Automated Insect Counting on Yellow Sticky Traps Using Deep Learning: A YOLOv8-Based Approach for Smart Pest Control

Author Names:
Mr. Christian Salz, M.Sc -Ing., Mr. Tim Riebner, M.Sc-Ing.

Email:
c.salz@fontys.nl, tim.riebner@procevolution.com

Received xxxxxx
Accepted for publication xxxxxx
Published xxxxxx

Abstract

Effective pest control is crucial for maintaining ecological balance within greenhouses and open-field cultivation, in which insect populations must be closely monitored and managed. Traditional monitoring methods rely on the manual inspection of yellow sticky traps — a labor-intensive and error-prone process requiring significant expertise and time. In this study, we present a novel approach to automated insect counting using a custom-built deep learning model based on the YOLOv8 architecture. We developed a specialized dataset of 227 insect images collected from yellow sticky traps, of which 104 were annotated with a single class named 'insect'. The model achieved an average precision of 56,16%, with the mAP50 stabilizing at around 21,88% toward the end of training. Further evaluation using the validation set demonstrated an average box precision of 79.8%, which is a strong starting result given the dataset size and complexity. A field study with 19 participants showed that manual insect counting took an average of 1 minute and 46 seconds per image, while the AI model completed the task in just 196.5 milliseconds, reducing the required time by over 99.81%. Additionally, 11 out of 19 participants preferred the AI-generated results over their own, and all participants indicated they would prefer an AI system if they had to perform insect counting for a full day. The total training time for 92 cycles was 1,964 hours, including 187 augmented images for training and 37 images for validation. The YOLOv8-m model was trained with an image size parameter of imgsz=1024. This automated system enables more frequent and accurate monitoring, contributing to the sustainable pest management ecosystems. Future research should focus on refining model performance, introducing a secondary model for insect classification, and integrating environmental data to improve population dynamics prediction.

Keywords: Automated insect counting, YOLOv8 architecture, dataset annotation and preparation, Computer vision in agriculture

1. Introduction

Greenhouses play a vital role in modern agriculture by providing controlled environments that optimize plant growth and productivity. However, maintaining ecological balance within greenhouses is essential to prevent pest outbreaks that could threaten crop yields and quality. One of the most widely used methods for pest monitoring involves the deployment of yellow sticky traps which attract and capture flying insects. These traps provide valuable insight into insect population dynamics, but the process of analyzing them remains largely manual and labor-intensive.

In most greenhouse and open-field operations, hundreds of yellow sticky traps are deployed throughout the area. Each trap must be visually inspected, and the number and type of insects captured must be carefully counted and recorded. This traditional approach is not only time-consuming but also highly prone to human error, especially when insect densities are high or when the traps accumulate dirt and debris over time. Figure 1 illustrates the challenge posed by these conditions, showing a typical yellow sticky trap with numerous captured insects like the *Drosophila suzukii, Tuta absoluta, Trialeurodes vaporariorum* and the *Musca domestica*.

Given the scale of both greenhouse and open-field cultivation, as well as the need for daily monitoring, the demand for an automated, reliable, and efficient insect counting system has become increasingly apparent. Automation in this context would significantly reduce the workload on human operators, minimize counting errors, and enable more frequent and consistent data collection. Accurate insect monitoring is crucial for timely pest control measures and for maintaining the delicate balance of greenhouse ecosystems.

Figure 1: Example yellow sticky trap

In this study, we propose an automated approach to insect counting using a deep learning model based on the YOLOv8 architecture. By leveraging a specialized dataset of insect images collected from yellow sticky traps, our model is trained to identify and localize individual insects on these traps. This approach provides a scalable and efficient alternative to traditional manual counting methods. The remainder of this paper details the dataset preparation, model development, training process, and evaluation results, demonstrating the potential of this automated system for practical deployment in greenhouse pest management. [1] [2]

1.1 Current approaches

Currently, insect counting is primarily done manually by biologists, which is time-consuming and prone to human error. There are no ready-to-use automated solutions on the market. Some companies offer isolated box solutions and robotic systems, but these have significant drawbacks. Fixed box solutions are expensive, require space and maintenance, and are not practical as insects fly at different heights depending on environmental conditions. Robotic systems, while innovative, are limited to ground-level measurements and often depend on network connectivity, which can be unreliable. Existing AI-based insect detection systems are typically trained only on specific insect species, limiting their ability to generalize to diverse real-world conditions. The yellow sticky traps used for training in these systems often do not reflect actual agricultural usage, as they contain minimal noise (such as dirt, leaves, or water) and generally have low insect densities per card. Such controlled environment reduces the complexity of detection tasks, making these models less effective when applied in real-world farming scenarios.

Our approach addresses these limitations by providing a simple, cost-effective solution that requires only a device to capture images of yellow sticky traps. This method is easy to use, even for non-technical farmers, and integrates seamlessly into existing processes without significant disruption. Unlike existing systems, our model is designed to generalize across different insect species rather than being limited to specific trained classes. Additionally, it can handle real-world challenges such as dirt, leaves, and other noise on the sticky traps. The system may also be capable of detecting previously unseen insect species, making it a more robust and adaptable solution for agricultural pest monitoring. [3] [4]

2. Dataset

The dataset used in this study was collected over the course of 2024 to monitor insect populations in both greenhouse and open-field cultivation environments. Photographs of yellow sticky traps were taken on a weekly basis using an iPhone 12 Pro equipped with a 12 MP camera, capturing images in a 3:4 aspect ratio. To ensure variety, all images were captured against a gray background with varying lighting conditions, angles, and distances. This approach was chosen to reflect real-world farming environments, where conditions are rarely perfect, and high-resolution imaging is not a practical use case.

In total, 227 images of yellow sticky traps were collected. Out of these, 104 images were annotated using the Roboflow platform. Annotation focused exclusively on a single class labeled "insect." Only fully intact insects—those with clearly visible wings, legs, and body structures—were included in the annotations. This decision was made to avoid ambiguities caused by damaged or partially obscured specimens, which could otherwise compromise the accuracy of model training. An example of an annotated image can be seen in Figure 2.

The dataset was split into training and validation sets using a 70/30 ratio. This division ensures that the model is exposed to a broad range of visual data while retaining sufficient unseen examples for evaluation purposes. To address the limited size of the dataset, image augmentation techniques were applied. These included 90-degree rotations in all dimensions and mirroring to increase data diversity and improve model generalization. Since the YOLO architecture requires a 1:1 aspect ratio for training, black padding was added to the images to avoid cropping while maintaining the integrity of the original content. These augmentation steps ensured the model was exposed to varied perspectives and orientations of the insect data, strengthening its detection capabilities.

The quality and consistency of the dataset form a solid foundation for developing and validating an automated insect counting system tailored to greenhouse pest monitoring needs.

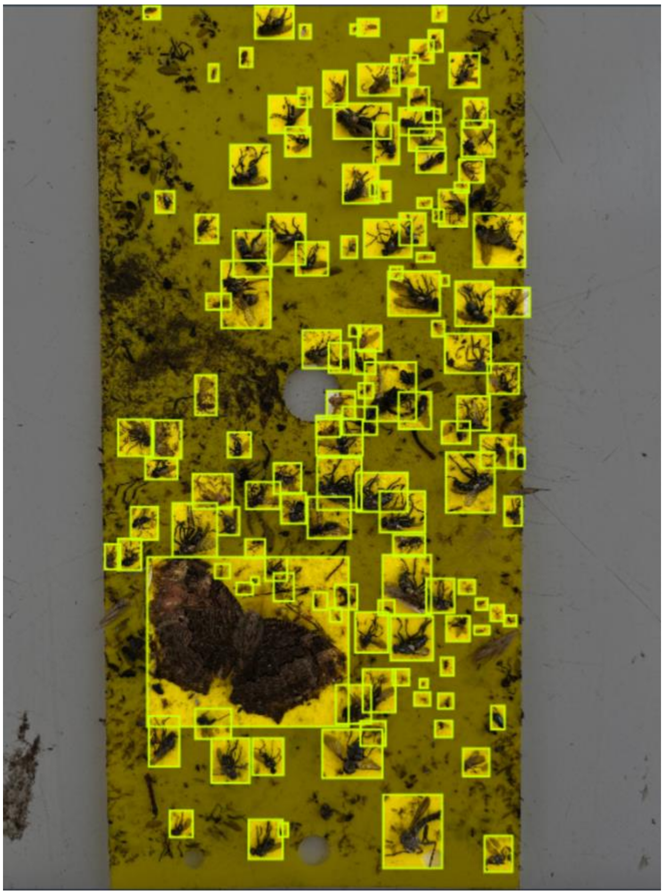

Figure 2: Annotated image

The dataset used in this study has been made publicly available for others to explore. Interested individuals can access and review the annotated images through the following link on: [https://app.roboflow.com/annotationswithrf/spohf-kur4x]. This allows researchers and practitioners to check the data and potentially use it for further research or applications. [5]

3. Model Architecture and Training

3.1 YOLOv8 Architecture

The YOLOv8 (You Only Look Once) architecture is a state-of-the-art object detection model known for its balance between speed and accuracy. YOLOv8 offers several network size options, ranging from the lightweight nano model (n) to the extra-large model (xl), allowing flexibility based on hardware capabilities and application requirements. For this study, the medium (m) model was selected as it provided an optimal balance of performance and feasibility given the hardware constraints.

3.2 Hardware and Software Environment

Model training was conducted on an M4 Max MacBook Pro equipped with 64 GB of RAM. The training environment utilized PyTorch with Metal acceleration, taking advantage of the device's GPU capabilities. Despite the hardware's advanced specifications, only up to the YOLOv8-m model could be trained with an appropriate batch size, due to the memory demands of larger models.

3.3 Pretrained Network and Transfer Learning

To enhance training efficiency and accuracy, the YOLOv8-m model was initialized using weights pretrained on the COCO dataset. The COCO (Common Objects in Context) dataset is a large-scale object detection dataset widely used for training deep learning models. By leveraging pretrained weights, the model benefits from feature representations learned on a diverse set of objects, accelerating convergence and improving performance on the insect detection task. When using an untrained YOLO model, it was observed that at least 32 epochs were required for the model to reach a comparable performance level to the version initialized with COCO weights, demonstrating the efficiency gained through transfer learning.

3.4 Data Preparation and Augmentation

Before training, the annotated insect dataset underwent preprocessing to align with YOLOv8's requirements. As described in the dataset section, images were resized to a 1:1 aspect ratio using black padding to prevent any loss of visual information. Image augmentation techniques, including 90-degree rotations and mirroring, were applied to increase dataset diversity and improve model generalization. The training set comprised 70% of the annotated images, while the remaining 30% were reserved for validation.

3.5 Training Experiments and Evaluation Metrics

For the training, several experiments were conducted, ranging from batch sizes of 1 up to 60. It was observed that changing the batch size did not improve the accuracy of the model but significantly affected the training time. In the end, the default batch size allocation by YOLO, based on the available hardware, was chosen for optimal performance. Early stopping was tested with the patience parameter set above 100 epochs, but it did not significantly impact the model's precision. To ensure the image aspect ratio remained unaffected during training, the *rect=True* parameter was applied. For model evaluation, the metrics used were mAP50 (mean Average Precision at 50% IoU), mAP95 (mean Average Precision at 95% IoU), and box precision. These metrics provided insights into the model's detection performance, balancing precision and recall across varying levels of intersection over union (IoU) thresholds.

3.6 Training Performance and Precision Stabilization

During the training process, the model's precision over a single training cycle of 92 epochs was monitored and visualized using TensorBoard. As shown in Figure 3, the model's precision stabilizes at 56,16% after an initial phase of fluctuation and rapid improvement. This behavior highlights the necessity of extended training cycles to allow the model to generalize effectively from the dataset.

In addition to the precision metric, the mAP50 (Mean Average Precision at 50% IoU) stabilizes at around 25% toward the end of training. While this may initially appear suboptimal, further evaluation using the validation set demonstrated an average box precision of 79.8%, which is a strong starting result for this model given the dataset size and complexity. The total training time for 92 cycles was 1,964 hours, including 187 augmented images for training and 37 images for validation. Regarding image size, the YOLOv8-m model was trained with an image size parameter of *imgsz=1024*. Attempts to exceed 1200 pixels frequently resulted in memory leaks and system crashes, while reducing the resolution significantly impacted model precision. This image size above presented an optimal balance between computational feasibility and model performance.

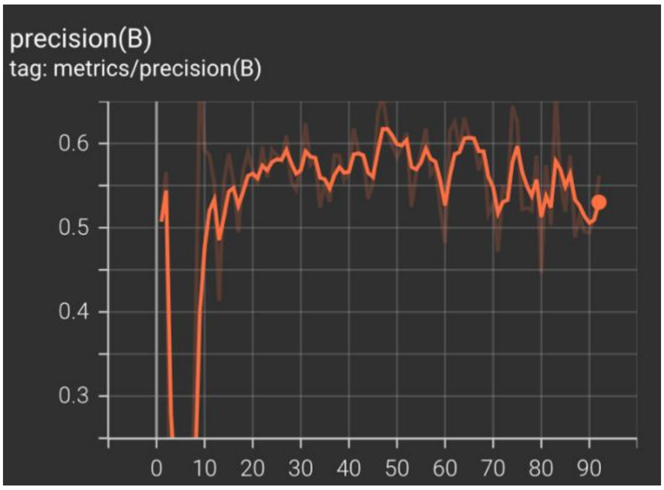

Figure 3: Precision graph

3.7 Post-Training Analysis

While the initial performance of the model, as indicated by the first few training epochs, did not seem very promising, the results were analyzed more thoroughly by manually testing the model on an additional set of 30 independent test images. These images were processed using the prediction command *results = model.predict(image, conf=0.65, iou=0.2)* to evaluate the model's performance on new data. Despite the model's relatively low precision score of 56,16%, it performed surprisingly well in detecting insects on the yellow sticky traps.

This outcome can be attributed to challenges during the annotation process. In some instances, the human labeler found it difficult to clearly identify certain objects and insects on the sticky traps, leading to ambiguous annotations. These ambiguities, where objects could not be definitively classified as insects, likely introduced confusion for the model during training. As a result, while the model's precision was affected, it still demonstrated a strong ability to detect insects under real-world conditions.

Figure 4 shows the result of the YOLO model, highlighting success in insect detection. The prediction time for the parameters above was exceptionally fast, with the following processing times per image: Speed = 4.7ms preprocess, 193.1ms inference, and 0.6ms postprocess, with an image shape of (1, 3, 768, 1024). These times are extremely fast and reproducible, especially when compared to human evaluation speeds.

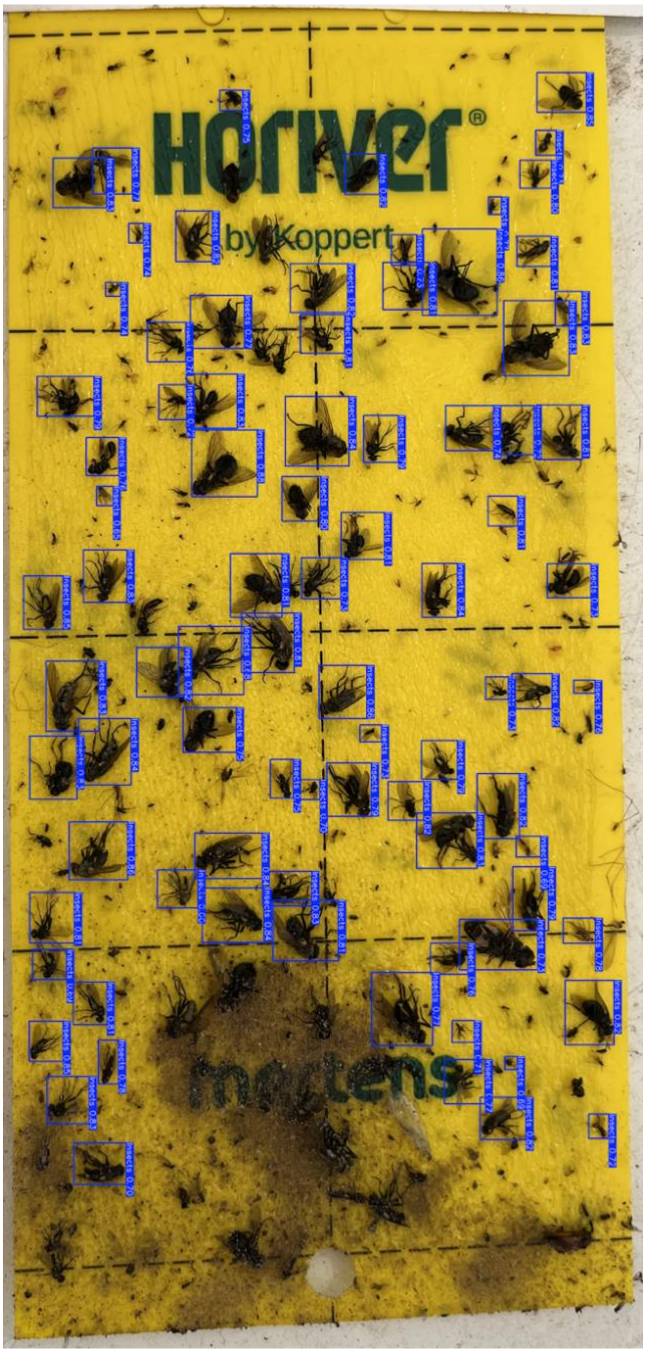

Figure 4: Evaluation of result

Further analysis revealed that the model struggled to identify insects in less sharp regions of the image, as seen at the bottom of the figure. When the confidence threshold was lowered from 65% to 35%, more insects were detected, but this also resulted in an increase in noise. Additionally, the model occasionally had difficulties distinguishing between leaves on the yellow sticky card and the insects, highlighting an area where further refinement could improve its performance.

3.8 Evaluation of Prediction Times

In this section, we evaluate the time efficiency of our deep learning model in predicting insect counts on yellow sticky traps. The prediction time is a critical factor in determining the practicality of deploying such models in real-world agricultural settings, where rapid and accurate pest monitoring is essential.

The evaluation was conducted using a unseen dataset comprising 20 samples, each representing the time taken (in milliseconds) to predict the insect count on a yellow sticky trap. The insect counts and conditions of the yellow crads varied across the samples, providing a comprehensive assessment of the model's performance under different conditions.

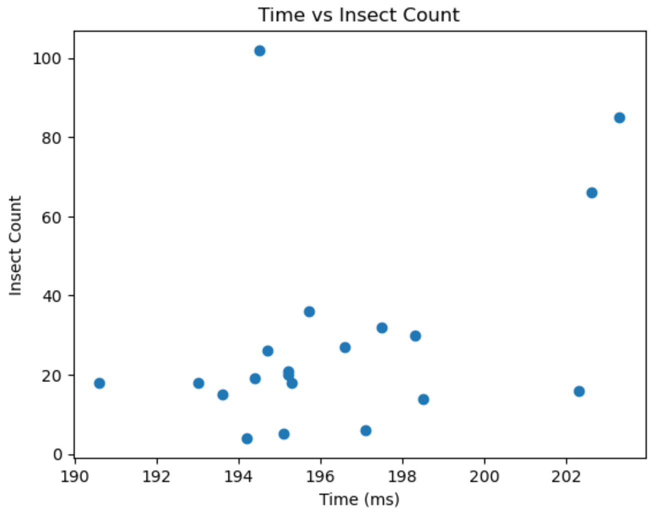

Figure 5: Insect count, prediction time

The scatter plot in Figure 5 illustrates the relationship between the prediction time and the insect count. The x-axis represents the time in milliseconds, while the y-axis represents the insect count. The data points show a relatively consistent prediction time across varying insect counts, with minor fluctuations. The average prediction time was approximately 196.5 milliseconds, demonstrating the model's efficiency in processing images and predicting outcomes.

The results indicate that the model maintains a stable prediction time regardless of the insect count, which is crucial for large-scale deployment. The consistency in prediction times ensures that the system can handle high volumes of data without significant delays, making it suitable for real-time pest monitoring in greenhouse environments.

The evaluation of prediction times highlights the model's capability to provide rapid and reliable insect counts. This efficiency, combined with the model's accuracy, supports its potential for practical application in automated pest control systems. [6] [7] [8]

3.9 Bounding Box Distribution

The distribution of the bounding boxes, which are rectangular boxes used to localize and identify objects within an image, as visualized in the heatmap (figure 6), shows a clear concentration of most boxes toward the center of the images, with some spread toward the left and right edges. This pattern arises due to the variation in the training dataset, where images were taken in both horizontal and vertical orientations. The annotation process preserved this orientation diversity, resulting in the observed distribution.

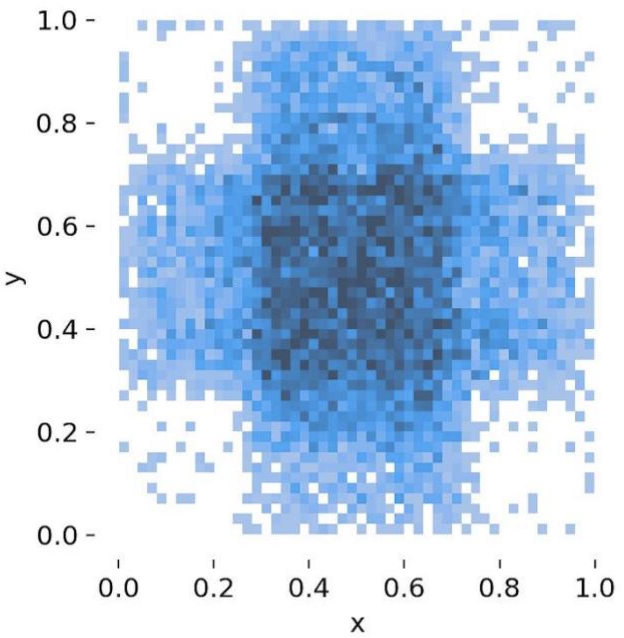

Figure 6: Distribution of boundinboxes

3.10 Bounding box size

The bounding box sizes in the training data are represented as a proportion of the total image dimensions. Figure 7 shows that most bounding boxes are concentrated in a narrow size range. Specifically, the width of the bounding boxes mostly falls between 0.01 and 0.075 percent of the total image width, and the height is similarly concentrated between 0.01 and 0.075 of the total image height. Only a few outliers exceed these dimensions. This indicates that most detected insects occupy a relatively small and consistent area within the images.

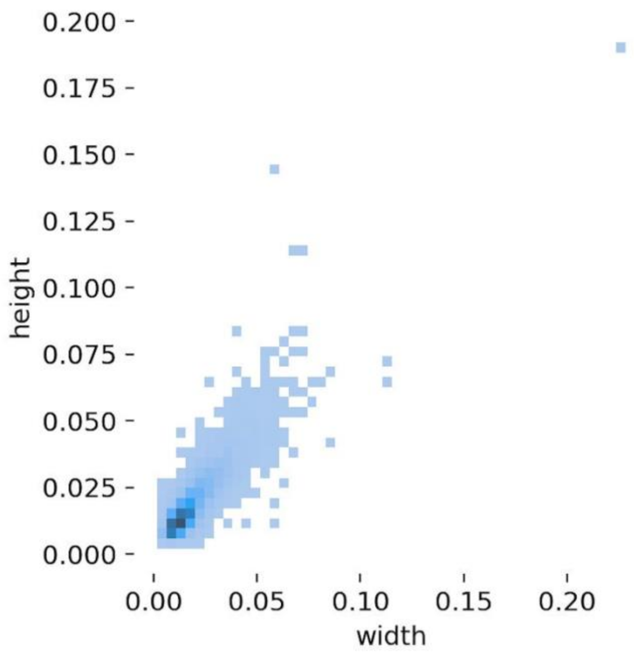

Figure 7: Relative bounding box size

3.11 Field study results: AI & Human Count

A total of 19 participants were asked to manually count insects on 19 different yellow sticky traps. The time required for each participant to complete the count was recorded, along with their preferred result (either their own count or the AI-calculated count). Additionally, participants were asked whether they would prefer to conduct insect counting manually for a full day and to rate the difficulty of the task on a scale from 1 (easy) to 3 (difficult).

The results show that the average counting time per yellow card was 1 minute and 46 seconds, with variations depending on the number of insects present. In contrast, the AI system completed the same task in an average of 196.5 milliseconds. This results in a time savings of approximately 99.81% compared to manual counting. The AI model's count differed from the human count by an average of 36 insects per image. Despite this discrepancy, 11 out of 19 participants preferred the AI-generated count over their own.

When asked if they would be willing to count insects manually for a full day, all participants indicated a preference for AI-assisted counting.

The results of this study can be seen in Figure 8, which presents the collected data and findings.

Picture (ID)	Human count	AI count	Difference	Time in MM:SS (Human)	Resul preference	Full day	Difficulty (1-3)
1	257	212	45	03:15	AI	AI	1,5
2	270	208	62	01:38	AI	AI	1
3	134	113	21	03:05	AI	AI	1
4	113	144	31	02:07	Human	AI	1
5	135	174	39	01:20	AI	AI	1
6	52	34	18	01:32	Human	AI	1,5
7	46	49	3	00:58	AI	AI	1
8	190	254	64	00:29	AI	AI	2
9	18	24	6	00:58	Human	AI	1
10	110	128	18	02:22	AI	AI	1
11	223	223	0	01:41	Human	AI	1
12	90	67	23	03:51	AI	AI	2
13	214	252	38	02:11	AI	AI	2,5
14	67	118	51	01:12	Human	AI	1
15	76	108	32	02:24	Human	AI	2
16	92	128	36	01:52	Human	AI	2
17	59	67	8	00:52	Human	AI	1,5
18	288	117	171	00:44	AI	AI	2
19	125	101	24	01:08	AI	AI	1
Average:			**36,32**	**01:46**			

Figure 8: Field study result

4. Discussion

4.1 Summary of Key Results

The results of the study indicate a significant breakthrough in the speed and efficiency of insect detection using AI models. Compared to human evaluation, the model consistently outperforms in terms of detection time, offering an exceptionally fast analysis of yellow sticky traps. The model's performance on unseen data was also notably successful, demonstrating the robustness and generalizability of the trained YOLOv8 model, even with a relatively small dataset. Despite a lower precision score of 56,16%, the model's ability to quickly detect insects remains impressive, especially when compared to traditional human evaluation times. These findings highlight the feasibility of using AI-driven models for monitoring of pest populations in agricultural settings.

4.2 Comparison with Existing Solutions

Current literature on insect detection using AI is limited, particularly when it comes to detecting insects on yellow sticky traps. Most existing models, including YOLO-based models, are designed for detecting isolated insects in single images, often under controlled conditions. There is a significant gap in research regarding the practical use of AI for insect detection in more complex environments, such as yellow card traps, which often feature overlapping objects and varying insect positions.

The model developed in this study represents the first dataset for insect detection specifically on yellow sticky traps, creating a new opportunity for real-world applications in agriculture. The current body of research is mostly focused on isolated detection, which limits its applicability in dynamic environments like greenhouses and open-field cultivation. Additionally, many existing solutions, such as robotic box systems, have proven ineffective. These systems, which rely on fixed boxes that capture images of insects, do not account for the fact that insects tend to fly higher or lower in greenhouses based on environmental factors like lighting, temperature, and humidity. These factors make fixed-position systems unsuitable for productive use, as insects are not consistently located in the same area, leading to missed detections.

In contrast, the solution we present using a camera to capture images of yellow sticky traps is more flexible and cost-effective. This approach, which takes into account the biological and environmental factors, is better than fixed box systems or robots. It provides a scalable and adaptable option for monitoring pests in agricultural environments. [9] []

4.3 Implications of Results

The findings of this study carry significant implications for agricultural pest management. Real-time, AI-driven insect detection on yellow sticky traps could enable farmers to monitor pest populations more effectively, reducing reliance on labor-intensive manual counting. The speed and accuracy of the YOLOv8 model offer a promising alternative to traditional human evaluation, making it feasible to deploy in large-scale agriculture environments for continuous pest monitoring.

Moreover, the development of a dedicated model for yellow sticky traps fills a crucial gap in the current literature, offering a foundation for future studies and real-world applications in agriculture. With further improvements in precision and the potential to expand the dataset, this model could become an essential tool for pest management, offering insights into insect population dynamics and helping to optimize pesticide and predator use.

4.4 Challenges Encountered

While the model demonstrated strong performance, several challenges were encountered during both the data annotation and the model training phases. One of the main obstacles was the ambiguity in insect identification during the annotation process. The labelers often faced difficulties distinguishing insects from other objects on the sticky traps, which led to some misclassifications in the training dataset. These ambiguities, although relatively minor, may have contributed to the model's lower precision score.

Another challenge was the size and diversity of the dataset. With only 227 images available, the model had limited exposure to variations in insect types, lighting conditions, and environmental factors. While data augmentation techniques helped mitigate this, expanding the dataset to include more diverse conditions could improve the model's performance in real-world scenarios.

Additionally, while the M4 Max MacBook Pro provided sufficient hardware for training, more powerful hardware could potentially lead to better results. A stronger system would allow for the use of the YOLOv8-xl version and enable higher image sizes, which would be beneficial for precision. There was a direct correlation observed between precision and image size during training, but hardware limitations restricted the ability to increase the image size beyond 1024 pixels.

4.5 Future Work

Future work could focus on several areas to further improve the model's performance. First, expanding the dataset with more annotated images under varying environmental conditions, such as different lighting, temperatures, and humidity levels, would allow the model to better generalize to real-world scenarios. Additionally, refining the annotation process to reduce ambiguity and improve consistency would likely enhance the model's precision.

Another area for future development is optimizing the model's ability to distinguish insects from other elements on the yellow sticky traps, such as leaves and background noise. This could involve integrating additional data sources or using advanced techniques like multi-class object detection or segmentation. Furthermore, exploring the use of synthetic data generated under various environmental conditions could increase dataset diversity without the need for extensive manual data collection.

As part of future work, the use of synthetically generated data should be investigated to further augment the dataset and improve model robustness. Counting insects is only the first step in the project. A next step will be to develop a second traditional CNN model that takes the coordinates from the YOLO predictions, extracts the detected insects, and processes them for classification. This model will then perform species identification using the softmax activation function.

5. Application

The model developed in this study offers a flexible and scalable solution for pest monitoring in agricultural environments. It can be easily deployed in a web-based system, allowing farmers and biologists to access insect detection results in real-time. Through a simple login interface, users can upload a picture of the yellow sticky traps, and within a matter of seconds, receive the detection results. This streamlined process ensures that farmers can promptly assess pest populations and make timely decisions on pest management strategies.

In addition to web-based deployment, the model can also be integrated into a mobile application. In this case, a biologist or field worker could capture a picture using their smartphone, and the model would process the image on the phone and display the results. This offers a convenient, on-the-go solution for pest monitoring, especially in large agricultural settings where field access is critical.

Currently, a first prototype of the web-based system is hosted on Microsoft Azure, providing a reliable and scalable cloud infrastructure. The system performs predictions quickly, with a cost of approximately 0.01 Euro per prediction, making it an affordable option for ongoing use. The cloud deployment also offers the advantage of being able to handle large volumes of data and predictions without significant infrastructure overhead for the user.

Alternatively, the model can also be deployed locally on a dedicated server, eliminating most costs aside from electricity and hardware and software support. Initial tests were successfully conducted on a Mac Mini equipped with an M4-Pro chip and 48 GB of RAM, demonstrating the feasibility of local deployment. This flexibility makes the system highly adaptable to various user needs, from small-scale farms to larger operations. By offering cloud-based, phone-based, and on-premise deployment options, the model ensures that farmers and biologists have access to accurate insect detection with minimal financial or technical barriers.

6. Summary and outlook

This study successfully developed an AI-based model for detecting insects on yellow sticky traps using YOLOv8. Despite challenges in data annotation and a limited dataset, the model performed well, surpassing human evaluation times and providing quick, accurate results an advantage for agricultural pest monitoring. The system's deployment is flexible, with a prototype hosted on Microsoft Azure offering real-time predictions at a low cost, and local deployment options available.

Future improvements might include expanding the dataset, refining annotations, and enhancing the model's ability to differentiate insects from background noise. Additionally, a second CNN model for insect classification based on YOLO predictions should be developed.

7. Funding and Acknowledgement

This research was conducted at Fontys University of Applied Sciences in Venlo & ProcEvolution in Krefeld as part of the Sustainable Production of Healthy Food (SPoHF) project. SPoHF focuses on advancing sustainable fruit and vegetable cultivation through digital technologies and AI. The project is supported by a consortium of partners and co-financed by the European Union and regional entities (Interreg VI).

References

[1] Böckmann E, Pfaff A, Schirrmann M and Pflanz M 2021 *Scientific Reports* 11 10419
[2] Domingues T, Brandão T, Ribeiro R and Ferreira J C 2022 *Agriculture* 12 1967
[3] Kapetas D, Christakakis P, Faliagka S, Katsoulas N and Pechlivani E M 2025 AgriEngineering 7 29
[4] Zhang X, Bu J, Zhou X and Wang X 2023 Frontiers in Plant Science 14 1255719
[5] Kapetas D, Christakakis P, Faliagka S, Katsoulas N and Pechlivani E M 2025 *AgriEngineering* 7 29
[6] Montgomery G A, Belitz M W, Guralnick R P and Tingley M W 2020 Frontiers in Ecology and Evolution 8 579193
[7] Sileshi G 2006 World Agroforestry Centre 62 268
[8] Kapetas D, Christakakis P, Faliagka S, Katsoulas N and Pechlivani E M 2025 AgriEngineering 7 29
[9] Kumar N, Nagarathna and Flammini F 2023 *Agriculture* 13 741