main counting strategies found in the literature. These strategies, as already mentioned in Section 2.1, include counting by detection, where we compute bounding boxes, and counting by regression, where we predict a target map. Specifically, for the regression-based approach, we explore two different methods that rely on distinct regressed target maps: counting by density estimation and counting by segmentation. Density maps represent the insect distribution, where each pixel indicates the "amount" of insects present at that specific location, while in segmentation maps, each pixel is assigned to a specific object class, typically with zeros representing the background class. On the other hand, the dot-generator module DG receives the output from the preceding pest counting module PC_θ. Its purpose is to generate the set of dot coordinates P that provide an approximate localization of the detected insects within the Region of Interest (RoI). Hence, we design this module in three variations, each tailored to the specific characteristics of the upstream pest counting module PC_θ. Subsequently, the score-generator module SG assigns a confidence score to indicate the level of reliability for each prediction. This module enables users to eliminate predictions of poor quality and obtain the final count of identified objects. Once again, the design of this module aligns with the different formulations of the upstream pest counting module PC_θ.

Below, we provide an overview of the three localization and scoring modules, which correspond to the three counting approaches they are built upon. Firstly, Section 4.2 describes the module that utilizes detection with bounding boxes. Secondly, Section 4.3 explains the module that relies on regression to density maps. Lastly, Section 4.4 showcases the module that is based on regression to binary segmentation maps.

4.2. Localization and scoring module based on detection with bounding boxes

In the approach that utilizes detection with bounding boxes, the module PC_θ consists of a state-of-the-art object detector that generates a set of bounding boxes to localize the predicted insects. Meanwhile, the final localization set P is created by the dot-generator module DG simply by computing the centers of these boxes. The confidence scores, finally, are provided natively by the detector and range from 0 to 1, representing the probability of an object being present within the bounding box. To train PC_θ, labels are generated using squared bounding boxes centered on the ground-truth dots. The side length of the bounding boxes is fixed and determined in advance, considering the typical sizes of objects in the dataset. An example of a ground-truth sample can be seen in column b) of Fig. 3.

In this paper, we utilize two different detectors implementing PC_θ. The first one is Faster-RCNN (Ren et al., 2017), a widely used network that operates under a two-stage paradigm. The initial stage aims to generate region proposals – plausible areas containing objects – using predefined boxes referred to as anchors. The subsequent stage fine-

tunes, categorizes, and assigns confidence scores to these regions. Specifically, we utilize the Faster-RCNN version with a ResNet-50 backbone (He et al., 2016). Henceforth, we will refer to this method as *Det-FRCNN*. On the other hand, our second detector is FCOS (Tian et al., 2019), a recently introduced fully convolutional network that follows a one-stage paradigm. It eliminates the need for anchor boxes and relies solely on post-processing techniques like non-maximum suppression, a popular procedure essential for removing redundant overlapping detections by merging bounding boxes that might belong to the same object (Felzenszwalb et al., 2010; Ren et al., 2017). Consistently, we use the FCOS detector with a ResNet-50 backbone (He et al., 2016) and will refer to it as *Det-FCOS*.

4.3. Localization and scoring module based on regression to density maps

Using this approach, PC_θ is designed as a density estimator, which is particularly effective for counting in densely populated scenarios. The objective is to learn a non-linear mapping between the input image's features and an associated density map D_{map}. In this density map, each pixel represents the "amount" of objects present at that specific location, in accordance with the concept of density from both physical and mathematical standpoints. The number of objects C within the image I, or a sub-region of it $I_{sub} \subseteq I$, is calculated by summing the pixel values within the considered area, i.e., $C = \sum_{p \in I_{sub}} D_{map_p}$. Given that an inherent drawback of this approach is its inability to achieve precise object localization, the responsibility of offering a rough positioning of the insects falls upon the dot-generator module *DG*, which analyzes the estimated density map, determining the $x - y$ coordinates of the top-C local peaks with the highest values (Xie et al., 2016). Finally, the confidence scores from the *SG* module are derived by considering the pixel values of these top-C local peaks. For training PC_θ, the generation of ground-truth density maps involves overlaying Gaussian kernels G_σ centered on the ground-truth dot-annotated locations. The parameter σ, determining the Gaussian spread, is established based on the standard sizes of insects gauged in the dataset. An example of a ground-truth density map can be seen in column c) of Fig. 3. Specifically, PC_θ is trained to minimize the mean squared error loss between the predicted and the ground-truth density maps.

In this context, we use two density estimators to represent PC_θ. The first one is the Congested Scene Recognition Network (CSRNet) (Li et al., 2018), initially designed for crowd density estimation but subsequently adapted for counting other object categories such as cells (Xie et al., 2016) and vehicles (Guerrero-Gómez-Olmedo et al., 2015). It employs a modified version of the well-known VGG-16 network (Simonyan and Zisserman, 2015) for feature extraction and utilizes dilated convolutional layers (Yu and Koltun, 2016) to capture deeper saliency information while maintaining the output resolution. We will refer to this method as *Den-CSRNet*. The second density estimator we employ is based on FCRN (Xie et al., 2016), a fully convolutional neural network initially used for cell counting and then applied to count whiteflies in eggplant leaves (Bereciartua-Pérez et al., 2022; Bereciartua-Pérez et al., 2023). Henceforth, we will refer to this method as *Den-FCRN*.

4.4. Localization and scoring module based on regression to segmentation maps

In this approach, inspired by Falk et al. (Falk et al., 2018), we train PC_θ to learn a non-linear mapping between the input image's features and an associated binary segmentation map $S_{map} \in 0, 1$. In this map, pixels with a value of one represent the objects of interest, while zeros correspond to the background. The segmentation map S_{map} is then thresholded and processed further to extract connected components. The dot-generator module *DG* generates the localization set P by computing the x-y coordinates of the centroids of these connected components. Finally, the confidence scores in S are calculated by the score-generator module *SG*. Each score value ranges from 0 to 1, corresponding to the maximum value within the connected components associated with each localized object. Following Falk et al. (Falk et al., 2018), the ground-truth segmentation maps for training PC_θ were created by superimposing a disc centered at the ground-truth locations and separating overlapping discs with a background ridge. The disc radius is fixed based on the typical insect sizes in the dataset. An example of a ground-truth segmentation map can be seen in column d) of Fig. 3. Specifically, PC_θ is trained to minimize the weighted binary cross-entropy loss between the predicted and the ground-truth segmentation maps, assigning more weight to the pixel along the ridges separating instances.

In line with Falk et al. (Falk et al., 2018), we implement PC_θ using a standard U-Net architecture (Ronneberger et al., 2015). We will refer to this approach as *Seg-UNet*.

5. Experiments and results

5.1. Overview

We assess our methodology against the PST dataset described in Section 3. We utilize the training subset for the supervised learning step concerning PC_θ, while the overall counting pipeline is evaluated on the test split. First, we investigate the counting ability of the proposed approaches using the counting golden standard evaluators: basically, we investigate the counting performance of our pipeline considering only the pest counting module PC_θ. Then, we perform further experiments taking into account some hybrid metrics that simultaneously consider the count error and the estimated raw locations of the insects: essentially, in this case, we consider the entire pipeline, i.e., the PC_θ, *DG*, and *SG* modules. It is worth noting that The PC_θ modules discussed in Section 4 encompass various general counting strategies, including detection and regression. These strategies are drawn from state-of-the-art methodologies for pest detection and counting, as mentioned in Section 2.2. Notably, many of these methodologies employ the same deep learning network, such as Faster R-CNN (Ren et al., 2017), which is adopted in constructing the PC_θ module. Some other approaches, like (Bereciartua-Pérez et al., 2022; Bereciartua-Pérez et al., 2023) inspired by FCRN (Xie et al., 2016), and others like (Rustia et al., 2020; Sun et al., 2018; Wang et al., 2021; Zhong et al., 2018), which utilize similar architectures to (Tian et al., 2019), are also considered. Summarizing, this section aims to demonstrate, over the newly established scenario represented by the PST dataset, the effectiveness of our counting pipeline; taking advantage of the latter being model-agnostic, we test several general counting strategies that also enclose primary SOTA pest detection and counting approaches, performing an in-depth counting analysis.

5.2. Experimental settings

Training Setup. In the training phase, we partition the training data into two subsets: training and validation. To mitigate potentially unbalanced sets arising from the varying insect numbers present in different images, we deviate from the typical per-image splitting approach. Rather, we embrace an alternative strategy. We vertically divide each image into halves, alternatingly assigning one portion to the training subset and the other to the validation set.

We process images in the wild, i.e., without performing any preprocessing operation, such as segmenting the region representing the trap from the background. To address the challenge of large image sizes,

Table 2
Evaluation of the Counting Performance. We show the obtained outcomes in terms of Mean Absolute Error (MAE), Mean Squared Error (MSE), and Mean Absolute Relative Error (MARE).

Method	MAE ↓	MSE ↓	MARE (%) ↓
Det-FRCNN (Ren et al., 2017)	**65.750**	**8532.125**	**9.3**
Det-FCOS (Tian et al., 2019)	66.125	9328.500	9.6
Den-CSRNet (Li et al., 2018)	179.875	110,914.875	76.1
Den-FCRN (Xie et al., 2016)	126.500	49,710.625	60.5
Seg-UNet (Falk et al., 2018)	80.375	12,853.625	10.7

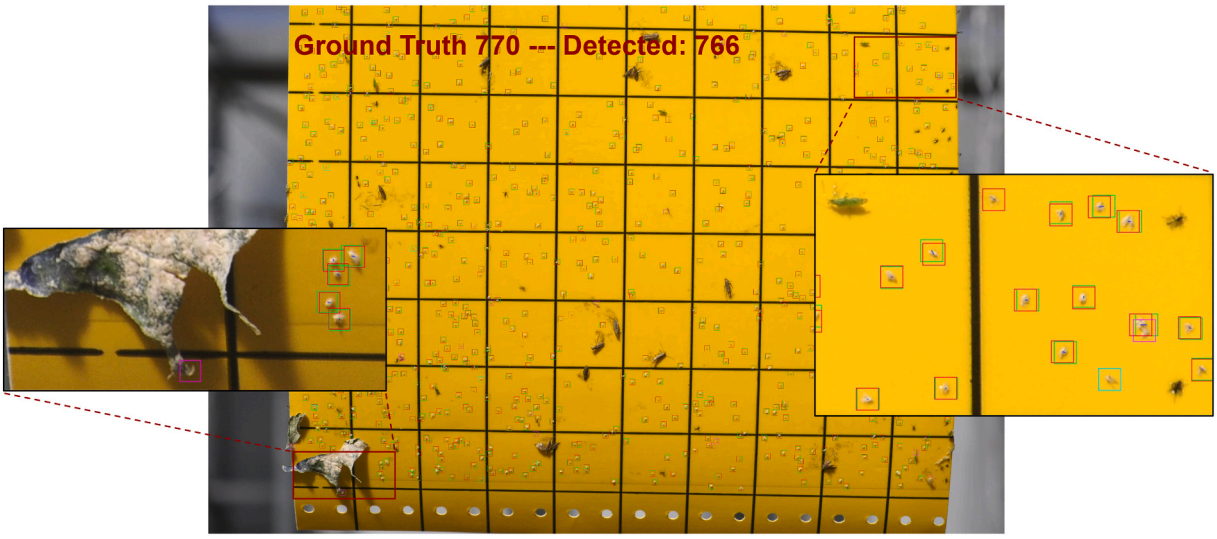


Fig. 4. Output sample of the best-performing counting module PC_θ**.** We show an output sample produced by the PC_θ module of the best-performing model, i.e., *Det-FRCNN*. We highlight the ground truth with green squares; furthermore, we indicate false positives in purple, false negatives in cyan, and true positives in red, connected via a thin blue line to the corresponding ground truth position. We also show some zoomed areas to highlight some errors due to contaminants (false positive in the left zoomed region) and very challenging to find whiteflies (false negative in the right zoomed region). Best viewed in electronic format. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

we break them down into smaller patches. Specifically, in the training stage, we randomly crop square patches (experimenting with various patch sizes, namely 256, 320, 480, 640, and 800 pixels), and we apply a simple augmentation strategy, flipping the patches horizontally and vertically with a 50% probability. On the other hand, during the validation phase, we divide the images of the validation split into regularly-spaced, overlapping patches, using the same size employed during training. We process these patches individually, reconstructing the overall output by combining the predictions from the patches, and subsequently, we calculate metrics over the complete image. For clarity, Fig. 2 provides a graphical representation of this procedure. In more detail, for the solution based on detection, we reconstruct the final global output using non-maximum suppression among the bounding boxes computed in the overlapping regions; about the solutions relying on regression to density and segmentation maps, we generate image-level maps by reassembling the patch-level maps and computing the average pixel values within the regions of overlap.

Counting Analysis. In the initial set of experiments conducted to assess counting performance, we utilize standard counting benchmarks, which include Mean Absolute Error (MAE), Mean Squared Error (MSE), and Mean Absolute Relative Error (MARE). These metrics are defined as follows:

$$\mathrm{MAE} = \frac{1}{N}\sum_{n=1}^{N}\left|c_{\mathrm{gt}}^{n} - c_{\mathrm{pred}}^{n}\right|, \tag{1}$$

$$\mathrm{MSE} = \frac{1}{N}\sum_{n=1}^{N}\left(c_{\mathrm{gt}}^{n} - c_{\mathrm{pred}}^{n}\right)^{2}, \tag{2}$$

$$\mathrm{MARE} = \frac{1}{N}\sum_{n=1}^{N}\frac{\left|c_{\mathrm{gt}}^{n} - c_{\mathrm{pred}}^{n}\right|}{c_{\mathrm{gt}}^{n}}. \tag{3}$$

where N stands for the number of images belonging to the test split, while c_{gt}^{n} and c_{pred}^{n} refer to the ground truth and the predicted numbers of insects present in the n-th image, respectively. It's important to highlight that, due to the squaring of differences, the MSE places more weight on larger errors, including outliers. Conversely, the MARE also considers how the error relates to the total object count in the image. This metric holds particular significance in our PST dataset, where the insect count exhibits significant variation across images.

Localization Analysis. While the MAE, MSE, and MARE serve as fair metrics for comparing counting performance and are considered the gold standard for this task, they can sometimes mask erroneous estimations. The reason behind this is that they do not account for *where* the

Table 3
Evaluation of the Localization Performance. We report the obtained results using the Grid Average Mean absolute Error (GAME) (Guerrero-Gómez-Olmedo et al., 2015), which considers both the counting error and the estimated positions of the objects. A higher value of L results in increased consideration of sub-regions, thereby imposing a stricter criterion for the GAME metric.

	GAME(L)↓				
Method	$L = 1$	$L = 2$	$L = 3$	$L = 4$	$L = 5$
Det-FRCNN (Ren et al., 2017)	**66.000**	**71.750**	84.500	**100.875**	133.000
Det-FCOS (Tian et al., 2019)	66.500	72.875	**84.000**	101.500	**129.750**
Den-CSRNet (Li et al., 2018)	205.125	227.625	254.125	279.875	328.375
Den-FCRN (Xie et al., 2016)	127.750	138.750	151.000	175.750	218.250
Seg-UNet (Falk et al., 2018)	90.750	101.625	118.125	145.000	202.000

estimations are made within the images, thereby failing to capture localization errors. Consequently, models may achieve low MAE, MSE, and MARE values while still providing inaccurate predictions (for instance, many false positives and negatives in detection-based techniques or misleading values in predicted density maps). Thus, relying solely on these counting metrics to select the best counting approach can lead to mistakes. To address this limitation and also consider the (coarse) localizations of the detected insects, we employ the Grid Average Mean Absolute Error (GAME) (Guerrero-Gómez-Olmedo et al., 2015), a metric that simultaneously takes into account the object count and the estimated object locations. Precisely, GAME is calculated by dividing the image into 4^L non-overlapping regions, where L represents the grid level. The metric sums the MAE computed in each sub-region, allowing for a more comprehensive assessment of the counting approach while incorporating information about the spatial distribution of the detected objects. Formally:

$$GAME(L) = \frac{1}{N}\sum_{n=1}^{N}\left(\sum_{l=1}^{4^L}|c_{gt}^{l} - c_{pred}^{l}|\right), \tag{4}$$

where the notation is the same as the one used for the MAE in Eq. 1. Notably, the MAE can be seen as a special case of the GAME when the parameter L is set to 0. Additionally, a higher value of L results in increased consideration of sub-regions, thereby imposing a stricter criterion for the GAME metric.

5.3. Results

Counting Analysis. Table 2 presents the results of all the proposed approaches concerning their optimal combination of patch size and threshold values. Among the tested solutions, the detection-based approaches demonstrate the best performance. Specifically, *Det-FRCNN* achieved the top results across all three counting metrics. Notably, the obtained MARE value of 9.3% is particularly noteworthy, indicating reliable performance across all images, whether in "crowded" scenarios or those containing only a few insects. On the other hand, the density-based solutions exhibit more significant errors, especially struggling with the MARE metric. While they perform well in "crowded" contexts, they face limitations when applied to scenarios with a small number of objects. Finally, *Seg-UNet* achieved slightly inferior performance compared to *Det-FRCNN* and *Det-FCOS*, but it outperforms the density-based approaches, particularly concerning the MARE. For a visual illustration, Fig. 4 showcases an output sample of the best-performing model, i.e., *Det-FRCNN*.

Localization Analysis. Table 3 presents the results obtained in terms of GAME, with L ranging from 1 to 5, for all the proposed approaches (again, concerning their optimal combination of patch size and threshold values). Once again, the detection-based approaches demonstrate superior performance, achieving the best results overall. Notably, *Det-FRCNN* remains the leading approach, although it is occasionally surpassed by *Det-FCOS* for specific values of L. On the other hand, *Den-CSRNet* and *Den-FCRN* exhibit lower performance in localizing the counted insects, which aligns with the inherent limitation of density-based techniques in accurately localizing objects within the region of interest. Last, the segmentation-based approach sits between these two extremes, providing intermediate performance.

Influence of patch size and threshold. In this section, we comprehensively analyze the outcomes obtained by the two top-performing approaches, namely *Det-FRCNN* and *Det-FCOS*, while changing the patch size and the threshold on the generated confidence score. Fig. 5 illustrates the findings. Firstly, we observe that the performances of these two approaches are not notably affected by the patch size. In general, smaller patch sizes lead to slightly better results, albeit with increased processing overhead. However, the difference in performance between different patch sizes is not very pronounced. Hence, we recommend opting for larger patch sizes if there are any computational constraints, as it allows for a fair compromise between execution time and result reliability. Regarding the threshold used to filter out unreliable predictions, we find that it closely depends on the evaluation metrics adopted. However, *Det-FCOS* is particularly affected by low thresholds. This is due to its architecture, which does not rely on a preliminary regions proposal stage, resulting in an abundance of bounding boxes that contain irrelevant object representations.

6. Conclusion

This paper introduced an AI-based tool to automatically count pests in pictures of sticky chromotropic traps gathered by smartphones or digital cameras. Accurately estimating pest populations is a crucial aspect of IPM strategies; however, existing procedures require significant human effort to manually inspect traps placed in suspected insect hotspots, leaving room for automated processes.

To this end, we proposed a flexible model-agnostic solution implemented in several ways, covering all counting strategies and enclosing most of the existing SOTA pest detection and counting approaches. Furthermore, regardless of the adopted counting approach, our solution can also provide a coarse localization of the found insects and a confidence score associated with the detected instances that practitioners can exploit to filter low-quality results. Additionally, for the first time, we conducted a comprehensive evaluation of counting performance, unlikely most existing works that primarily focus on measuring detection capabilities. Our assessment has been conducted using the PST - Pest Sticky Traps dataset, a collection of dot-labeled images depicting sticky traps with whiteflies, which we have made freely available as an additional contribution.

The results obtained indicate that, in this particular scenario,

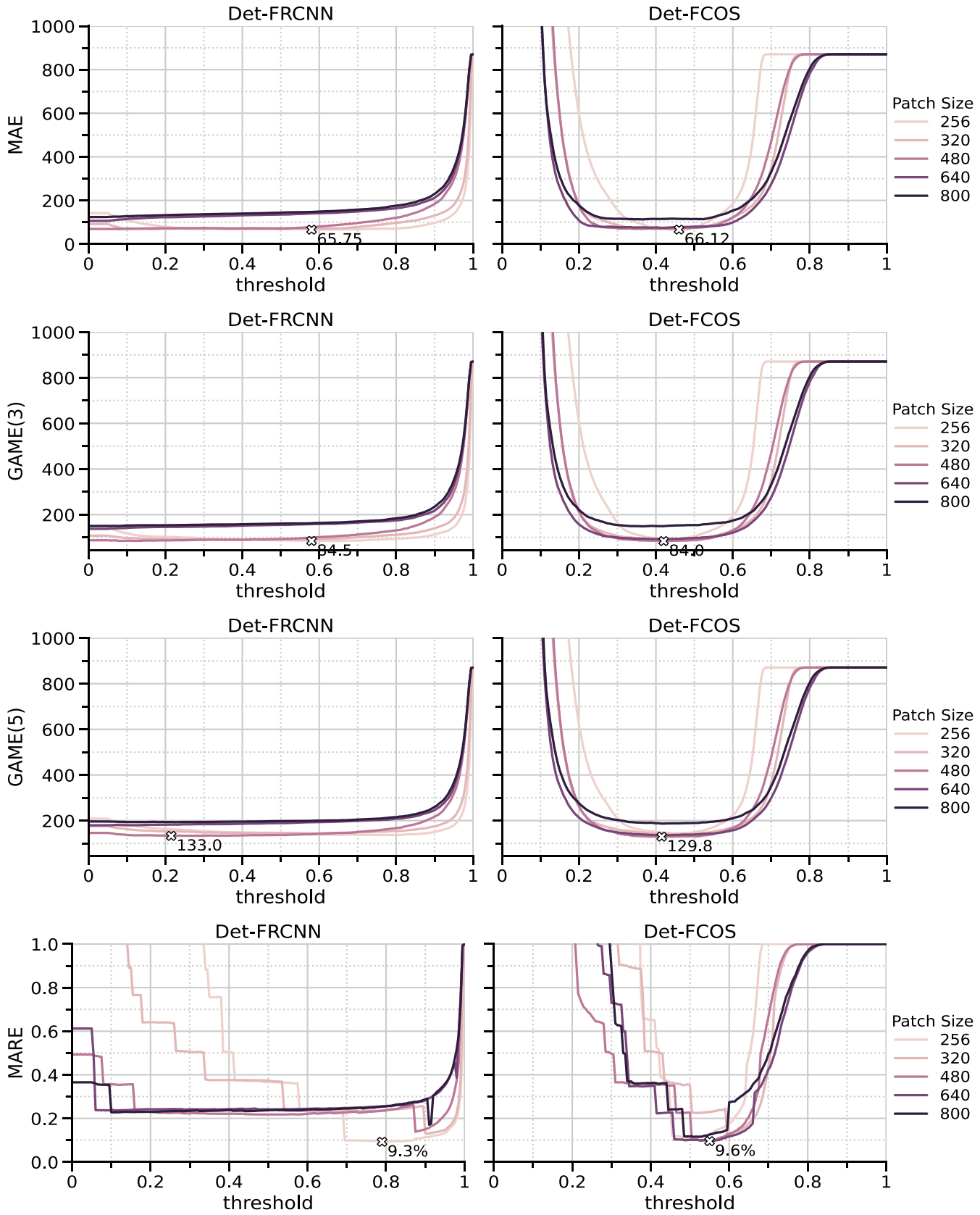


Fig. 5. Influence of patch size and threshold on best approaches. We present the outcomes concerning MAE, GAME, and MARE metrics for the top-performing methodologies, namely *Det-FRCNN* and *Det-FCOS* while changing the patch size and the threshold on the generated confidence score.

detection-based approaches yielded the most effective results, with an average counting error of approximately 9% compared to human capabilities, along with satisfactory localization precision. Notably, our solution can perform pest population estimation on sticky trap images within seconds, compared to the hours required for manual human inspections. Given that our solution is model-agnostic, i.e., it can easily incorporate subsequent models from the literature and still utilize the proposed pipeline, and that it works in the wild, i.e., any pre-processing strategy of the input images is needed, we deem it represents a valuable AI-based tool to help farm owners to control pest spread and implement effective, sustainable, IPM strategies.

Declaration of Competing Interest

None.

Data availability

Data is available in the Zenodo repository.

Acknowledgements

This work was funded by: (i) AI4Media - A European Excellence Centre for Media, Society and Democracy (EC, H2020 n. 951911); (ii) PNRR - M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 - "FAIR - Future Artificial Intelligence Research" - Spoke 1 "Human-centered AI", funded by European Union - NextGenerationEU; (iii) European PRIMA Project "Innovative Greenhouse Support System in the Mediterranean Region efficient fertigation and pest management through IoT based climate control (iGUESS-MED) (GA: 1916). The study design, data collection, analysis, decision to publish, and manuscript preparation were conducted without any involvement from the funding sources.

References

Amato, G., Ciampi, L., Falchi, F., Gennaro, C., 2019. Counting vehicles with deep learning in onboard UAV imagery. In: In 2019 IEEE Symposium on Computers and Communications (ISCC). IEEE.
Arteta, C., Lempitsky, V., Zisserman, A., 2016. Counting in the wild. In: Computer Vision – ECCV 2016. Springer International Publishing, pp. 483–498.
Barbedo, J.G.A., 2014. Using digital image processing for counting whiteflies on soybean leaves. J. Asia Pac. Entomol. 17 (4), 685–694.
Benedetto, M.D., Carrara, F., Ciampi, L., Falchi, F., Gennaro, C., Amato, G., 2022. An embedded toolset for human activity monitoring in critical environments. Expert Syst. Appl. 199, 117125.
Bereciartua-Pérez, A., Gómez, L., Picón, A., Navarra-Mestre, R., Klukas, C., Eggers, T., 2022. Insect counting through deep learning-based density maps estimation. Comput. Electron. Agric. 197, 106933.
Bereciartua-Pérez, A., Gómez, L., Picón, A., Navarra-Mestre, R., Klukas, C., Eggers, T., 2023. Multiclass insect counting through deep learning-based density maps estimation. Smart Agricult. Technol. 3, 100125.
Bhagat, S., Kokare, M., Haswani, V., Hambarde, P., Kamble, R., 2022. Eff-unet++: A novel architecture for plant leaf segmentation and counting. Ecol. Inform. 68, 101583.
Bochkovskiy, A., Wang, C., Liao, H.M., 2020. Yolov4: Optimal Speed and Accuracy of Object Detection. *CoRR*, abs/2004.10934.
Ciampi, L., Carrara, F., Totaro, V., Mazziotti, R., Lupori, L., Santiago, C., Amato, G., Pizzorusso, T., Gennaro, C., 2022a. Learning to count biological structures with raters' uncertainty. Med. Image Anal. 80, 102500.
Ciampi, L., Gennaro, C., Carrara, F., Falchi, F., Vairo, C., Amato, G., 2022b. Multi-camera vehicle counting using edge-AI. Expert Syst. Appl. 207, 117929.
Ciampi, L., Zeni, V., Incrocci, L., Canale, A., Benelli, G., Falchi, F., Amato, G., Chessa, S., 2023. Pest Sticky Traps: a dataset for Whitefly Pest Population Density Estimation in Chromotropic Sticky Traps. Zenodo.
Costa, C.S., Gonçalves, W.N., Zanoni, V.A.G., dos Santos de Arruda, M., de Araújo Carvalho, M., Nascimento, E., Marcato Junior, J., Diemer, O., Pistori, H., 2023. Counting tilapia larvae using images captured by smartphones. Smart Agricult. Technol. 4, 100160.
Dara, S.K., 2019. The New Integrated Pest Management Paradigm for the Modern Age. J. of Integrat. Pest Manag. 10 (1), 12.
Ding, W., Taylor, G., 2016. Automatic moth detection from trap images for pest management. Comput. Electron. Agric. 123, 17–28.
Falk, T., Mai, D., Bensch, R., Çiçek, Özgün, Abdulkadir, A., Marrakchi, Y., Böhm, A., Deubner, J., Jäckel, Z., Seiwald, K., Dovzhenko, A., Tietz, O., Bosco, C.D., Walsh, S., Saltukoglu, D., Tay, T.L., Prinz, M., Palme, K., Simons, M., Diester, I., Brox, T., Ronneberger, O., 2018. U-net: deep learning for cell counting, detection, and morphometry. Nat. Methods 16 (1), 67–70.
Farina, A., Barbera, A.C., Leonardi, G., Cocuzza, G.E.M., Suma, P., Rapisarda, C., 2022. *Bemisia tabaci* (hemiptera: Aleyrodidae): What relationships with and morpho-physiological effects on the plants it develops on? Insects 13 (4), 351.
Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D., 2010. Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. 32 (9), 1627–1645.
Guerrero-Gómez-Olmedo, R., Torre-Jiménez, B., López-Sastre, R., Maldonado-Bascón, S., Oñoro-Rubio, D., 2015. Extremely overlapping vehicle counting. In: Pattern Recognition and Image Analysis. Springer International Publishing, pp. 423–431.
He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.
Hsieh, M.-R., Lin, Y.-L., Hsu, W.H., 2017. Drone-based object counting by spatially regularized regional proposal network. In: In 2017 IEEE International Conference on Computer Vision (ICCV). IEEE.
Jocher, G., Chaurasia, A., Qiu, J., 2023. YOLO by Ultralytics.
Kamilaris, A., Gao, F., Prenafeta-Boldu, F.X., Ali, M.I., 2016. Agri-IoT: A semantic framework for internet of things-enabled smart farming applications. In: In 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT). IEEE.
Kasinathan, T., Singaraju, D., Uyyala, S.R., 2021. Insect classification and detection in field crops using modern machine learning techniques. Inform. Proc. Agricult. 8 (3), 446–457.
Khalid, S., Oqaibi, H.M., Aqib, M., Hafeez, Y., 2023. Small pests detection in field crops using deep learning object detection. Sustainability 15 (8).
Lamichhane, J.R., Aubertot, J.-N., Begg, G., Birch, A.N.E., Boonekamp, P., Dachbrodt-Saaydeh, S., Hansen, J.G., Hovmøller, M.S., Jensen, J.E., Jørgensen, L.N., Kiss, J., Kudsk, P., Moonen, A.-C., Rasplus, J.-Y., Sattin, M., Streito, J.-C., Messéan, A., 2016. Networking of integrated pest management: A powerful approach to address common challenges in agriculture. Crop Prot. 89, 139–151.
Lempitsky, V.S., Zisserman, A., 2010. Learning to count objects in images. In: Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada. Curran Associates, Inc, pp. 1324–1332.
Li, Y., Zhang, X., Chen, D., 2018. CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes. In: In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE.
Li, W., Chen, P., Wang, B., Xie, C., 2019. Automatic localization and count of agricultural crop pests based on an improved deep learning pipeline. Sci. Rep. 9 (1).
Li, W., Wang, D., Li, M., Gao, Y., Wu, J., Yang, X., 2021a. Field detection of tiny pests from sticky trap images using deep learning in agricultural greenhouse. Comput. Electron. Agric. 183, 106048.
Li, W., Zheng, T., Yang, Z., Li, M., Sun, C., Yang, X., 2021b. Classification and detection of insects from field images using deep learning for smart pest management: A systematic review. Ecol. Inform. 66, 101460.
Lima, M.C.F., de Almeida Leandro, M.E.D., Valero, C., Coronel, L.C.P., Bazzo, C.O.G., 2020. Automatic detection and monitoring of insect pests—a review. Agriculture 10 (5), 161.
Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollar, P., 2017. Focal loss for dense object detection. In: In 2017 IEEE International Conference on Computer Vision (ICCV). IEEE.
Lins, E.A., Rodriguez, J.P.M., Scoloski, S.I., Pivato, J., Lima, M.B., Fernandes, J.M.C., da Silva Pereira, P.R.V., Lau, D., Rieder, R., 2020. A method for counting and classifying aphids using computer vision. Comput. Electron. Agric. 169, 105200.
Liu, W., Salzmann, M., Fua, P., 2019. Context-aware crowd counting. In: In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.
Mahmud, M.S., He, L., Heinemann, P., Choi, D., Zhu, H., 2023. Unmanned aerial vehicle based tree canopy characteristics measurement for precision spray applications. Smart Agricult. Technol. 4, 100153.
Moazzam, S.I., Khan, U.S., Qureshi, W.S., Nawaz, T., Kunwar, F., 2023. Towards automated weed detection through two-stage semantic segmentation of tobacco and weed pixels in aerial imagery. Smart Agricult. Technol. 4, 100142.
Nieuwenhuizen, A.T., Hemming, J., Suh, H.K., 2018. Detection and Classification of Insects on Stick-Traps in a Tomato Crop using Faster r-cnn.
Partel, V., Nunes, L., Stansly, P., Ampatzidis, Y., 2019. Automated vision-based system for monitoring asian citrus psyllid in orchards utilizing artificial intelligence. Comput. Electron. Agric. 162, 328–336.
Putra, Y.C., Wijayanto, A.W., Chulafak, G.A., 2022. Oil palm trees detection and counting on microsoft bing maps very high resolution (vhr) satellite imagery and unmanned aerial vehicles (uav) data using image processing thresholding approach. Ecol. Inform. 72, 101878.
Ramasamy, S., Ravishankar, M., 2018. Integrated pest management strategies for tomato under protected structures. In: Sustainable Management of Arthropod Pests of Tomato. Elsevier, pp. 313–322.
Redmon, J., Farhadi, A., 2018. Yolov3: An Incremental Improvement. *CoRR*, abs/1804.02767.
Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.
Ren, S., He, K., Girshick, R., Sun, J., 2017. Faster r-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 39 (6), 1137–1149.
Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich,

Germany, October 5 - 9, 2015, Proceedings, Part III, volume 9351 of Lecture Notes in Computer Science. Springer, pp. 234–241.
Rustia, D.J.A., Chao, J.-J., Chiu, L.-Y., Wu, Y.-F., Chung, J.-Y., Hsu, J.-C., Lin, T.-T., 2020. Automatic greenhouse insect pest detection and recognition based on a cascaded deep learning classification method. J. Appl. Entomol. 145 (3), 206–222.
Sekine, T., Takanashi, T., Onodera, R., Oe, T., Komagata, Y., Abe, S., Koike, T., 2022. Potential of substrate-borne vibration to control greenhouse whitefly trialeurodes vaporariorum and increase pollination efficiencies in tomato solanum lycopersicum. J. Pest. Sci. 96 (2), 599–610.
She, J., Zhan, W., Hong, S., Min, C., Dong, T., Huang, H., He, Z., 2022. A method for automatic real-time detection and counting of fruit fly pests in orchards by trap bottles via convolutional neural network with attention mechanism added. Ecol. Inform. 70, 101690.
Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015. Proceedings, Conference Track.
Sun, Y., Cheng, H., Cheng, Q., Zhou, H., Li, M., Fan, Y., Shan, G., Damerow, L., Lammers, P.S., Jones, S.B., 2017. A smart-vision algorithm for counting whiteflies and thrips on sticky traps using two-dimensional fourier transform spectrum. Biosyst. Eng. 153, 82–88.
Sun, Y., Liu, X., Yuan, M., Ren, L., Wang, J., Chen, Z., 2018. Automatic in-trap pest detection using deep learning for pheromone-based *Dendroctonus valens* monitoring. Biosyst. Eng. 176, 140–150.
Tan, X.L., Chen, J.L., Benelli, G., Desneux, N., Yang, X.Q., Liu, T.X., Ge, F., 2017. Pre-infestation of tomato plants by aphids modulates transmission-acquisition relationship among whiteflies, tomato yellow leaf curl virus (TYLCV) and plants. *Frontiers*. Plant Sci. 8.
Tian, Z., Shen, C., Chen, H., He, T., 2019. FCOS: Fully convolutional one-stage object detection. In: In 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE.
Tian, H., Wang, T., Liu, Y., Qiao, X., Li, Y., 2020. Computer vision technology in agricultural automation —a review. Inform. Proc. Agricult. 7 (1), 1–19.
Tshernyshev, W.B., 1995. Ecological pest management (epm): general approaches. J. Appl. Entomol. 119 (1–5), 379–381.
Wang, Q.-J., Zhang, S.-Y., Dong, S.-F., Zhang, G.-C., Yang, J., Li, R., Wang, H.-Q., 2020. Pest24: A large-scale very small object data set of agricultural pests for multi-target detection. Comput. Electron. Agric. 175, 105585.
Wang, D., Wang, Y., Li, M., Yang, X., Wu, J., Li, W., 2021. Using an improved YOLOv4 deep learning network for accurate detection of whitefly and thrips on sticky trap images. Trans. ASABE 64 (3), 919–927.
Wang, T., Zhao, L., Li, B., Liu, X., Xu, W., Li, J., 2022. Recognition and counting of typical apple pests based on deep learning. Ecol. Inform. 68, 101556.
Xie, W., Noble, J.A., Zisserman, A., 2016. Microscopy cell counting and detection with fully convolutional regression networks. Comput. Methods Biomech. Biomed. Eng. Imaging Vis. 6 (3), 283–292.
Yu, F., Koltun, V., 2016. Multi-scale context aggregation by dilated convolutions. In: In 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings.
Zhong, Y., Gao, J., Lei, Q., Zhou, Y., 2018. A vision-based counting and recognition system for flying insects in intelligent agriculture. Sensors 18 (5), 1489.