Halyomorpha Halys Detection in Orchard from UAV Images Using Convolutional Neural Networks

Alexandru Dinca[1], Dan Popescu[1], Cristina Maria Pinotti[2], Loretta Ichim[1], Lorenzo Palazzetti[3], and Nicoleta Angelescu[4]

[1]University POLITEHNICA Bucharest, Bucharest, Romania
[2]University of Perugia, Italy
[3]University of Florence, Italy
[4]Valahia University of Targoviste, Targoviste, Romania
marius.dinca1411@stud.acs.upb.ro, dan.popescu@upb.ro, cristina.pinotti@unipg.it, loretta.ichim@upb.ro, lorenzo.palazzetti@unifi.it, nicoleta.angelescu@valahia.ro

Abstract. Halyomorpha Halys, commonly known as the brown marmorated stink bug, is an invasive insect that causes significant damage in orchards. Neural networks have the potential to improve insect pest detection and classification in modern agriculture, which can lead to better pest management. The detection of these insects in orchards using drones imposes special problems because the images are taken from a limited distance and the foliage of the trees makes detection difficult. In this article, we studied the possibility of detecting the respective insects using the latest generation YOLOv8 neural networks and compared the results with the well-known YOLOv5 network. The results were obviously better for YOLOv8 (accuracy of 94.55%). However, satisfactory results were also obtained in the case of YOLOv5 (accuracy of 90.91%).

Keywords: Convolutional Neural Networks, Harmful Insects, Pest Detection, Orchard.

1 Introduction

The insect pest causes damage to crop, livestock, forests, or other natural resources, causing economic losses or ecological imbalances. Halyomorpha Halys (HH), commonly known as the brown marmorated stink bug, is a species of insect in the family Pentatomidae. It is native to East Asia, including China, Japan, and Korea, but has become an invasive species in many parts of the world, including North America and Europe [1]. Following this context there are many studies available for HH and its effects as an insect pest. They provide valuable information on the behavior, biology, and management of the brown marmorated stink bug, as well as its impact on various crops and industries [2-5].

HHs feed on a variety of plants, including fruits, vegetables, and ornamental plants, and can cause significant damage to crop [2]. Some of the main crops that are affected by this invasive species include apples, grapes, peaches, and soybeans [3]. The invasive nature of the brown marmorated stink bug has caused concern among farmers and homeowners, and efforts are being made to control its spread and limit its impact on agriculture and the environment [4].

Regarding insect pests, effective management strategies include technological, biological, and chemical methods to control their populations and reduce their impact on ecosystems and agriculture [5]. Deep learning and neural networks are modern ways of detecting insect pests in the field. In recent years, there has been a growing interest in using machine learning techniques to automate pest detection in agriculture [6]. However, it is important to note that these models require significant amounts of high-quality training data and computational resources to develop and deploy [7-12]. Additionally, the accuracy of the model's predictions can be affected by factors such as lighting conditions, camera angles, and the diversity of insect pests in the field. Therefore, careful calibration and validation of the model are necessary to ensure reliable pest detection.

Deep learning has become popular in modern agriculture due to its ability to analyze and process large amounts of data efficiently. With the increasing availability of sensors and data collection devices, agriculture has become a data-rich industry. Deep learning algorithms can process this data and extract valuable insights that can help farmers make informed decisions. For precision agriculture, deep learning algorithms can analyze various environmental and crop-related data [6]. On the other hand, deep learning algorithms can analyze images of crops taken from drones, digital cameras, or other devices and identify patterns of growth, disease, and pest infestation [7]. This information can help farmers make timely decisions about crop management. To detect insect pests using deep learning and neural networks, the need is to train the models using large datasets of images of both healthy and infested crops [8, 9, 11]. The model would learn to recognize patterns in the images that distinguish healthy plants from infested ones. Once trained, the model can then be used to automatically detect pest infestations in new images of crops taken in the field [12].

Neural networks have the potential to improve insect pest detection and classification in modern agriculture, which can lead to better pest management and higher crop yields. One common approach is to use image classification techniques based on convolutional neural networks (CNNs) to identify and classify insect pests. This involves training a neural network on a large dataset of images of crops and insects, where the images are labeled with information about the presence of insect pests. The neural network learns to recognize the patterns and features of the insects in the images and can then classify new images as containing a specific insect pest or not.

This paper is organized in four sections, presenting the details that were the basis of the study. Considering the presented introduction section, the rest of the sections are organized as follows: Section 2, Materials and Methods, presents the data set used to develop the present work in relation to the object detection task for HH, the models used are presented. On the other hand, the hardware and software part used to implement neural networks is also noted. Section 3 presents the experimental results, performances, and corresponding discussions. Finally, a short section of conclusions is presented.

2 Material and Methods

2.1 Dataset Used

To develop and train accurate neural networks for pest detection, a substantial amount of data is required, which can be a challenge in some agricultural contexts. A proprietary dataset was created for this work using real images taken in the orchard field. The dataset tracks the presence of the HHs insect pest in the images. A custom dataset was designed and implemented for the purpose of YOLO (You Only Look Once) family to HH detection. Transfer learning applied to a custom dataset with images in the field and fine-tuning hyperparameters were followed for this study. The data set is proprietary, and no external or public image databases were queried or included in this study.

Because this work relies on an object detection task, LabelImg [13] was the choice to create the labeled data to train the YOLOv5 and YOLOv8 models to detect object classes in our case. The labels were exported to YOLO format, using a .txt file to describe the manually bounding box labeled objects in each image, with one object in each row. A dataset partition was implemented to group images in each directory for training, validation, and testing. The created dataset consists of 312 images divided as follows: 70% for training (218 images), 20% for validation (62 images), and finally 10% for testing (32 images). For the data set created, it is important to mention that the images may contain one or more instances of the insect of interest.

The images in the dataset depict a real context represented by various orchard images taken using a precision camera drone. The images taken from the drone went through a pre-processing step before being fed into the training part of the network. An example of images from the training set can be viewed in Fig. 1 below. In this way, manual identification of regions of interest was pursued and image patches of size 640 × 640 pixels were gathered to create the mentioned dataset. The image size followed is the one from the official YOLOv8 documentation.

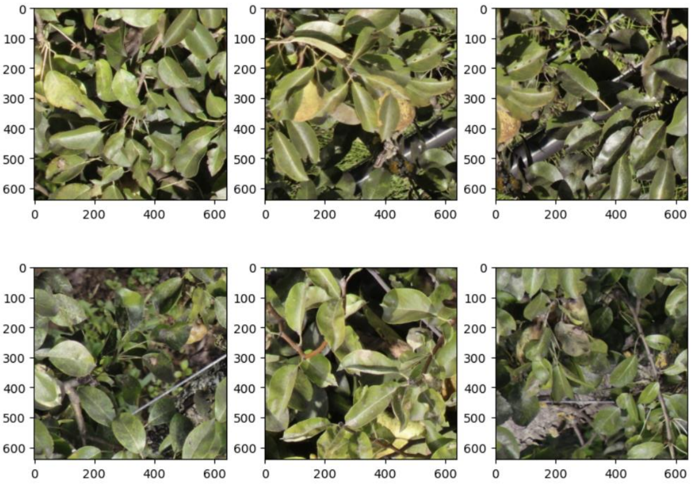


Fig. 1. Example of training images.

2.2 Neural Networks Used

Pre-trained models from YOLO object detection family were used for training and evaluation for HH detection using digital images in the field. YOLO is an object detection algorithm developed in [14]. It was first introduced in 2015 and has since then become one of the most popular and widely used object detection algorithms in computer vision. The YOLO algorithm is based on a deep convolutional neural network (CNN) architecture that uses a single pass to perform object detection. Unlike other object detection algorithms that rely on region proposals, YOLO uses a grid of cells to divide the image into smaller regions and predicts the object class and location for each cell. The algorithm is trained on large datasets, such as the COCO (Common Objects in Context) dataset and has been designed to be fast and accurate. Today, YOLO is used in a wide range of applications, including autonomous vehicles, security systems, and video analytics. The YOLO algorithm has been integrated into many popular computer vision frameworks, such as OpenCV, TensorFlow, or PyTorch, and has become a standard benchmark for object detection in computer vision. The key feature of the YOLO family is its ability to perform object detection in real-time on GPU hardware.

YOLOv5 is a cutting-edge deep learning-based object recognition method based on the YOLO family of object detection models [15]. The structure of the YOLOv5 model can be broken down into several main components: input processing, backbone network, neck network, and detection head. The input image is first preprocessed by resizing it to a fixed size and then normalized by subtracting the mean pixel value and dividing it by the standard deviation. The backbone network is responsible for extracting features from the input image. YOLOv5 uses a CSPNet backbone network, which consists of a series of convolutional layers that gradually reduce the spatial resolution of the feature maps while increasing their depth. Next, the neck network is placed for combining the features extracted by the backbone and producing a set of feature maps with rich spatial information. In the case of YOLOv5 the spatial pyramid pooling (SPP) module is used, pooling features at multiple scales and capturing objects of different sizes. The detection head is the next component for YOLOv5, using a multi-scale anchor-based detection head. This one predicts a set of bounding boxes and class probabilities for each anchor at multiple scales. Using non-maximum suppression (NMS) the predicted boxes are filtered and refined to compute the final detections.

The current state-of-the-art (SOTA) release, YOLOv8, is an upgraded version from the YOLO family that has improved accuracy and speed compared to the previous versions [16]. YOLOv8 has a more complex architecture that includes multiple neural network layers and skip connections to better model object relationships and features. It has improved anchor box design and data augmentation techniques, resulting in better accuracy. At the time of the development of the present study, there was no official published paper for the new YOLOv8. The ideas developed and the analysis of the YOLOv8 architecture were based on the online documentation made available by the team that implemented the new version and the research of the open-source code [16].

As a quick summary, developed by Ultralytics, YOLOv8 offers a boost in performance and flexibility over its predecessors and still follows a wide area in object detection and image classification, and segmentation tasks. Following the Ultralytics release, all YOLOv8 models could be used as pre-trained models under different availability – YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, YOLOv8x. The models used for detection and segmentation were tested and evaluated over MS COCO (Microsoft Common Objects in Context) dataset, while the models used for classification are pre-trained on ImageNet data. The extensibility of YOLOv8 is an important feature.

The current SOTA is built as a framework that supports all prior YOLO versions, having the option to move between them and evaluate their performance. To note the extensive changes starting from the previous versions and the novelty part, YOLOV8 is described using a new head with anchor-free detection and introducing a new backbone and loss function. The anchorless addition will not follow any anchor box offset and will detect the center of the object. Also, this addition will have a big impact on Non-Maximum Suppression (NMS). Closing the mosaic augmentation was also implemented for the last epochs in the training steps. The PyTorch integration and a well-documented CLI could also describe the innovations in this YOLOv8 release.

For this study, YOLOv8 was used as the main model for HH detection using images from a pear orchard. The choice to implement also the YOLOv5 model was made to describe a comparison area with the new version in the YOLO family - YOLOv8.

As a programming language, Python version 3.10 was used for the implementation. PyTorch was considered as deep-learning framework in this context with Miniconda and Jupyter Notebook to define and organize the code. The hardware part was defined by a system that aimed to integrate a GPU component for training and evaluating the used models. In this regard, the system features an NVIDIA RTX 2080Ti GPU with 11GB memory and CUDA V11.7. The other components of the system were represented by 128GB of RAM and an Intel Core i9-11900K processor. The operating system was represented by Ubuntu V22.

3 Experimental Results and Discussions

Training experiments are based on several model settings using the YOLOv8 and YOLOv5 architectures: the number of epochs was 300, the models used were YOLOv8m and YOLOv5m, and the input image size for object detection was as each model suggests 640 × 640 pixels. The custom class for object detection to highlight the HH pest was denoted with the acronym: HH. A configuration YAML file was added to define the paths for each directory involved in the training and evaluation part. The images in the test and validation set are different from the ones the model analyzes in the training phase so that the evaluation is robust and based on new contexts and information. Both models were trained and evaluated using the same dataset. An example with validation images is shown in Fig. 2.

For both models used, training and validation accuracy and loss plots are shown in Fig. 3 for YOLOv8m and in Fig. 4 for YOLOv5m to describe the model performance. Following the object detection task, the final output for each model is a list of bounding boxes and associated class labels, along with their confidence scores. Analyzing the training results, we can note a great accuracy for the YOLOv8m model using transfer learning for object detection regarding the HH pest from real orchard context images.

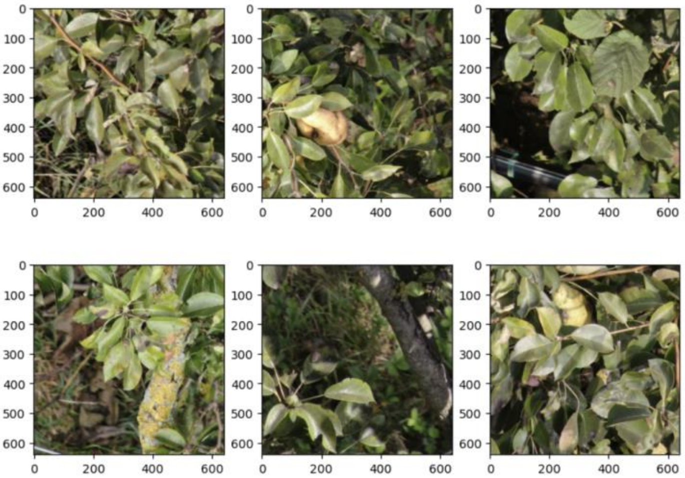


Fig. 2. Example images from the validation dataset.

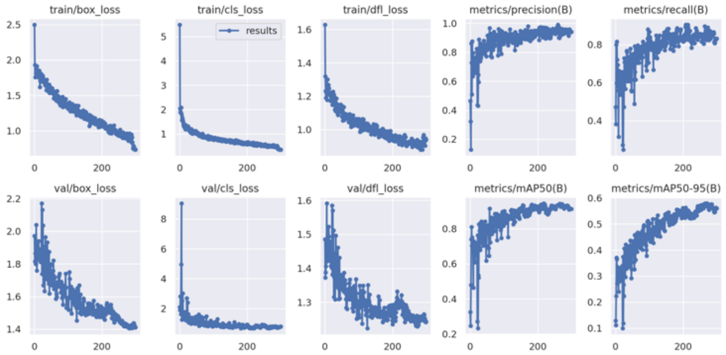


Fig. 3. YOLOv8m training and validation results.

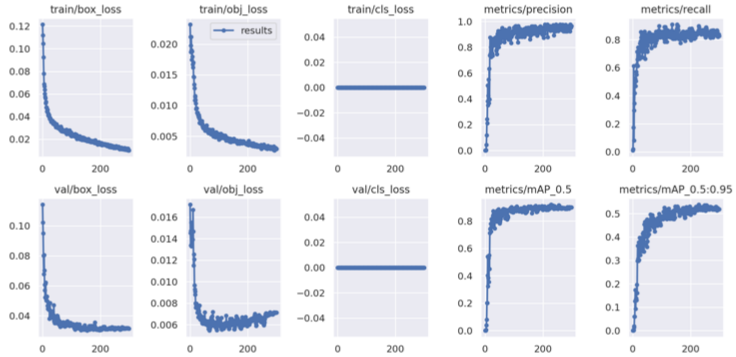


Fig. 4. YOLOv5m training and validation results.

In the same scenario for HH detection, Fig. 5 shows the computed precision-recall curve obtained for YOLOv8m and Fig. 6 for YOLOv5m. For HH object detection, this is a valuable tool for evaluating algorithm performance and making decisions about the threshold for predictions. In practice, it is commonly used to evaluate the performance of an object detection algorithm and to make decisions about the appropriate threshold for predictions. The precision-recall curve is plotted as a scatter plot with precision on the y-axis and recall on the x-axis. Each point on the curve represents a different threshold for predictions, and the curve is created by connecting the points in order of increasing the threshold. Precision is the proportion of true positive predictions out of all positive predictions, while recall is the proportion of true positive predictions out of all true positive cases. As the threshold increases, precision will typically increase while recall will decrease. The ideal curve would have a slope of 1 and achieve a precision of 1 and a recall of 1 at the highest threshold. However, in practice, the precision-recall curve will often be an S-shaped curve, to understand the balance between false positive and false negative errors.

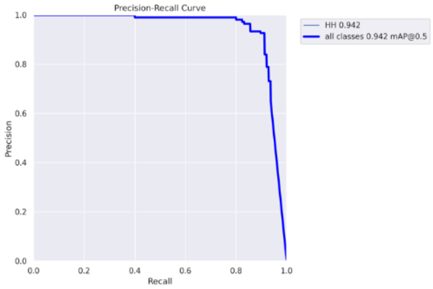


Fig. 5. The precision-recall curve obtained for HH pest detection using YOLOv8m.

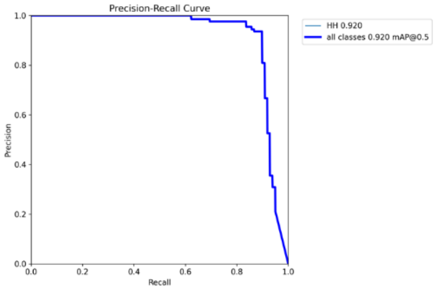


Fig. 6. The precision-recall curve obtained for HH pest detection using YOLOv5m.

Examples of the prediction part resulting from network training and analysis are shown in Fig. 7 for YOLOv8m and in Fig. 8 for YOLOv5m. It can be seen in the attached prediction figure for YOLOv5m that the model failed to identify the insect for row one, image number two, and in image number three the model identified a spot on the leaf as an HH-type insect. The same reference images were used for both models to illustrate the area of comparison for the obtained performances and predictions.

For both models, we aimed to attach the confusion matrix resulting from the training and evaluation processes to calculate the performance indicators. The figures with representative confusion matrices are attached in Fig. 9 and Fig. 10 for YOLOv8m and YOLOv5m, respectively in the case of learning and validation only with images containing HH and in Fig. 11 and Fig. 12 in the case of testing with images containing HH and images without HH (background).

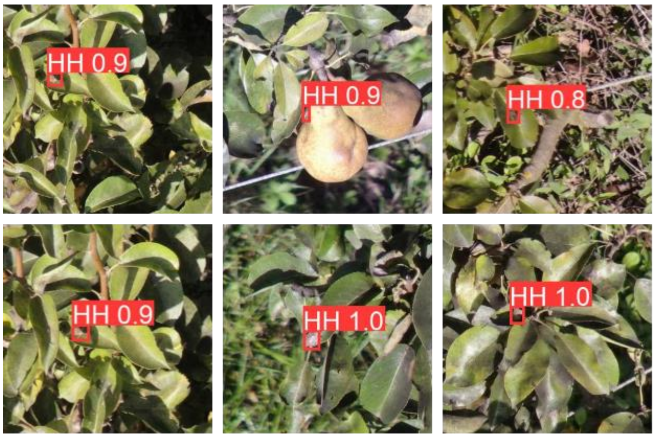


Fig. 7. Examples with HH predictions generated by YOLOv8m.

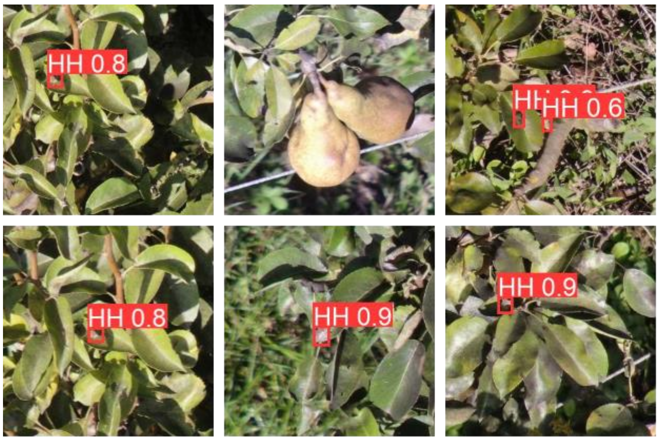
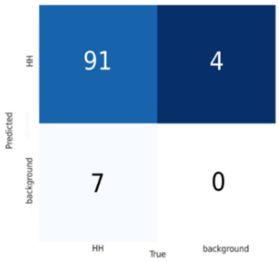


Fig. 8. Examples with HH predictions generated by YOLOv5m.



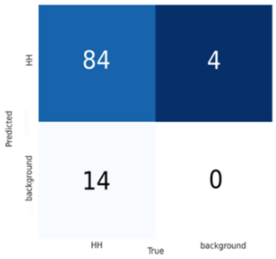


Fig. 9. The confusion matrix obtained for HH pest detection using YOLOv8m after training and validation.

Fig. 10. The confusion matrix obtained for HH pest detection using YOLOv5m after training and validation.

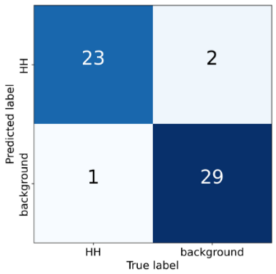


Fig. 11. YOLOv8m confusion matrix obtained using the test dataset.

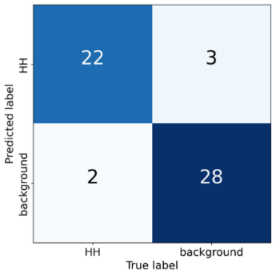


Fig. 12. YOLOv5m confusion matrix obtained using the test dataset.

For testing the network various images were considered. These contain images in which the insect of interest (HH) is present in one or more instances, images with a complex background that do not feature this insect or any other insects, and selected images containing other types of insects to test the ability used models trained for HH detection and rejection of other insects or artifacts at the digital image level. In the case of the models in the present study, it is observed that they manage with high accuracy to identify the insect of interest of the HH class and avoid the detection of artifacts from the background area or representations of other insects.

Testing the models was also done by attaching images in which the insect of interest is partially visible, illustrating various positions of it or being partially visible because it can be covered by various elements (flowers, leaves, branches) or located under various lighting areas (sun rays, shading or insects present at the base of the fruits). Representative example types were also attached to the training and validation datasets. The results are notable in these cases, the models used to be able to locate the insect of interest with great precision. Examples of this type are attached in Fig. 13a. for YOLOv8m and Fig. 13b. for YOLOv5m. We note in the first instance in Fig. 13b. – second row that the YOLOv5m network fails to correctly identify the partially visible insect in the top left. At the same time, the models used can successfully identify the presence of several insects at the image level. These are exemplified in Fig. 14a for YOLOv8m and Fig. 14b for YOLOv5m. From these examples, the trend of increased accuracy can be observed in the case of the newly developed network of the YOLO family, namely v8, although we can note the performance gradually resulting from the optimization of the v5 model.

On the other hand, the model is not perfect and sometimes manages to identify areas in the images that are not represented by the insect of interest, defining parts of the background area that are represented by the presence of spots on leaves or spots on fruits. Examples of this type are indicated in the previously attached confusion matrices (Fig. 11 and Fig. 12), where the problem of false positive and false negative detections is observed.

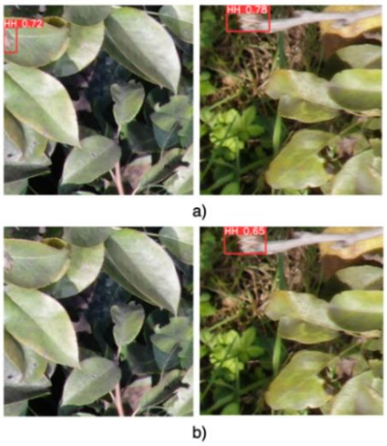


Fig. 13. Examples of HH predictions with partially visible insects.

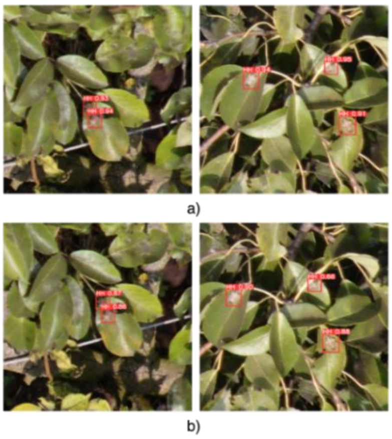


Fig. 14. Examples of predictions with multiple insects in the image.

To evaluate the performance and robustness of the models used, the test phase aimed to evaluate the networks based on images that represent real contexts, from the orchard level, in which the detection of insects of interest of the HH type is aimed. The images illustrate both areas of complex background where the insect of interest is not present, as well as images illustrating the presence of HH class insects in poses where the insect is difficult to spot, partially hidden, or even difficult to see due to differences in brightness or even of the blur that can be present from the image acquisition area. The test dataset consisted of 31 images with complex backgrounds and 19 images where the HH insect is visible. For images where the insect is visible, the number of insect instances that have been attached in this test dataset is 24. In this case, the insect may be visible in one or more instances at the image level.

To test the network, it was followed to pass the trained and validated models through the new data set to calculate accuracies and prediction areas, and finally a confusion matrix to be populated with the false and true predictions. Since the resulting confusion matrices from the validation dataset of both networks do not provide the necessary indices, it was aimed to create new instances of the confusion matrices based on the test dataset. After testing, the confusion matrices are attached in Fig. 11 for the YOLOv8m model and in Fig. 12 for the YOLOv5m model. Following the results obtained, increased performance can be seen in the case of the new YOLOv8 model. However, satisfactory results were also obtained in the case of YOLOv5, opening a research and development direction for both models. The statistical result indicators were as follows:

▪ YOLOv8m: Precision = 0.957, Recall = 0.928, mAP50 = 0.942 (training and validation), Accuracy = 94.55% (test).

▪ YOLOv5m: Precision = 0.954, Recall = 0.857, mAP50 = 0.920 (training and validation), Accuracy = 90.91% (test).

Looking from the perspective of an explanatory analysis to describe the behavior of the YOLO models in relation to the identification of the HH insect, one observes the tendency of the models to detect the object represented by the insect of interest when it is present on leaves or even on fruits. In these cases, the model identification accuracy tends to the maximum. For these image areas, the insect is depicted surrounded by a uniform background, its features being distinctly different.

It is observed that most false detections intervene when the insect is present on the branches or is in occlusion areas being covered by background areas. In these cases, the model tends to direct its attention to the leafy area, as previously mentioned, but the insect being visible in another context, in areas of shading, occlusion or in areas close to branches, the accuracy values start to decrease for such examples. The pixels representing the branch areas appear to have similar colors to those of the HH insects, and occlusions or various shading areas decrease the algorithm's ability to detect the full features of the insect. For these marginal cases, image enhancement and pre-processing techniques can bring out the characteristics of the insect much better, so that it is better differentiated from the background area.

To describe the novelty and the research directions, the implementations presented in this study represent a strong point, with notable results in research on the automatic detection and HH identification, at the level of digital images illustrating real contexts and which are based on techniques that include convolutional neural networks and related software modules. The results obtained based on neural network models bring to the fore the advantages and popularity of these automatic identification techniques. Although there are some studies based on the HH identification at the level of digital images and using deep learning techniques, from our knowledge there are no studies that focus on the identification of this insect of interest at the level of real context by using representative images or data sets in cases of various types of crops that are affected by the presence and massive spread of this insect pest. In this sense, due to the highly invasive characteristics of this population of harmful insects, we emphasize an important and necessary direction of research that can be developed along the way in the precise identification of this type of pest using various techniques and developing the recent studies attached to this topic.

4 Conclusions

In this study, a novel implementation using YOLOv8 was used for HH detection. The new version of YOLOv8 opened the road regarding the state of the art in real-time object detection and improved the accuracy of its predecessors. It is worth noting that the YOLO architecture has undergone several iterations, with each new version introducing improvements in terms of accuracy, speed, and robustness. Each new version of YOLO has built upon the previous versions and has added new features and techniques to improve object detection performance. Although the number of images was relatively small, good results were obtained through transfer learning. Unfortunately, no comparison articles were found for the investigated application.

Acknowledgment

This work was supported by HALY.ID project. HALY.ID is part of ERA-NET Cofund ICT-AGRI-FOOD, with funding provided by national sources [Funding agency UEFISCDI, project number 202/2020, within PNCDI III] and co-funding by the

European Union’s Horizon 2020 research and innovation program, Grant Agreement number 862665 ERA-NET ICT-AGRI-FOOD (HALY-ID 862671).

References

1. Haye, T., Weber, D.C.: Special issue on the brown marmorated stink bug, Halyomorpha halys: an emerging pest of global concern. J Pest Sci 90, 987–988 (2017).
2. Ivancic, T., Grohar, M.C., Jakopic, J., Veberic, R., Hudina, M.: Effect of Brown Marmorated Stink Bug (Halyomorpha halys Stål.) Infestation on the Phenolic Response and Quality of Olive Fruits (Olea europaea L.). Agronomy 2022, 12, (2022).
3. Aigner, B. L., Kuhar, T. P., Herbert, D. A., Brewster, C. C., Hogue, J. W., Aigner, J. D.: Brown Marmorated Stink Bug (Hemiptera: Pentatomidae) Infestations in Tree Borders and Subsequent Patterns of Abundance in Soybean Fields, Journal of Economic Entomology, 110(2), 487–490 (2017).
4. K., B., Rice, et al. Biology, Ecology, and Management of Brown Marmorated Stink Bug (Hemiptera: Pentatomidae), Journal of Integrated Pest Management, 5(3), A1–A13 (2014).
5. Parvizi, Elahe, et al.: Population genomic insights into invasion success in a polyphagous agricultural pest, Halyomorpha halys. Molecular Ecology 32.1, 138-151 (2023).
6. Li, W., Zheng, T., Yang, Z., Li, M., C. Sun, Yang, X.: Classification and detection of insects from field images using deep learning for smart pest management: A systematic review. Ecological Informatics, 66, (2021), 101460, ISSN 1574-9541.
7. Ayan, E., Erbay, H., Varçın, F.: Crop pest classification with a genetic algorithm-based weighted ensemble of deep convolutional neural networks, Computers and Electronics in Agriculture, (179), 105809 (2020).
8. Bereciartua-Pérez, A. et al.: Insect counting through deep learning-based density maps estimation, Computers and Electronics in Agriculture. 197, 106933 (2022).
9. Rustia, D. J., Chao, J.J., Chiu, L. Y., & Wu, Y. F., Chung, J. Y., Hsu, J. C., Lin, T. T.: Automatic greenhouse insect pest detection and recognition based on a cascaded deep learning classification method. Journal of Applied Entomology. 1-17 (2020).
10. Teng Y., Zhang J., Dong S., Zheng S., Liu L.: MSR-RCNN: A Multi-Class Crop Pest Detection Network Based on a Multi-Scale Super-Resolution Feature Enhancement Module. Frontiers in Plant Science.13:810546 (2022).
11. Zhichao, S., Dang, H., Liu, Z., Zhou, X.: Detection and Identification of Stored-Grain Insects Using Deep Learning: A More Effective Neural Network. IEEE Access. 8. 163703-163714 (2020).
12. Nanni, L., Manfè, A., Maguolo, G., Lumini, A., Brahnam S.: High performing ensemble of convolutional neural networks for insect pest image detection, Ecological Informatics, 67 (2022).
13. Tzutalin. LabelImg. Git code. https://github.com/tzutalin/labelImg (2015).
14. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection, arXiv:1506.02640 (2015).
15. Jocher, G.: YOLOv5 by Ultralytics (Version 7.0) [Computer software]. https://doi.org/10.5281/zenodo.3908559 (2020).
16. Jocher, G., Chaurasia, A., Qiu, J.: YOLO by Ultralytics (Version 8.0.0) [Computer software]. https://github.com/ultralytics/ultralytics (2023).