Proc. of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)
16-18 November 2022, Maldives

Dual Networks Based System for Detecting and Classifying Harmful Insects in Orchards

Dan Popescu
Faculty of Automatic Control and Computer Science
University POLITEHNICA of Bucharest
Bucharest, Romania
dan.popescu@upb.ro

Trandafir-Liviu Serghei
Faculty of Automatic Control and Computer Science
University POLITEHNICA of Bucharest
Bucharest, Romania
sergheiliviu@yahoo.com

Loretta Ichim
Faculty of Automatic Control and Computer Science
University POLITEHNICA of Bucharest
Bucharest, Romania
loretta.ichim@upb.ro

***Abstract*— Halyomorpha Halys is a type of a dangerous insect for agricultural crops, in general, and especially for orchards. The detection and evaluation of their abundance are of great importance for starting the elimination procedure in ecological conditions. There are few studies conducted on insect detection using deep neural networks, but none are dedicated to the Pentatomidae family. The paper proposes an intelligent system for the detection of two similar species, Halyomorpha Halys and Nezara Viridula, the first being the dangerous one. As a novelty, the paper proposes an integrated intelligent system consisting of two interconnected neural networks trained on a custom data set using transfer learning. The first is configurated for insect detection (YOLO v4-tiny) and the second is dedicated to insect classification (EfficientNet-B3). This approach has the goal of decreasing the values of false negative and false positive errors, thus increasing the detection performance.**

Keywords—image processing, deep convolutional neural networks, transfer learning decision fusion, insect detection, insect classification

I. Introduction

The problem of detecting and classifying harmful insects is of major importance for precision agriculture, where this aspect is essential for environmental protection, preventing crop infestation, combating pests, improving the production ratio per cultivated area, and food safety [1]. This involves accurate classification and rapid recognition of pests on a large scale and with accuracy to be able to intervene effectively both to limit the spread of infestation and to protect the environment through the correct use of insecticides. This context is favored by technological progress in recent years and the implementation of modern methods involving complex artificial vision algorithms that use deep learning, easy access to knowledge bases, and extensive data sets. All of this provides a new perspective on the use of modern technology and algorithms for the recognition and control of insects harmful to agricultural crops.

The problem of insect detection using deep neural networks for images acquired from UAVs (Unmanned Aerial Vehicles) has been addressed in some papers in recent years [2]. Thus, testing the use of the Inception-V3, Resnet-50, and VGG-19 networks [3] led to the conclusion that fully training them on sets gives poorer results than transfer learning and training starting from pre-trained models on large sets and various data [4], [5]. The authors of [4] conducted studies on image segmentation methods by fragmenting them into smaller regions that can later be properly analyzed.

A UAV-acquired dataset for 15 insect classes and one null class (containing leaf images on soybeans) was used in [5], with a total of 2792 images. The aim of the study was to correctly classify each existing class of insects. In this sense, the architecture models of Deep Convolutional Neural Networks (DCNN) like Inception-V3, ResNet-50, VGG-16, VGG-19, and Xception were used. These DCNNs were initially pre-trained with ImageNet. All the higher levels were kept, only the output levels being included for retraining. Later these were reconfigured and optimized, thus achieving the transfer learning. The results obtained were good, reaching a maximum accuracy of up to 93.82% for Resnet-50, 91.87% for Inception-V3, 91.80% for VGG-16, 91.33% for VGG-19, and respectively 90.52% for Xception. Results recorded without learning transfer were 50% lower.

In [6], a study was carried out on 7 DCNNs. (VGG-16, VGG-19, ResNet-50, Inception-V3, Xception, MobileNet, and SqueezeNet) pretrained on a dataset containing 40 insect classes. The study concluded that the efficiency of retraining networks is directly proportional to the amount of data. The sum of maximum probabilities is proposed as a voting method, and to avoid the problem in which several classes obtain the same score, the GAE (Genetic Algorithm Ensemble) algorithm was used, which proposes the initial weights of the network as a voting method. Following the tests performed, a maximum accuracy of 97.06% was obtained for Inception-V3, 97.93% for Xception, and 97.39% for MobileNet.

The method proposed in [7] used a fast insect identification technique by combining two architectures: one for detection and proposal of regions of interest and another for validation through a classifier.

As a result of the study of the recent works analyzed, the efficiency of the learning transfer, in conjunction with the optimization of the hyperparameters, was noted. The idea of decision fusion to increase detection accuracy is emphasized. To this end, this paper proposes the implementation of an intelligent system for the detection and recognition of harmful insects, especially, Halyomorpha Halys (HH) and Nezara Viridula (NV), belonging to the same family – Pentatomidae [8], in orchards through the transfer learning of deep convolutional neural networks. As a novelty, the paper

978-1-6654-7095-7/22/$31.00 ©2022 IEEE

proposes an integrated intelligent system consisting of two neural networks connected with the role of reducing false positive and false negative errors for the detection of insects such as HH and NV (Fig.1).

(a) (b)

Fig. 1. Detected insects: (a) Halyomorpha halys; (b) Nezara viridula.

II. Materials and Methods

A. Data Bases Used

The paper improves the classification and detection of two or more classes of insect pests by using state-of-the-art deep neural network architectures pre-trained on large datasets, ImageNet and MSCOCO, and applying transfer learning on a custom data set [6].

For the two studied insect classes, images belonging to the site [9] were used for the training set (marked DS1). They come from various sources presenting some frame acquisition flaws (focusing, lighting, point of view, various resolutions, and aspect) which could impact CNN accuracy as pointed out in [2] and [5]. Images contain insects in developmental stages ranging from instar 3 to adult (instar 6). The images of the test set were also acquired from the natural habitat of the studied insects. Some of these come from the University POLITEHNICA of Bucharest premises (marked DS2), another part is acquired from UAVs (Unmanned Aerial Vehicles) in Romania and Italy (marked DS3), and another set from Comana Natural Park (Romania) (marked DS4). The images were taken at a distance of about 50 cm and with the use of automatic and macro shooting modes to better highlight the insect. Information on the contents of the data set can be found in Table I. The latter is used for prediction testing after the training process is completed. In Fig. 2 two samples of HH in the natural environment from the DS2 and DS3 datasets are shown. Data acquisition was done taking into consideration the procedure from [5].

TABLE I. Content and Structure of Data Sets Used

Data Set	HH number	NV number	Using	Resolution (pixels)
DS1	730	756/	Training/ Validation	240×192- 2932×3008
DS2	79	-	Testing	5184×3888
DS3	130	-	Testing	5184×3888
DS4	401	55	Testing	4608×3456

Both images used for training the models and for testing them were subjected to pre-processing operations, such as a) Resizing images according to the neural networks' input; b) Cropping images or centering objects of interest; c) Contrast normalization.

Indexing of all images and creating lists of data needed to train the network are created automatically, generating lists of images for training (70%), validation (20%), and testing (10%).

(a) HH from DS2 (b) HH from DS3

Fig. 2. Samples from the test datasets.

Each region of interest corresponding to the appearance of one or more insects in the images was labeled using the Labeling application, made in the Python programming language. After marking, the class index (0 for HH and 1 for NV respectively), as well as the coordinates of the square of the region of interest, were automatically saved for each individual image in a label file.

To augment the data set with the aim of avoiding the phenomenon of overlearning and achieving an optimal generalization [6], several augmentation procedures were used through photometric and geometric distortion: angle, clipping, mirroring, saturation, exposure (luminosity), hue, and mosaic (combining four images into one).

B. Neural Networks Used

An extensive study was carried out on the existing models of architectures (backbone) of the latest generation and with special results in competitions with the role of evaluating the accuracy/rate of recorded errors [2]. The study was carried out in two directions: 1. Image Classification: Transfer Learning of Deep Neural Networks and reconfiguring the structure and hyperparameters. 2. Detecting insects of interest from the images and framing the ROI (region of interest) with a rectangular outline, displaying the belonging class as well as the percentage prediction (confidence) score. This is accomplished by transfer learning, structural reconfiguration, and hyperparameter optimization.

The following deep convolutional neural network architectures are proposed for analysis:

• For detection: YOLO v4 – DarkNet [10], Scaled-YOLOv4 [11] by transfer learning.

• For classification: EfficientNet B3 [12] by transfer learning.

The configuration of the YOLO v4 architecture started from a 137-level pre-trained model with a large data set – MSCOCO, containing 91 classes. The lower levels were reconfigured according to the particularities of the objects to be detected by introducing two classification levels corresponding to the two species of insects that are the subject of the study: HH and NV, from the data, set created especially for this purpose. The network thus trained is optimized for identifying and framing the objects belonging to the two classes of objects in a rectangular box and displaying the detection score related to the class as well as its label, using the inference process based on the prediction made. The model thus reconfigured contains 162 layers. The flowchart of the entire transfer learning process is depicted in **Error! Reference source not found.**.

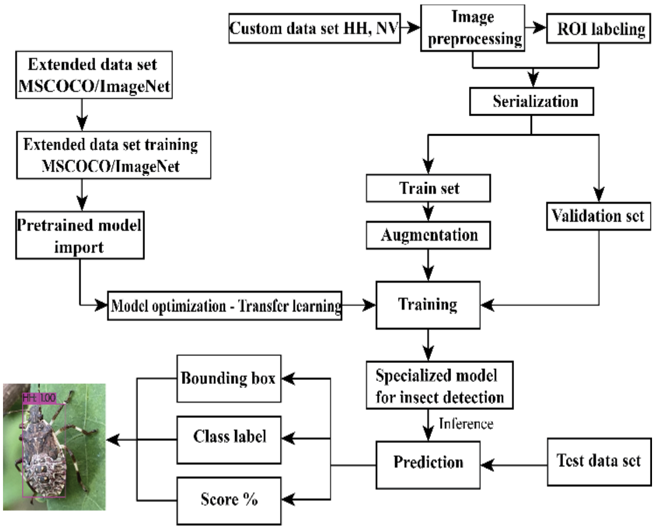


Fig. 3. Transfer learning process for insect data set.

The Scaled-YOLO v4 architecture [11] was chosen to train the HH and NV insect detection model because it is focused on the optimal combination, automatically, for each stage of the parameters depth, initial width, slope, quantization, the ratio of linearization layers (bottleneck), and group width, to achieve maximum efficiency. The tiny model also considers the resources of the working environment (especially for local training, not the cloud). Numerous optimizations are made, including memory and bandwidth allocation. From the point of view of functionality, the functions regarding the order of operations within the layers are optimized. To reduce the size of the feature map, an OSANet (One-shot aggregation) backbone (Fig. 4) is used, which offers greater efficiency than DenseNet-type dense blocks.

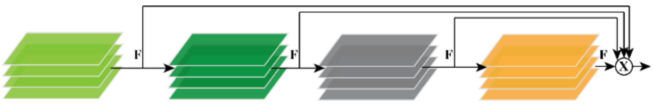


Fig. 4. OSA (One Shot Aggregation) aggregation method.

This is the first step in identifying the two insect pests. The framed regions of interest are automatically extracted and saved to later be analyzed using a DCNN (EfficientNet) based classifier to reduce errors.

EfficientNet [12] is a model made by Google, with an innovative concept, to balance three essential parameters in structuring a DCNN: depth, width, and resolution. It should be noted that EfficientNet is very well behaved providing top results (in terms of accuracy) while maintaining a lower resource requirement by using fewer parameters. The scalable structure of this architecture allows the division into 8 performance models, which are designed, depending on the complexity, from B0 to B7. Taking this aspect into account, the B3 module was proposed for use as a classifier.

In order to carry out the learning transfer, the dataset containing the two classes of insects was prepared. A tensor level augmentation method is proposed that will be applied so that a balanced number of images can be obtained for each class. Higher levels are blocked in the training process to maintain good generalization ability of the model acquired from training on the ImageNet set. The output levels, with the role of classification, are eliminated, being introduced new layers correspond to the classification of the two species of insects.

C. Hardware Used

For the DS2 test dataset, image acquisition was performed with a 16.1MP Nikon Coolpix P510 camera and a 12MP dual-camera Samsung Galaxy Note9 phone. For the DS3 test set, a DJI Mavic2 Pro drone equipped with a 20MP Hasselblad L1D-20c camera was used.

Image processing and dataset creation were performed using a Dell Precision Portable Graphics Station with an I7-9750H processor. Programming and training of the neural networks were performed using two Clevo portable graphics stations with custom configurations (Table II).

TABLE II. HARDWARE USED FOR CNN IMPLEMENTATION

Portable graphics station	Module	Characteristics
Dell Precision M7540	CPU	I7-9750H@2.6GHz-4.5GHz Intel UHD Graphics 630
	GPU	Nvidia Quadro T2000
Clevo 15 (P15)	CPU	I7-9750H@2.6GHz-4.5GHz Intel UHD Graphics 630
	GPU	Nvidia GTX 1660Ti

D. Software Used

The software content is made using MS Windows 10 Pro and Ubuntu 20.4 operating systems. NVidia research development drivers and libraries are used for graphics and tensor processing. The programming language in which the code necessary for each stage of the work development and testing is written in Python, and the integrated development environment, PyCharm. The development platforms and their related APIs used to train the networks are contained by TensorFlow and Keras. All frameworks (wrappers) and libraries used for developing applications and training networks are written specifically for the Python programming language.

E. Proposed System

After the tests were carried out, it was found that for images at a high resolution but with a low aspect ratio of the object of interest, as in the case of the detection and identification of harmful insects from trees in orchards, many false-negative and false-positive detections are obtained. This happens because of the positioning of the insects in the trees: on the leaves, on the fruits, on the branches, in the shade or light, at a smaller or larger distance from the camera. To reduce false negative errors, it is proposed to generate a pyramidal window capable of extracting regions by applying them to scale. To make time and data flow more efficient, the image processing will be tensorial, so that the segmented regions of interest will be transmitted directly to the detector, without being exported. Fig. 5 illustrates the operating principle of the sliding window pyramid. For each uploaded file, up to a maximum of 972 (constructively imposed) regions extracted from the original image are generated, depending on the desired detection accuracy. This is done by generating a window of variable size that sweeps over the source image with a defined step at the beginning and at the end of the stroke which changes the scale factor.

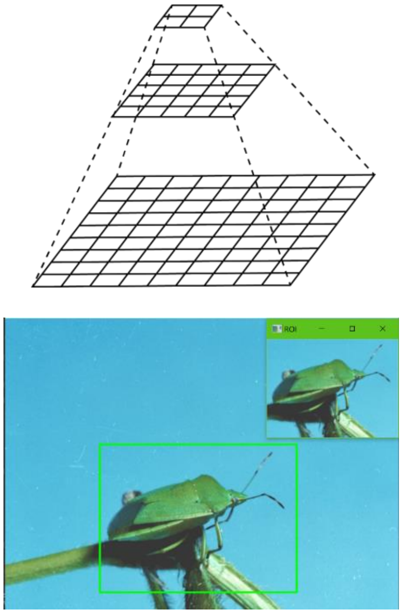


Fig. 5. The pyramidal ROI segmentation mechanism.

A mechanism for combining the two methods in an expert system capable of detecting the two classes of insects with good accuracy is shown in Fig. 6. After performing the detection using the trained YOLO v4-tiny model for the two classes of insects, HH and NV, the source images are marked and saved separately. The detected objects from the images are segmented so that later, in the case of a positive detection, they are analyzed using the EfficientNet-B3 classifier, and otherwise, for null detections from the source image, it is sent to the pyramid segmentation mechanism to extract new regions of smaller size to be reanalyzed by means of the Scaled-YOLO v4-tiny detector. Another advantage of using this mechanism is provided by segmenting the original image into sub-regions, in this way no details are lost as happens when scaling the original image to match the resolution of the input layer of the detector. Through this process, the aim is to reduce erroneous results, but also to improve the detection score and implicitly the precision.

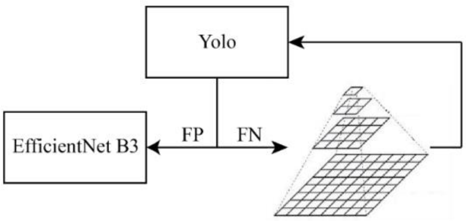


Fig. 6. The principle of operation of the proposed mechanism.

The images processed with the detector based on YOLOv4 are analyzed according to the detection result: for the presence of insects, they are classified using the EfficientNet-based network for evaluation, and for non-detections or results that do not meet the imposed parameters, the mechanism of pyramidal segmentation. The scheme describing the operation mode of the intelligent system for insect detection and the steps taken to validate and optimize the results is presented in Fig. 7.

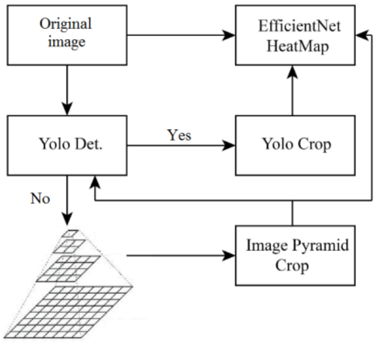


Fig. 7. The principle of operation of the proposed mechanism.

F. Result Indicators

Several metrics such as specificity, sensitivity, and accuracy were considered for the evaluation of convolutional neural networks (Table III). The meaning of the notations are as follows: TP - true positive cases, FN - false negative cases, TN - true negative cases, FP - false positive cases.

TABLE III. METRICS USED FOR CNN EVALUATION

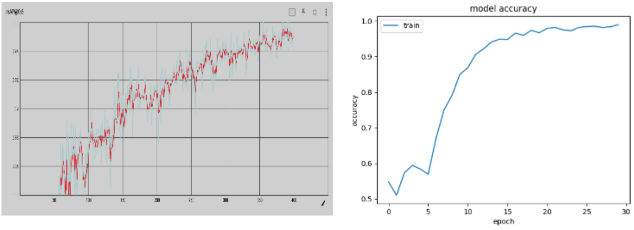
Name	Abbreviation	Formula
Recall (True Positive Rate)	*TPR*	$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$
Specificity (True Negative Rate)	*TNR*	$TNR = \frac{TN}{TN + FP}$
Precision (Positive Predictive Value)	*PPV*	$PPV = \frac{TP}{TP + FP}$
Accuracy	*ACC*	$ACC = \frac{TP + TN}{TP + TN + FP + FN}$
F1-score	*F1*	$F1 = \frac{2TP}{2TP + FP + FN}$

III. EXPERIMENTAL RESULTS

Following the tests performed on the DS 2 and DS 3 data sets, it was observed that the detection accuracy is strongly influenced by the sharpness of the acquired images, the correct focus of the regions of interest, as well as the object/image ratio. In the case of preprocessing the test image and fragmenting it into several sub-images (patches) the detection rate was improved as the proportion of the region of interest to the background plays an important role in the network's decision. Also, in the case of false-positive (FP) detections for NV, simultaneously with HH true-positive (TP), the score for HH was always higher.

For the Scaled-YOLO v4 custom model, the best model has been obtained for the training scenario of 400 iterations with the SAM SGD (Sharpness-Aware Minimization Stochastic Gradient Descent) optimization algorithm (Fig. 8.a). For the classifier based on the EfficientNet-B3 model, after training for 30 iterations, the accuracy curve in Fig.8.b was obtained.

Table IV shows the calculated result indicators for the validation data set. The data input resolution is 300x300 pixels, corresponding to the input level of the EfficientNet-B3 architecture. The number of images was 200: 100 for HH and 100 for NV. Similarly, in Table V, the result indicators were calculated for the Scaled-YOLO v4-tiny model, optimized with the SAM SGD algorithm.



(a) (b)

Fig. 8. Performances of the networks obtained during training. (a) Scaled-YOLOv.4: mAP plot for iterations 50–400 (b) EfficientNet B3: accuracy graph.

In Table VI, the result indicators were calculated for the combined Scaled-YOLO v4-tiny model with EfficientNet-B3 and pyramidal ROI segmentation mechanism.

TABLE IV. TEST RESULTS FOR EFFICIENTNET-B3 MODEL

CLASS	PPV	TPR	F1	Images
0 (HH)	0.92	0.86	0.88	100
1 (NV)	0.91	0.93	0.92	100
ACC			0.905	

TABLE V. TEST RESULTS FOR SCALED-YOLO V4-TINY

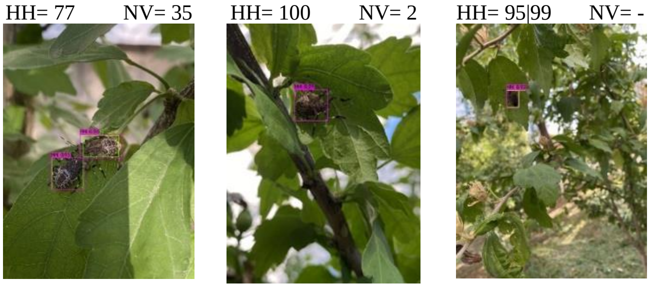
CLASS	PPV	TPR	F1	Images
0 (HH)	0.64	0.75	0.69	100
1 (NV)	0.81	1	0.89	100
ACC			0.79	

TABLE VI. TEST RESULTS FOR COMBINED STRUCTURE

CLASS	PPV	TPR	F1	Images
0 (HH)	0.94	0.90	0.92	100
1 (NV)	0.84	1	0.92	100
ACC			0.95	

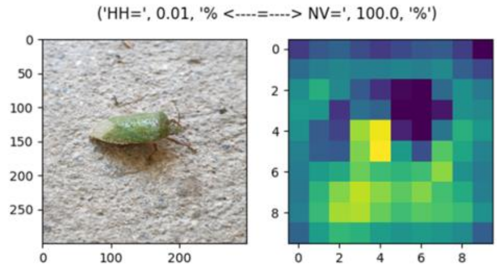
Some examples for the DS 2 test set are given in Fig. 9, for one (cases HH 77, HH 100, HH99, and HH 93) or two insects (cases HH95/99 and HH 95/96) in the image. Fig. 10 a) and b) present some results after applying the classification mechanism.

HH= 77 NV= 35 HH= 100 NV= 2 HH= 95|99 NV= -

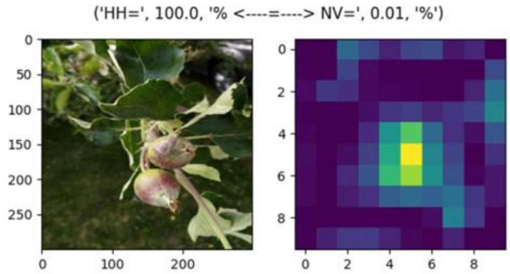


HH= 95|96 NV=13|2 HH= 99 NV= - HH= 93 NV= 12

Fig. 9. Examples of insect detection using Scaled-YOLO v4-Tiny (DS2).



(a)

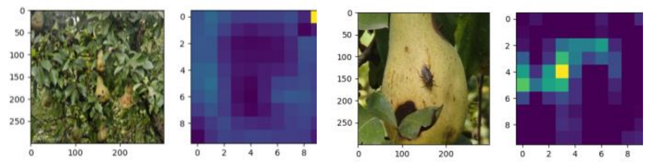


(b)

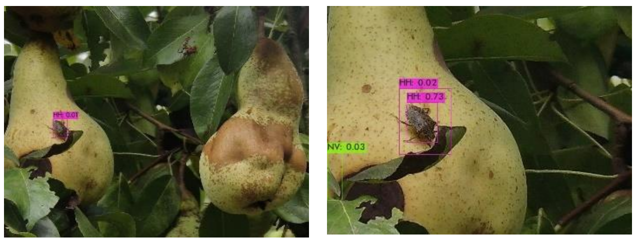
Fig. 10. Examples of insect classification using EfficientNet-B3 (DS1): (a) NV and (b) HH.

It was observed that most of the FP detections for the NV class in the DS 2 set were for HH. To reduce these FP detections, the proposed algorithm was applied, by reducing the tolerance threshold, partitioning the image into sub-images, and re-evaluation by returning the threshold to a higher value (Fig. 11).

(a) (b)



(c) (d)



(e) (f)

Fig. 11. Exemplification of the detection mechanism by the proposed method (DS2). (a) Source image - no object detected (FN), (b) NV= 2% FP detection, (c) Activation of the EfficientNet network after the split, (d) HH=1%, (e) HH=73%; HH=2%,NV=3% « 70%.

The test results were obtained under the following conditions: the detector will lower the tolerance threshold to 40% (it will exclude detections with scores < 40%) and on the next re-evaluation it will raise the tolerance threshold to 70% so that only images with a score of detection greater than 70% to be considered. For the classifier, the minimum threshold was imposed for all predictions, regardless of class, to be 50% (Table VII). Regarding the mechanism of dividing the initial image by sliding window, it was imposed that the displacement step of the window should not be less than 90% of its size, regardless of the scale in order not to repeat regions and introduce new disturbing factors into the system. It is assumed that segmenting the region of interest will be easier to identify.

TABLE VII. VALIDATION CRITERION FOR FUSED DECISION

Criterion	**Rule**	**Result**
Prediction score >50%	HH=HH/NV=NV	VALID
Prediction score <50% (for both classes)	HH≠HH/NV NV≠NV/HH	INVALID FP
Prediction score >50% (for opposite class)	HH=NV/NV=HH	FP (class err.)

IV. CONCLUSION

Through its flexibility, the proposed model can serve as detection, classification, monitoring, and evaluation tool for classes of insect pests in organic agriculture. The proposed solution can reduce false-positive (FP) or false-negative results. The empirical character of the realized deep convolutional neural network models preserves a margin of continuous improvement of the parameters, and through the obtained results contributes to the composition of both a theoretical and practical basis for the development of an expert system for the detection of harmful insects for extended areas and in - a short time interval.

In the future, it is proposed to develop a two-stage model, which uses a voting mechanism based on weights, and real-time data analysis for each subsystem of the proposed detector.

ACKNOWLEDGMENT

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] D. H. Lee, "Current status of research progress on the biology and management of Halyomorpha halys (Hemiptera: Pentatomidae) as an invasive species," Appl Entomol Zool, vol. 50, no. 3, pp. 277–290, Aug. 2015.

[2] Y. Li, H. Wang, L. M. Dang, A. Sadeghi-Niaraki, and H. Moon, "Crop pest recognition in natural scenes using convolutional neural networks," Comput Electron Agric, vol. 169, Feb. 2020,

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," http://arxiv.org/abs/1409.1556, 2014.

[4] W. P. Amorim, E. C. Tetila, H. Pistori, and J. P. Papa, "Semi-supervised learning with convolutional neural networks for UAV images automatic recognition," Computers and Electronics in Agriculture, vol. 164, 104932, Sep. 2019.

[5] E. C. Tetila, B. B. Machado, G. Astolfi, N. A. de Souza Belete, W. P. Amorim, A. R. Roel, and H. Pistori, "Detection and classification of soybean pests using deep learning with UAV images," Computers and Electronics in Agriculture, vol. 179, 105836, Dec. 2020.

[6] E. Ayan, H. Erbay, and F. Varçın, "Crop pest classification with a genetic algorithm-based weighted ensemble of deep convolutional neural networks," Computers and Electronics in Agriculture, vol. 179, 105809, Dec. 2020.

[7] H. Takimoto, Y. Sato, A. J. Nagano, K. K. Shimizu, and A. Kanagawa, "Using a two-stage convolutional neural network to rapidly identify tiny herbivorous beetles in the field", Ecological Informatics, Vol. 66, 101466, Dec. 2021.

[8] ITIS, Integrated Taxonomic Information System, "Report: Halyomorpha halys." https://www.itis.gov/servlet/SingleRpt/SingleRpt?search_topic=TSN&search_value=915660#null (accessed May 11, 2022).

[9] Maryland Biodiversity Project, "Brown Marmorated Stink Bug (Halyomorpha halys)." https://marylandbiodiversity.com/view/6416 (accessed May 11, 2022).

[10] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection." https://arxiv.org/abs/2004.10934?sid=NDAqzT, Apr. 2020.

[11] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-YOLOv4: Scaling Cross Stage Partial Network," Nov. 2020, [Online]. Available: http://arxiv.org/abs/2011.08036.

[12] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," http://arxiv.org/abs/1905.11946, May 2019.