Revista de Informática Teórica e Aplicada - RITA - ISSN 2175-2745
Vol. 32, Num. 1 (2025) 47-53

RESEARCH ARTICLE

Automatic Detection and Counting of Tuta Absoluta Insect in Trap Images

Detecção e Contagem Automática do Inseto Tuta Absoluta em Imagens de Armadilhas

Gabriel Santos de Souza[1]*, Camila Corrêa Vargas[2], Jean Carlo Hamerski[1]

Abstract: The integration of artificial intelligence (AI) into agriculture offers solutions to challenges such as pest control. AI can improve productivity and sustainability through precision agriculture. This paper presents an automatic system for identifying and counting *Tuta absoluta* pests in trap images, integrated into a monitoring platform. The platform uses a biological defensive system for sustained pest control. The solution employs ImageAI deep learning algorithms to detect and classify pests, using YOLOv3 and TinyYOLOv3 models. We provide assessments of the performance and resource consumption of the evaluated models. YOLOv3 achieved a detection precision of 95.28% in images with 10-50 insects, decreasing to 87.51% for around 100 insects. Despite YOLOv3 demonstrating higher precision in the detection of the number of insects, the Tiny YOLOv3 model was shown to be 4.5 times faster in the training process and occupies almost 8 times less storage space.
Keywords: Biological Pest Control — Precision Agriculture — Computer Vision — Deep Learning — *Tuta absoluta*

Resumo: A integração da inteligência artificial (IA) na agricultura oferece soluções para desafios tal como o controle de pragas. O emprego de IA pode melhorar a produtividade e a sustentabilidade por meio da agricultura de precisão. Este artigo apresenta um sistema automático para identificação e contagem de da praga denominada *Tuta absoluta* em imagens de armadilhas de campo, integrado a uma plataforma de monitoramento de pragas. A plataforma usa um sistema defensivo biológico para controle sustentado de pragas. A solução emprega algoritmos de aprendizado profundo baseado na biblioteca ImageAI para detectar e classificar pragas, usando os modelos YOLOv3 e TinyYOLOv3. Apresentamos resultados de desempenho e consumo de recursos dos modelos avaliados. O YOLOv3 atingiu uma precisão de detecção de 95,28% em imagens com 10-50 insetos, diminuindo para 87,51% quando o número de insetos aumenta para cerca de 100. Apesar do YOLOv3 demonstrar maior precisão na detecção do número de insetos, o modelo Tiny YOLOv3 mostrou-se 4,5 vezes mais rápido no processo de treinamento e ocupou quase 8 vezes menos espaço de armazenamento.
Palavras-Chave: Controle Biológico de Pestes — Agricultura de Precisão — Visão Computacional — Aprendizado Profundo — *Tuta absoluta*

[1] *Federal Institute of Education, Science and Technology of Rio Grande do Sul (IFRS), Brazil*
[2] *BioIn Biotecnologia, Business Incubator of the Biotechnology Center (IECBiot) - UFRGS, Brazil*
***Corresponding author**: 2023009655@aluno.restinga.ifrs.edu.br
DOI: http://dx.doi.org/10.22456/2175-2745.143522 • **Received:** 24/10/2024 • **Accepted:** 02/12/2024

1. Introduction

In 2023, Brazil's conventional food sector achieved revenue of R$ 1.16 trillion, corresponding to a volume of 10.8% of the national GDF 10% of the national Gross Domestic Product (GPD) [1]. Specifically, organic food production, which is characterized by the absence of chemical pesticides, is expected to grow 27% in Brazil by 2025, reaching revenues of more than R$ 127 billion [2]. Both conventional and organic agriculture production face significant challenges in the effective management of pests and diseases, which are responsible for an estimated reduction in potential crop yields ranging from 20% to 40% [3].

The challenge of pest control has traditionally been managed using chemical pesticides. However, numerous pests have developed resistance to these methods. Furthermore, the current availability of pesticides for organic farmers is inadequate to meet market requirements in terms of logistics, effectiveness against specific pests, and scalability for extensive production.

Companies such as BioIn Biotecnologia[1] (BioIn) have developed eco-friendly biological solutions to control pests. These biological products are integrated into production through precision agriculture, which uses pest monitoring software to provide an accurate application. BioIn's flagship product, *BIOIN-TRICHO-P* is a biopesticide comprising *trichogramma pretiosum* microwasps. When these microwasps are introduced into the fields, they seek out and parasitize pest eggs, thus avoiding crop damage caused by them. Figure 1 shows the cycle of this process. To achieve the expected effectiveness, the wasps must be released in the field when the pest is still in the egg stage.

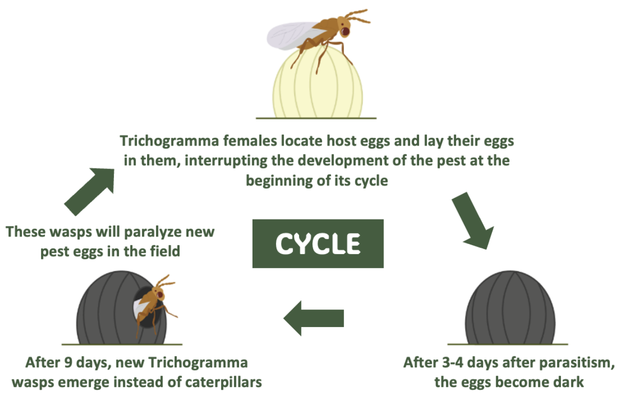

Figure 1. QR Code Scan

Recent works have explored the use of innovative solutions to resolve traditional agricultural issues [4] such as the BioIn case. They explore the widespread use of smartphones, using sophisticated technologies that include image processing methods, Artificial Intelligence (AI), Deep Learning (DL) and Machine Learning (ML) [5–10].

In order to achieve the desired efficacy, BioIn has created a decision-making mechanism supported by the *Monitora* platform. It is designed to track the entry of pests into agricultural fields, providing producers with alerts about the optimal timing for pesticide application, infestation levels, and the most affected areas. One of the main pests that affects tomato production is the *Tuta absoluta* insect. Figure 2 shows the development stages of the insect. In BioIn's solution, the adult stage of *Tuta absoluta* is detected through the use of baited traps that collect male species of the pest, indicating the presence of females in the field laying your eggs on the plantation.

To accelerate the decision-making process on the *Monitora* platform, this paper proposes a solution for the automated identification and counting of *Tuta absoluta* pests in images taken from baited traps, integrated into the platform. This solution utilizes autonomous computer vision (CV) capabilities to enhance the accuracy and efficiency of this process. The key benefit of this approach is its ability to significantly scale up the solution, an improvement over the previously manually managed human process.

[1] https://bioinagro.com.br/

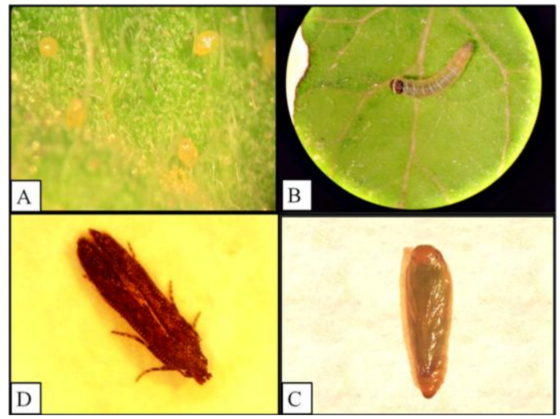

Figure 2. *Tuta absoluta* development stages: A – adult, B – egg, C - caterpillar, D – pupa [11].

The main contribution of this paper is to present a performance and resource usage evaluation of the AI library used for the AI training and insect detection processes. As given in the next section, only a few studies present similar comparisons, but the solutions they evaluate are not useful for our case study, since the application domains are radically different and less restricted when compared to the insect trap images domain.

2. Related Works

Recent advances in AI and CV have resulted in very powerful tools for the detection and classification of pests on a wide crop base. This section presents a general overview of advanced methods that have been applied to deal with this problem, compared to the approach adopted in our work.

Christakakis et al. [5] developed a mobile tool for real-time *Tuta absoluta* detection in tomato crops using the deep-learning model YOLOv8. It detects the pest itself, but also the affected areas of the leaves. The application, combined with large image databases, is a method for the diagnosis and management of plant diseases. Our work otherwise focuses on the detection and counting of *Tuta absoluta* insect in traps images, allowing for a more controlled and targeted monitoring approach for this purpose.

Nasim et al. [6] performed a comprehensive review of image processing techniques aimed at detecting pests in banana plantations. Their study evaluated 22 different works, noting that deep learning approaches, particularly *YOLOv3*, have enhanced pest detection accuracy by up to 92%. However, these techniques still face the challenge of achieving the precision needed for the identification of small-scale pests in agricultural environments. It is a general analysis of multiple studies and contrasting with our research, which focuses on custom detection and classification, performing the current training based on an image dataset of *Tuta absoluta* pest within the traps.

Rahman and Ravi [7] discussed various AI-based tools such as mobile applications (e.g. Plantix, FAMEWS, and

Nuru) and systems (e.g. drone-based monitoring), which are designed to assist in identifying and managing pest populations. The study focuses on applying and analyzing technologies directly to plantations and evaluating several tools already available.

Bütüner et al. [8] and Georgantopoulos et al. [9] concentrated on identifying the damage induced by *Tuta absoluta* rather than the pest itself. The former employed an extensive dataset of 1000 images and achieved a predictive accuracy of 98% by incorporating background elements directly into the training process, a notable improvement over previous techniques requiring manual background elimination. The latter integrated Near-Infrared and RGB channels to improve the detection of lesions in tomato plants caused by *Tuta absoluta* and *Leveillula taurica* pests. This combination of visible and infrared data allowed for a more thorough and detailed examination of plant conditions, resulting in a better detection accuracy.

Lastly, Ullah et al. [10] reviewed the limitations of deep learning models used for pest classification. The study pointed out that while these models have shown success in controlled laboratory environments, their performance often declines in real-world settings due to factors such as complex backgrounds, lighting conditions, and the diverse appearances of pests. The authors emphasize the importance of developing models capable of generalizing across different datasets and real-world scenarios to improve pest recognition.

The cited works present important proposals for the domains they address, but do not have direct application to the research objective of this research, which focuses on the automatic identification and counting of *Tuta absoluta* using traps with a customized detection approach. This method is designed to enhance the scalability of agricultural monitoring solutions and specifically targets the prevention of pest infestations by enabling precise interventions.

3. Proposed Approach

This section offers an in-depth explanation of the method adopted in this work, including pest monitoring, pest detection, and deep learning model and training processes, in addition to presenting the evaluation metrics used in experiments.

3.1 Pest Monitoring Process

Figure 3 illustrates a high-level overview of BioIn's Monitora platform. Here are the steps of the solution:

- ***Trap Installation:*** The first step involves setting up traps in the field. These traps are strategically placed in areas prone to pest activity and are designed to attract harmful insects. Each trap is equipped with a unique QR code that is geolocated for precise monitoring.

- ***Opening the Application***: User uses a mobile application specifically designed for this monitoring process. Upon arriving at a trap location, they open the application, which is linked to the *Monitora* platform.

- ***QR Code Scan:*** User scans the QR code attached to the trap, linking the captured image and subsequent data to the specific trap's geolocation. After scanning the QR code, the application automatically activates the camera system of the mobile device. The user is then prompted to take a clear photo of the trap's surface, where insects are caught. The application is supported by a remote service to which the image is sent. The remote service is augmented with CV capabilities to detect and count the number of *Tuta absoluta* insects.

- ***Generating Insect Infestation Reports:*** The data generated by the remote service are used o create infestation management reports and provide the producer with technical advice on the optimal timing and location for applying biological pesticides.

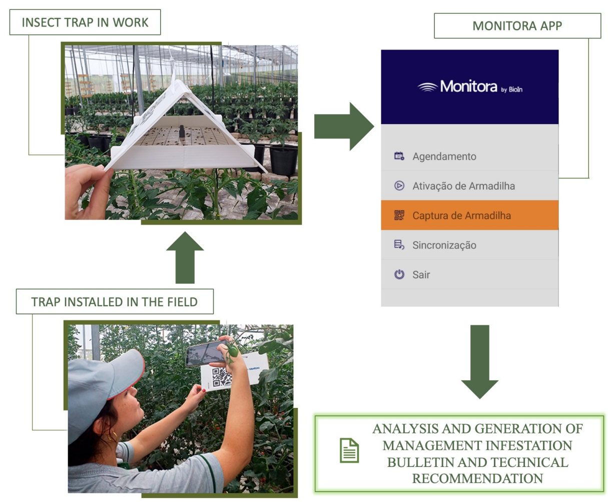

Figure 3. BioIn's complete solution schema

3.2 Pest Detection Process

The pest detection and counting process is handled by the remote service, located on a server. The remote service uses deep learning techniques to automatically perform this process. It was developed using Python, supported by *ImageAI*[2] library. The detection model chosen to be evaluated in our experiments was *YOLOv3* (and your variant, *TinyYOLOv3*), which is well known for its accuracy and speed in scenarios involving insects, as observed in the related works. A dataset from BioIn, containing 225 training images, 67 validation images, and 15 test images of *Tuta absoluta*, was utilized to perform the model's training.

3.3 Deep Learning Model and Training

YOLOv3 and its lightweight variant, *TinyYOLOv3* (*TYOLOv3*), were selected for this case study. *YOLOv3* is a real-time object detection model that works by predicting bounding boxes and class probabilities directly from full images in a single evaluation. This approach makes *YOLOv3* extremely fast compared

[2]https://github.com/OlafenwaMoses/ImageAI

to other methods that require region proposals or multiple passes through the image.

TinyYOLOv3 offers a balance between speed and accuracy, making it suitable for environments with limited computational resources. Both models were implemented using the *ImageAI* library, which simplifies the process of setting up and running object detection models.

3.4 Evaluated Metrics

The performance of the evaluated models was assessed using the following metrics:

- ***mAP (mean Average Precision):*** *mAP* is a robust measure that evaluates the accuracy of object detection by balancing both localization and classification performance. *mAP* was calculated with an Intersection over Union (IoU) threshold of 0.2. This lower IoU threshold was chosen because the primary objective of the research focuses on accurate object counting rather than achieving high precision in the size and position of bounding boxes.

- ***Recall:*** Recall quantifies the model's ability to detect all instances of a given class. Specifically, in the context of *YOLO*, it measures the proportion of true positives among all actual positives, reflecting the model's sensitivity to detecting objects.

- ***Precision:*** Precision assesses the proportion of true positives among all predicted positives. For *YOLO*, this metric is crucial for understanding how well the model avoids false positives, thereby ensuring that detected objects are indeed relevant. *mAP*, Recall, and Precision are calculated by the model during the validation of the training process.

- ***MPV (Manual Precision Verification):*** *MPV* is a metric created by us to represent a manual counting approach for verifying the model's precision. This manual verification helps cross-check the accuracy of the model's predictions, ensuring that the automated counts are reliable.

- ***HPIP (High Populated Images Precision):*** HPIP is a metric created by us to represent the model's precision when dealing with highly populated images, typically containing 100 or more insects.

4. Experiments

For the experiments, a set of images taken in BioIn traps positioned in real tomato plantations were collected. Figure 4 shows one of the images used in model training with dozens of *Tuta absoluta* insects.

The models were trained using the following hardware setup: Intel® Core i5-10400F processor, GTX 1050 Ti GPU with 4GB VRAM, and 8GB of RAM. The necessary tools and libraries used in experiments was as follows:

Figure 4. Trap cropped image.

- ***Python***: version 3.9.0.

- ***LabelImg Tool***: version 1.8.6, it was used to label the insects in the collected images, necessary in the training process.

- ***ImageAI library***: version 3.0.3, it is an open-source Python library that facilitates object detection through *YOLOv3* and *TinyYOLOv3* models.

- ***ImageAI Requirements:*** the main dependencies required by *ImageAI* included Pillow (>= 7.0.0), Numpy (>= 1.18.1), OpenCV-Python (>= 4.1.2), and Torch (>= 1.9.0) along with Torchvision (>= 0.10.0).

For the setup of the **training process**, two folders are created within the project directory to store the labeled images and their corresponding annotations: *'train'*, which is designated for 80% of the labeled images, and *'validation'*, which contains the remaining images.

The training file used in the experiments, shown in Figure 5, contains the training model specification, the directory of the labeled image dataset, and those specifications. Those include the class names (in this case, the categories of insects to be recognized) identified in the dataset, the batch size (the number of image examples to be processed in a single iteration during the training), and the number of epochs (number of complete passes through the entire dataset during the training).

Initially, multiple training sessions were conducted using the *YOLOv3* and *TinyYOLOv3* models, varying the number of epochs up to 9000. It was observed that after Epoch 40 there were little or no significant progress in terms of *mAP* and *Precision*, regardless of the model or training duration (TD). Based on this, we concluded that 100 epochs would be sufficient for the experiments in this case study.

For the setup of the **detection process**, the following parameters was adjusted to enhance detection performance, as shown in Figure 6:

```
from imageai.Detection.Custom import
    DetectionModelTrainer

trainer = DetectionModelTrainer()
trainer.setModelTypeAsYOLOv3()
trainer.setDataDirectory(data_directory="armadilha
    /insetos")
trainer.setTrainConfig(object_names_array=["tuta-
    absoluta"],
batch_size=3, num_experiments=100)
trainer.trainModel()
```

Figure 5. Training file configuration.

- ***nms_threshold***: This parameter, set to 0.03, is used to control the Non-Maximum Suppression (NMS) process, which helps to reduce the number of overlapping bounding boxes by keeping only the most confident ones. Adjusting the *nms_threshold* allows for balancing the trade-off between detecting overlapping objects and avoiding multiple detections of the same object.
- ***objectness_threshold***: Set to 0.1, this threshold determines the minimum confidence score required for a prediction to be considered as containing an object. By tuning this parameter, we aim to filter out predictions with low confidence, thus improving the detections precision.
- ***minimum_percentage_probability***: This parameter ensures that only predictions with a probability greater than a certain percentage are considered valid (20 in the case). This helps in further refining the detection results by excluding low-confidence detections and reducing the incidence of false positives.

5. Results and Discussion

In this section, we present the results observed in the training process when comparing the *YOLOv3* and *TinyYOLOv3* models (subsection 5.1). In addition, we conducted an experiment to evaluate the accuracy of automatic detection processes compared to manual detection (subsection 5.2). Lastly, we discuss general insights (subsection 5.3).

5.1 Training Process: Comparison between *YOLOv3* and *TinyYOLOv3* models

Figure 7 shows the learning curve graph for the metrics evaluated in the training process for the *YOLOv3* and *TinyYOLOv3* models. The graph presents the results up to epoch 42, as the following epochs do not show significant variations. For the *YOLOv3* model, the full 100-epoch training with a batch size of 3 lasted 83 minutes, reaching its *mAP* peak at epoch 35 with a value of 84.9%. For the *TinyYOLOv3* model, also trained for 100 epochs, had a total duration of 18 minutes and 42 seconds, with the highest *mAP* of 81.93% achieved at epoch 41.

```
from imageai.Detection.Custom import
    CustomObjectDetection

detector = CustomObjectDetection()
detector.setModelTypeAsYOLOv3()
detector.setModelPath("armadilhas/insetos/models/
6-0.55275_epoch-26-.07-.001-1.0.pt")
detector.setJsonPath("armadilhas/insetos/json/SL
    -6.json")
detector.loadModel()
detections = detector.detectObjectsFromImage(
    input_image="armadilhas/teste_artigo/510.jpg",
    output_image_path="armadilhas/teste_artigo/6/D
    -510.jpg",
    display_box=True,
    extract_detected_objects=False,
    minimum_percentage_probability=20,
    display_percentage_probability=False,
    display_object_name=False,
    custom_objects=None,
    nms_treshold= 0.03,
    objectness_treshold= 0.1
)
for detection in detections:
    print(detection["name"], " : ",
    detection["percentage_probability"], " : ",
    detection["box_points"])
```

Figure 6. Detection file configuration.

Although the *TinyYOLOv3* model has started with better numbers in the initial epochs and has exhibited a stable curve throughout the training, the *YOLOv3* model stabilized at a higher *mAP* value. Notably, *YOLOv3* experienced a significant surge in *mAP* starting from epoch 4, leading to its superior performance.

Another important observation is that when the *YOLOv3* model reaches its highest *mAP* value at any epoch, the model file is not updated unless a higher *mAP* is achieved in subsequent epochs. This means that after reaching its peak at epoch 35, even if the *mAP* values fluctuate and decrease in the following epochs, the model file remains unchanged, retaining the value achieved at epoch 35 throughout the remaining 100 epochs of training.

5.2 Detection Process: Accuracy of automatic detection processes

In this experiment, we first performed a manual count of insects in a set containing 13 images to evaluate the average accuracy of the models. Table 1 shows the results. By doing the *MPV* to validate the real accuracy of the model, the *YOLOv3* model achieved an *MPV* of 90.16% , while *TinyYOLOv3* obtained 83.61%. Both models demonstrated high *MPV* value in images with a moderate population of insects (10-50), achieving around 95%. However, for images with a large number of insects (around 100), such as in Figure 8, there was a reduction in HPIP, reaching 87.51% and 72.16% for *YOLOv3* and *TinyYOLOv3*, respectively.

As observed in Table 1, while the overall difference in precision between the models is relatively small in both *MPV*

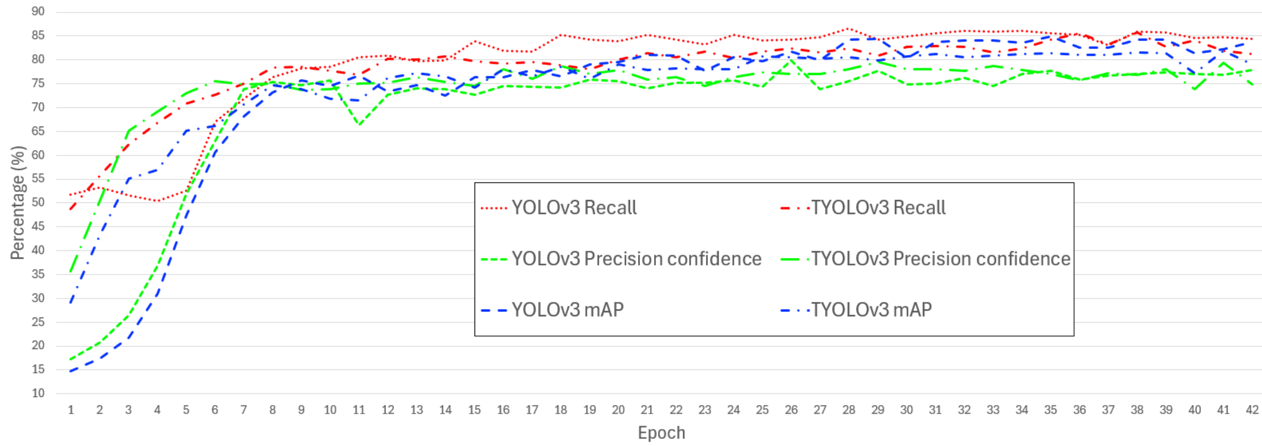

Figure 7. Learning Curve of Metrics over Epochs for *YOLOv3* and *TinyYOLOv3* (*TYOLOv3*) Model

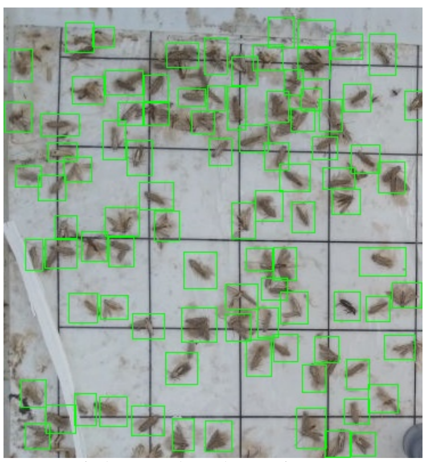

Figure 8. High populated trap image (*YOLOv3*).

and *mAP*, *TinyYOLOv3* experienced an even more pronounced drop in performance compared to *YOLOv3* when dealing with images containing large populations of insects. This indicates that while *TinyYOLOv3* is suitable for many use cases, it may struggle significantly in scenarios involving densely populated insect clusters, where *YOLOv3* demonstrates greater accuracy.

Table 1. Comparison accuracy between the yolov3 and *TinyYOLOv3* models.

Model	mAP	MPV	HPIP	TD
YOLOv3	84.90%	90.16%	87.51%	1h 23min
TinyYOLOv3	81.93%	83.61%	72.16%	18min 42s

From another point of view, *TinyYOLOv3*, although slightly less accurate than *YOLOv3*, proved to be extremely faster in the training process and more compact considering the generated model file, making it an attractive option for applications where storage resources are more constrained. The generated *YOLOv3* model file presented a size of 230 MB, while *TinyYOLOv3* model file is only 30 MB.

5.3 Discussion

Despite the promising results, it is important to note the limitation of the dataset used. Many references recommend an average of 1000 images per class for effective training of models like *YOLO*. This limitation was reflected in the model's difficulty in detecting insects that are very close to each other. To address this, strategies such as increasing the dataset size and diversity could significantly enhance the model's performance. Additionally, experimenting with different labeling techniques, such as creating a specific label for scenarios where many insects are clustered together, or distinguishing between insects with open wings and closed wings, may help refine detection accuracy.

A potential adjustment to improve detection in situations where insects are clustered would be to increase the *nms_threshold* parameter in the detection file. However, this adjustment could lead to a significant increase in the number of false positives, with the model repeatedly marking already detected insects or identifying insects where there are none.

In addition, both models demonstrated consistent precision and coherence in various scenarios of lighting conditions and image resolutions, as observed in Figure 9 compared to the other images, further corroborating their adaptability to real-world situations where such variables can often fluctuate.

Employing newer models such as YOLOv4, YOLOv5, or YOLOv8 may enhance accuracy and more effectively manage clustered insects due to advancements in architecture.

6. Conclusion

The experiments showed that even with a relatively small dataset, the *YOLOv3* and *TinyYOLOv3* models were able to achieve satisfactory accuracy in different scenarios and conditions, with *YOLOv3* obtaining the best results. However, to improve the robustness of the models in more challenging scenarios, it would be necessary to significantly increase the number of training images and carefully adjust the detection parameters to avoid generating false positives.

It is also important to consider that while some other works

Figure 9. Trap image detected with *YOLOv3*.

report higher precision, they often operate in vastly different contexts. For example, some studies use high-resolution images captured in laboratory settings, images with significant zoom for fine classifications, or detection in fields where insect populations are not as densely clustered. In contrast, our work focuses specifically on the prevention and monitoring of insects within traps. The goal is to track increases or decreases in insect populations, aiding in the decision-making process for the use of biological control agents. In future work, we pretend to evaluate the *TinyYOLOv3* model to use it directly in the *MONITORA* application, eliminating the need for communication with a remote server, making the process even more efficient.

Automated pest detection offers significant practical benefits for agricultural pest management. By enabling precise monitoring of pest populations, this technology promotes more rational use of resources, particularly biopesticides, which have limitations in large-scale applications. It also supports better decision-making regarding the optimal timing for biopesticide application. This quantitative feedback can help assess their effectiveness and determine if alternative approaches are necessary. In future work, a detailed quantitative analysis of these practical benefits will be performed to validate the broader impact of this research on agricultural efficiency and sustainability.

Acknowledgements

This study was supported by the Federal Institute of Education, Science and Technology of Rio Grande do Sul (IFRS), developed in partnership with BioIn Biotecnologia, at the Innovation Habitat InovaLab@Restinga located in the IFRS - Campus Restinga.

Author contributions

Authors Gabriel Santos de Souza and Jean Carlo Hamerski contributed to the development of the code, tests, and the writing of the article. Author Camila Corrêa Vargas contributed with her technical knowledge on agricultural biodefensives and review of the text.

References

[1] ABIA. *Notícias - ABIA - Associação Brasileira da Indústria de Alimentos*. 2024. ⟨https://www.abia.org.br/noticias/industria-de-alimentos-do-brasil-gera-70-mil-vagas-de-emprego-em-2023⟩. (Accessed on 09/05/2024).

[2] BANDNEWS. *Mercado de alimentos saudáveis deve faturar R$ 127 bilhões*. 2024. ⟨https://bandnewsfmcuritiba.com/mercado-de-alimentos-saudaveis-deve-faturar-r-127-bilhoes/⟩. (Accessed on 09/05/2024).

[3] INVESTING. *Defensivos Agrícolas e Bioinsumos — Investing.com*. 2022. ⟨https://br.investing.com/analysis/defensivos-agricolas-e-bioinsumos-200449414⟩. (Accessed on 09/05/2024).

[4] ALETDINOVA, A. A. Popular mobile applications for crop production. *IOP Conference Series: Earth and Environmental Science*, IOP Publishing, v. 666, n. 3, p. 032036, mar 2021.

[5] CHRISTAKAKIS, P. et al. Smartphone-based citizen science tool for plant disease and insect pest detection using artificial intelligence. *Technologies*, v. 12, n. 7, 2024. ISSN 2227-7080.

[6] NASIM, S. et al. Artificial intelligence techniques for the pest detection in banana field: A systematic review. *Pakistan Journal of Biotechnology*, v. 20, n. 02, p. 209–223, Jun. 2023.

[7] RAHMAN, S. M.; RAVI, G. Role of artificial intelligence in pest management. *Current Topics in Agricultural Sciences Vol*, v. 7, p. 64–81, 2022.

[8] BüTüNER, A. K. et al. Enhancing pest detection: Assessing tuta absoluta (lepidoptera: Gelechiidae) damage intensity in field images through advanced machine learning. *Journal of Agricultural Sciences*, Ankara University, v. 30, n. 1, p. 99–107, 2024.

[9] GEORGANTOPOULOS, P. et al. A multispectral dataset for the detection of tuta absoluta and leveillula taurica in tomato plants. *Smart Agricultural Technology*, v. 4, p. 100146, 2023.

[10] ULLAH, N. et al. An efficient approach for crops pests recognition and classification based on novel deeppestnet deep learning model. *IEEE Access*, IEEE, v. 10, p. 73019–73032, 2022.

[11] VEIGA, A. C. P. *Compatibilidade entre produtos químicos e biológicos à base de Bacillus thuringiensis Berliner no controle de Tuta absoluta (Meyrick)(Lepidoptera: Gelechiidae)*. Tese (Doutorado), 2014.