2022 30th Mediterranean Conference on Control and Automation (MED)
June 28 - July 1, 2022. Athens, Greece

2022 30th Mediterranean Conference on Control and Automation (MED) | 978-1-6654-0673-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/MED54222.2022.9837254

Comparative Study of Neural Networks Used in Halyomorpha Halys Detection*

Dan Popescu, *Member, IEEE*, Loretta Ichim, *Member, IEEE*, Mihai Dimoiu, and Raluca Trufelea,

Abstract— **The paper's purpose was to investigate some methods based on neural networks for the detection and classification of harmful insects for agriculture as the Halyomorpha Halys. The implementation of different object detection networks for image categorization was analyzed. Images from the Maryland Biodiversity database were used for neural network training and testing. Rotation, scaling, blurring, mirroring, and other techniques were employed for data augmentation. For the detection and classification of Halyomorpha Halys, some neural networks that include multiple smaller networks were implemented and investigated. The networks used are the following: YOLOv5s, SSD with different backbones such as MobileNet V1, MobileNet V2, and ResNet-50, Faster R-CNN with ResNet-50 backbone, and EfficientDet-D0. Moreover, neural networks were evaluated and compared based on performance metrics such as accuracy and time. Performances like accuracy between 0.49 – 0.86 and time between 36 ms – 55 ms were obtained. The best results were obtained for YOLOv5s, in terms of accuracy, and EfficientDet-D0, in terms of time.**

I. Introduction

Precision agriculture now employs a combination of techniques to identify pests and extend the life of trees. Identification and object detection can assist us in extracting important information about the number of species present in a picture while also reducing the manual labor of professionals. Now experts can concentrate on preventing the spread of insects and illness. The degree of individual tree infection can predict the insect population of an orchard and pest management programs can be triggered [1]. Manual insect identification is typically slow, inaccurate, tiring and prone to mistakes, which prevents large-scale use. The use of automated insect monitoring allows different degrees of tree diseases to be identified. Manual specimen identification takes a long time and requires the knowledge of a professional. Automation is essential for lowering costs and increasing production. To accurately identify insects of interest in digital images, computer vision and machine learning approaches are increasingly used.

Diseases and pests that spread quickly, on the other hand, cause crop loss and lower farmer revenue. Despite the knowledge that excessive pesticide use harms the ecosystem, farmers use large dosages of chemical pesticides throughout the crop growing season to minimize bug infestations and reduce crop loss from diseases and pests.

Various features can be extracted from images to recognize insects such as the structure of the wings, color histogram, and the texture of the insect [2]. These features can be applied to a deep learning algorithm to learn the pattern of the insects. Also, authors in [3] noticed that deep learning in insect detection is unexplored. Deep learning has been widely employed in agricultural applications, including farmland mapping, crop image segmentation, and insect detection. Deep learning is a sophisticated machine learning approach that may be used to tackle a multitude of activities in image processing, remote sensing, and computer vision. In terms of deep learning, there are two types of neural networks: supervised and unsupervised neural networks. For image segmentation and data restoration, a supervised neural network is utilized. For the detection module to be trained, supervised classification involves learning information about the study region. The multispectral images acquired from sensors are subjected to unsupervised change detection systems. Preprocessing, semantic segmentation, and postprocessing are the three primary categories of these techniques [4].

A new neural network called PestNet [5] was used for multi-class pest detection. It combines both localization and classification and is significantly more complex than the neural networks for generic object detection. PestNet is organized into three primary components. The first, a new module for feature extraction and improvement, called channel-spatial attention (CSA) is designed to be fused into the convolutional neural network (CNN) backbone. The second is the region proposal network (RPN), which is used to provide region proposals as probable pest spots using feature maps derived from images. The third component, the position-sensitive score map (PSSM), is leveraged to replace fully connected (FC) layers for pest detection and bounding box regression.

This paper's purpose was to investigate some methods based on neural networks for the detection and classification of harmful insects for agriculture as the Halyomorpha Halys (HH). The neural networks that include multiple smaller networks were implemented and investigated. The networks used are the following: YOLO v5s, SSD with different backbones such as MobileNet V1, MobileNet V2, and ResNet-

* This work was supported by a grant from the Ministry of Research, Innovation and Digitization, CNCS/CCCDI – UEFISCDI, project number 202/2020, within PNCDI III.

D. Popescu is with the Faculty of Automatic Control and Computer Science, University Politehnica of Bucharest, 060042 Bucharest, Romania (phone: +40766218363; dan.popescu@upb.ro).

L. Ichim, M. Dimoiu, and R. Trufelea are with the Faculty of Automatic Control and Computer Science, University Politehnica of Bucharest, 060042 Bucharest, Romania (loretta.ichim@upb.ro, mihaidimoiu@yahoo.com, and trufelearaluca@yahoo.com).

50, Faster R-CNN with ResNet-50 backbone, and EfficientDet-D0.

The paper is organized as follows: Section I discusses related work concerning about application of neural networks used in precision agriculture; Section II proposes the analysis of neural networks as well as the approach to image processing with insects while Section III analyzes and discusses the obtained results and comparisons between methods. Finally, Section IV concludes the paper and outlines future directions.

II. Materials and Methods

A. Dataset used

The images used both for the learning and testing phases are collected from the Maryland Biodiversity Project database [6]. Some examples of images selected from the Maryland database are presented in Figure 1.

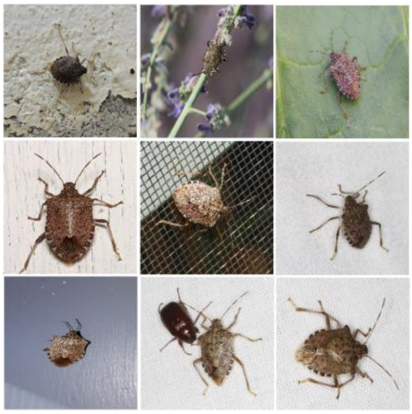


Figure 1. Samples from the database used.

The database has over 11,000 species, including about 3,700 species, and the work of more than 200 photographers and naturalists. For the CNN training, we manually labeled the data with the aid of the Computer Vision Annotation Tool (CVAT) created by OpenCV [7]. The application is free and allows us to export labels in a variety of formats. First, they are exported in YOLO 1.1 format, and then we made minor changes to match the YOLO v5 structure. Some annotation best practices [8] for object detection can be to fit the complete items and the boxes should be as narrow as feasible. The model's ability to learn is limited if any items are missing. Also, we used some best practices for data collection [9]. It is usually preferable for the model to view the item in as many different contexts as possible since this allows it to learn more effectively.

Rotation, scaling, blurring, mirroring, and other techniques were employed for data augmentation concerning HH images.

A. Neural Networks Used

The authors in [10] used a single neural network (YOLO) to process the entire picture. The image is divided into regions by the network, which predicts bounding boxes. Anticipated probabilities are used to weigh these bounding boxes. At the test time, it examines the entire image, thus its predictions are influenced by the image's overall context. In contrast to systems like R-CNN, which need thousands of network evaluations for a single image, it provides predictions with just one.

Another important neural network used for image segmentation is EfficientDet [11] launched by Google. In computer vision, model efficiency is becoming indispensable. The authors of [12] investigated neural network architecture design choices for object identification in-depth and proposed many significant enhancements to boost performance. They offer a weighted bi-directional feature pyramid network (BiFPN) for easy and quick multiscale feature fusion, and a compound scaling technique that scales the resolution, depth, and width of all backbone, feature network, and box/class prediction networks at the same time. They introduced a new family of object detectors called EfficientDet based on these improvements and enhanced backbones.

The first CNN considered in our study was investigated YOLO v5s, developed in the Ultralytics PyTorch framework, which is easy to use and makes quick inferences. Because the model has learned to detect items in the early levels, we will just retrain the subsequent layers to understand what distinguishes sunglasses from other objects. We aim to transfer as much information as possible from the previous task the model was trained on to the new task at hand in transfer learning.

We used YOLO v5s which is the small version of YOLO v5. YOLO v5s (Figure 2) unifies what was formerly a multi-step process by performing both classification and prediction of bounding boxes for identified objects using a single neural network. As a result, it is extensively tuned for detection efficiency and can identify and categorize objects considerably quicker than two independent neural networks. It does this by repurposing standard image classifiers for the regression job of determining object bounding boxes. Because YOLO v5 is a single-stage object detector it contains three key components: backbone, neck, and head. This model summary is constructed from 191 layers. This model has 7.3M parameters. We train for 100 epochs with a batch size of 16, an image size of 416×416, and a batch size of 16.

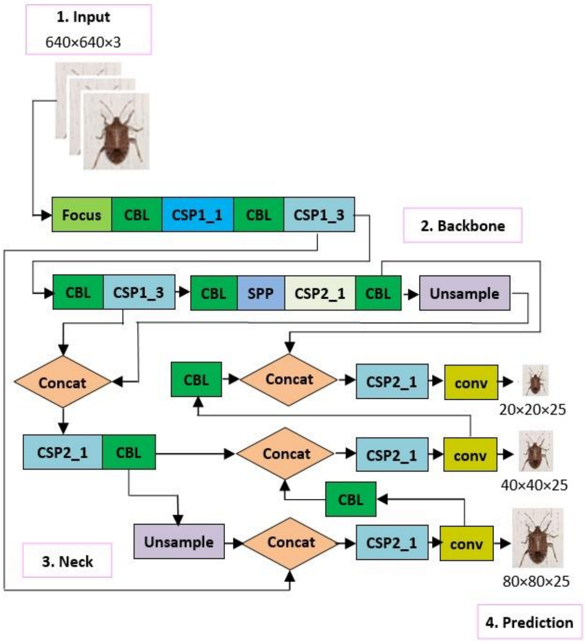


Figure 2. YOLO v5s architecture.

The core objective of Backbone is to extract important features from an input image. The CSPNet (Cross Stage Partial Networks) backbone [13] is utilized in YOLO v5s to extract rich important features from an input image. The main purpose of Neck is to create feature pyramids. Feature pyramids help in the generalization of models on an objective scale. It aids in the identification of the same object in various sizes and scales. PANet (Path Aggregation Network) is applied as the neck in YOLO v5s to get feature pyramids. The middle/hidden layers employ the Leaky ReLU activation function, whereas the final detection layer uses the Sigmoid activation function.

We trained the network for 100 epochs using transfer learning and achieved an mAP @ 0.5 IoU of 0.8689 (Figure 3). The mAP is the area under the precision-recall curve, and it represents a measure of quality across all recall levels for single class categorization. The average prediction time is 55 milliseconds. The mAP is calculated by averaging the AP for each class. The Mean Average Precision, or mAP score, is computed by averaging the AP across all courses and/or the total IoU threshold. mAP @ 0.5 IoU means the metric value mean Average Precision at Intersection over Union with a threshold of 0.5.

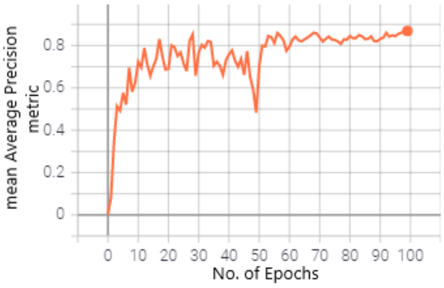


Figure 3. mAP@0.5.

The loss value is 0.01 after 100 epochs (Figure 4).

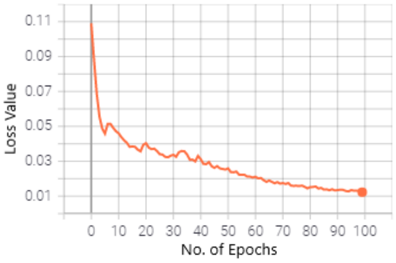


Figure 4. Loss value in the training session.

For the second network, we chose MobileNetV1. It is based on a simplified architecture that builds lightweight deep neural networks with minimal latency for mobile and embedded devices using depthwise separable convolutions. MobileNet is a CNN architecture (Figure 5) that is both efficient and portable, and it is employed in real-world applications [14]. To develop lighter models, MobileNets typically employ depthwise separable convolutions (DSConv) instead of the typical convolutions used in previous designs. Convolution layers that are depthwise separable are used to implement MobileNets. A depthwise convolution (DConv) and a pointwise convolution (PConv) make up each depthwise separable convolution layer. The net contains 28 layers if DConv and PConv are counted separately. The width multiplier hyperparameter can be adjusted to reduce the number of parameters in a conventional MobileNetV1 to 4.2 million. One DSConv block is composed of DConv followed by PConv.

A newer, similar net, MobileNetV2 architecture [15] has two types of convolutional blocks called Bottleneck residual blocks, which are presented in Figure 6. There are three layers for both blocks. The first layer is 1x1 convolution with activation function as ReLU6. The second layer is depthwise convolution and the third layer is another 1x1 convolution, this time simple, without linearity.

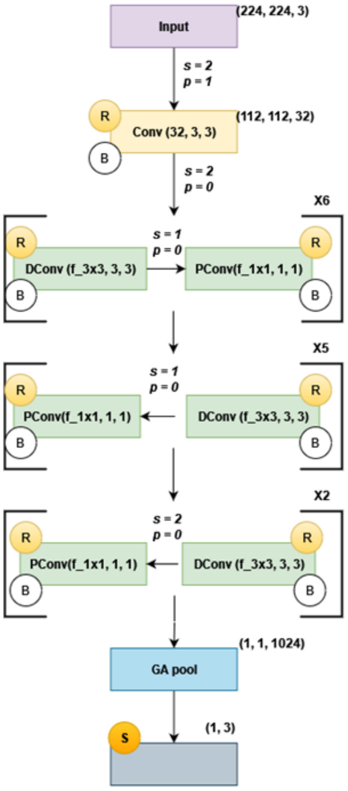


Figure 5. MobileNetV1 Architecture (DConv - Depthwise Convolution, PConv - pointwise convolution, R - ReLu, B – Batch Normalization, S – Softmax layer, s - stride, p - padding).

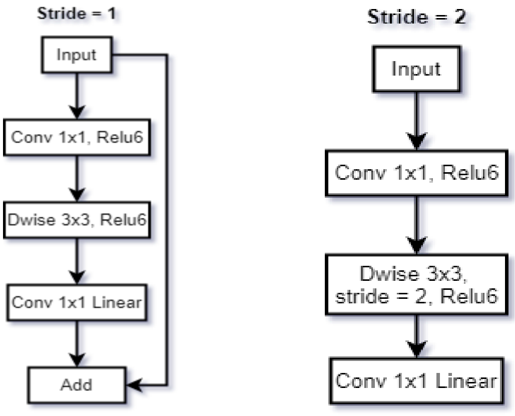


Figure 6. MobileNet V2 Blocks.

The third network we tested is SSD. It is a single-shot detector and is intended for real-time object detection. Faster-

RCNN creates boundary boxes using a region proposal network and then uses those boxes to categorize items. By removing the requirement for the region proposal network, SSD speeds up the procedure. SSD implements a number of enhancements, including multi-scale features and default boxes, to overcompensate for the decreased accuracy. These enhancements allow SSD to match the accuracy of the Faster R-CNN using lower quality pictures and increasing the speed. The average precision was 72.3% @ 0.5 IoU and 64% @ 0.75 IoU. After 3000 steps of transfer learning, our loss was minimized to 0.08. We used a batch size of 8.

The fourth network was EfficientDet [16] created by Google Brain. This neural network uses EfficientNet pre-trained architecture as a backbone (Figure 7). We implemented the small-size EfficientDet-D0 using as the backbone the EfficientNet for feature extraction and the BiFPN (Bi-directional feature network) for feature fusion (Figure 8).

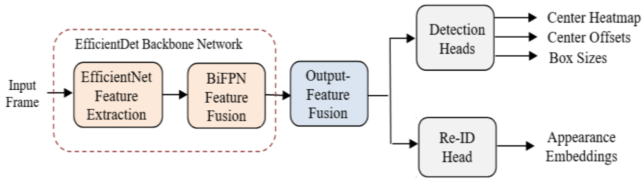


Figure 7. EfficientDet Architecture [16'].

EfficientDet is a neural network architecture for object detection designed to increase model efficiency. Across a wide range of resource limitations, this design is significantly more efficient than traditional design.

We applied transfer learning to this model with our dataset and achieved a mean Average Precision @ 0.5 Intersection over Union of 0.63. The Average Precision @ 0.75 is 0.44. The model is trained on 512×512 image resolution.

The fifth network is Faster R-CNN with ResNet50 as the backbone model (Figure 8). By processing the input images with convolutional and max-pooling layers, the Fast R-CNN architecture generates a convolutional feature map [17]. The feature maps are used by the region proposal network to predict a rectangular object with a score (probability). Input images with a resolution of 640×640/ RGB pixels are used in the actual implementation of the model. In place of the original ZF-NET and VGG-NET, a 50-layer ResNet was employed as the backbone (which used to be pretrained on ImageNet). ResNet has the benefit over VGG in that it is larger, implying that it has a greater ability to learn what is required. ResNet also takes advantage of residual connections and batch normalization, both of which were not available when VGG originally came out.

The Faster R-CNN's fine-tuning and training convolutional neural networks have been shown to be effective visual models that can conduct precise insect counting in images. These techniques treat an image as a pixel matrix with a size (kernel) of (height-width-depth), where depth is the number of image channels (3 for RGB crop images).

For the sixth network, we chose SSD with ResNet50 [19] as a backbone for extracting features. We used SSD as the fundamental network structure and replace the VGG16 network on the inside with a ResNet50 network. The ResNet network is good at recognizing a variety of objects in relatively small data sets, according to the outcomes of the experiments. In terms of learning efficiency and accuracy, the proposed model surpasses existing neural networks. An SSD is made up of two parts: a backbone model and an SSD head. As a feature extractor, the backbone model is mainly a pre-trained image classification network. This is the ResNet50 network trained on ImageNet that has had the last fully linked classification layer removed. As a result, we have a deep neural network that can extract semantic meaning from an input image while keeping the image's spatial structure, although at a poorer resolution.

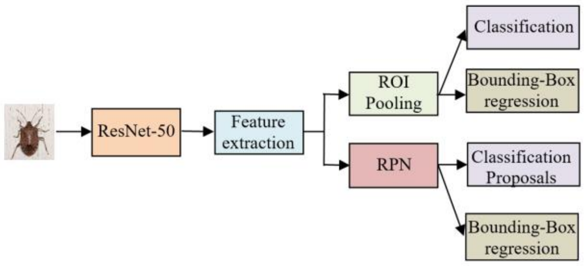


Figure 8. Faster R-CNN with ResNet50 Backbone Architecture (adapted from [18]).

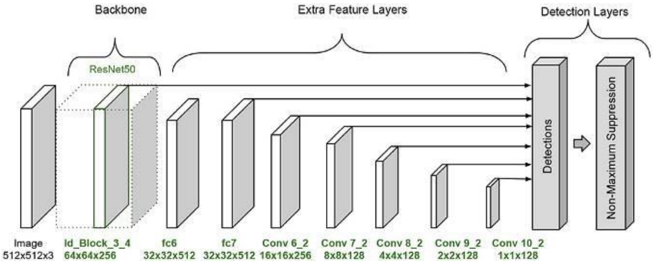


Figure 9. SSD with ResNet50 Backbone Architecture [20].

III. EXPERIMENTAL RESULTS

There are a total of 562 images in the created dataset which was divided into three parts: 70% training images (meaning 406 images), 10% validation images (meaning 45 images), and 20% test images (meaning 111 images). We used Amazon Web Services to set up a Virtual Machine and deploy this software. Because CVAT is based on Docker, we chose NginX as the webserver and reverse proxy. NginX is a reverse proxy, load balancer, mail proxy, and HTTP cache that can be used as a web server. Predictions for the Yolo network using batch images from the testing dataset are shown in Figure 10. Examples of predictions of HH in different contexts for the six compared networks are presented in Figure 11.

As it can see the error of detection and classification is also influenced by the context (background). If it is more complex, there is a higher probability of error. Thus, the EfficientDet and Faster R-CNN with ResNet50 Backbone are wrong in the classification for the image on the left (the cases g and l)), but they frame and classify the image on the right correctly. The detection confidence also takes higher values for the image on the right than the image on the left. The accuracy and testing (operating) time are presented in Table 1 for all the proposed networks. YOLOv5s, SSD with different backbones (such as MobileNet V1, MobileNet V2, and ResNet-50), Faster R-

CNN with ResNet-50 backbone, and EfficientDet-D0. Performances like accuracy between 0.49 – 0.86 and time between 36 ms – 55 ms were obtained. The best results are obtained for YOLOv5s as accuracy (mAP@0.5IoU = 0.86) and for EfficientDet as operating time (36 ms).

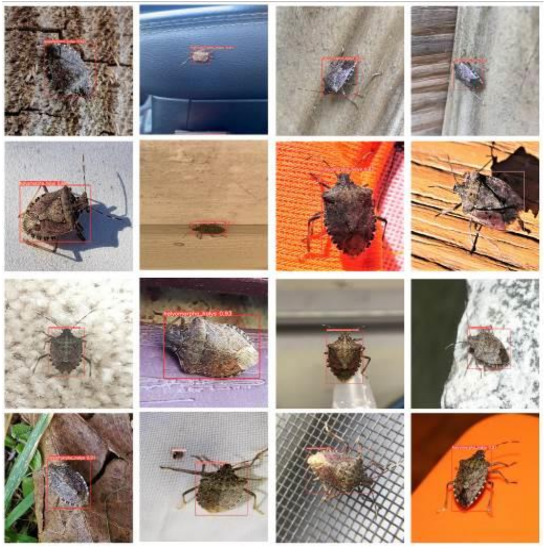
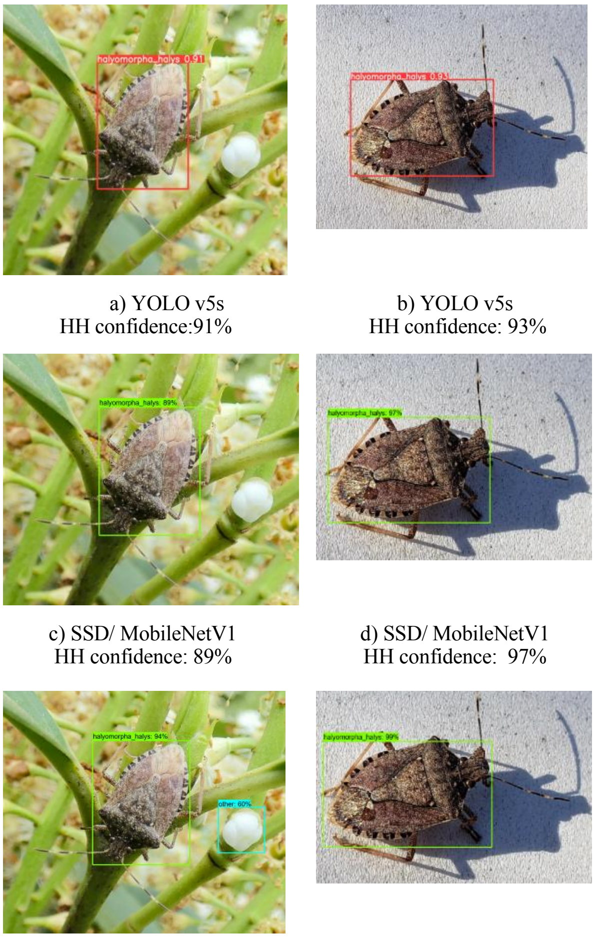


Figure 10. Predictions from YOLO v5s Test Dataset.



a) YOLO v5s
HH confidence:91%

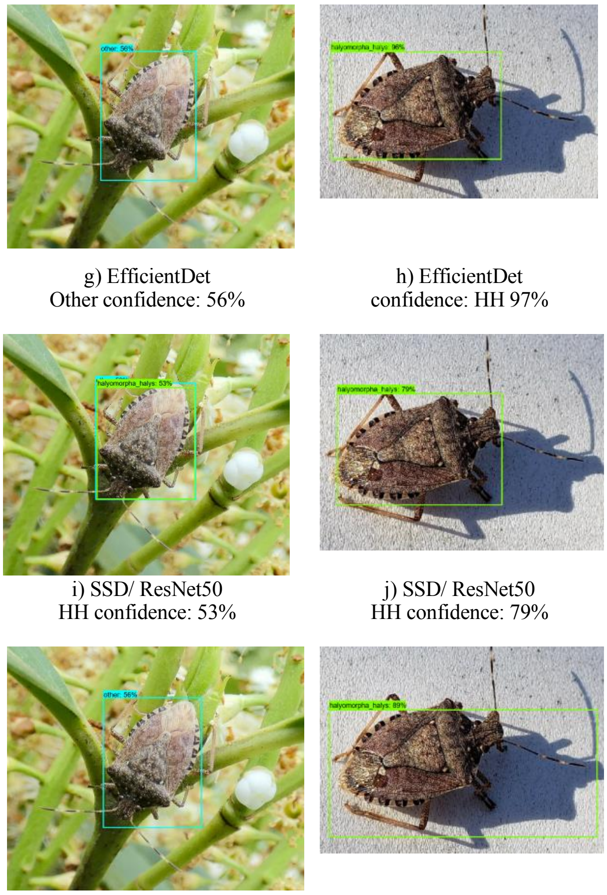
b) YOLO v5s
HH confidence: 93%

c) SSD/ MobileNetV1
HH confidence: 89%

d) SSD/ MobileNetV1
HH confidence: 97%

e) SSD/ MobileNetV2
HH confidence: 94%

f) SSD/ MobileNetV2
HH confidence: 99%



g) EfficientDet
Other confidence: 56%

h) EfficientDet
confidence: HH 97%

i) SSD/ ResNet50
HH confidence: 53%

j) SSD/ ResNet50
HH confidence: 79%

l) Faster R-CNN
Other confidence: 56%

m) Faster R-CNN
HH confidence: 89%

Figure 11. Example of test dataset predictions of HH in different contexts for the six compared networks.

If we consider a majority vote on the decisions of the networks, we notice that the error in Figure 11 disappears: for the image on the left the voting score is 4/6, and for the image on the right the voting score is 6/6 so that in each of the two images detects HH.

TABLE I. NETWORK COMPARISON FOR ACCURACY AND SPEED

Model	mAP@0.5IoU	mAP@0.75IoU	Time (ms)
YOLOv5s	0.86	0.68	55
SSD/ MobileNet V1	0.72	0.68	48
SSD/ MobileNet V2	0.72	0.64	39
EfficientDet-D0	0.63	0.44	36
SSD/ ResNet50	0.59	0.49	46
Faster R-CNN/ ResNet50	0.49	0.29	53

IV. CONCLUSIONS

No relevant publications in the usage and detection of the Halyomorpha Halys insect were identified, according to the authors. We tested and evaluated many networks to discover which ones were the most effective at recognizing and categorizing this species. Deep learning algorithms have enabled us to build new image-based applications that would be impossible to execute using traditional image processing approaches. The potential benefits of CNN are encouraging for their further application in smarter, more sustainable farming and food supply. In terms of various components, this system could be enhanced or modified. More current CNN designs, for example, may be pushed to the limits, including fine-tuning the CNN through a retraining process to make it more color sensitive.

We will introduce new onsite shots to this dataset and retrain the network and it will be deployed on a UAV to generate real-time forecasts in next-gen farms. We conducted multiple pieces of training on several neural networks in order to determine which one is optimal for real-time prediction on a UAV. Because it blends precision and response time so effectively, we believe YOLOv5s is the ideal choice for transfer learning and deployment on embedded systems in UAVs for analyzing real-time images from next-generation precision agriculture. In future work, we intend to improve the methodology by a weighted fusion of decisions of the best classifiers based on neural networks.

References

[1] L. C. Wright and W. W. Cone, "Population dynamics of Brachycorynella asparagi (Homoptera: Aphididae) on undisturbed asparagus in Washington State," *Environ. Entomol.,* vol. 17, no. 5, pp. 878–886, 1988.

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[3] A. Nazri, N. Mazlan, and F. Muharam, "PENYEK: Automated brown planthopper detection from imperfect sticky pad images using deep convolutional neural network," PLoS ONE, vol. 13(12):e0208501, Dec. 2018.

[4] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2094–2107, 2014.

[5] L. Liu, R. Wang, C. Xie, P. Yang, F. Wang, S. Sudirman, and W. Liu, "PestNet: An end-to-end deep learning approach for large-scale multi-class pest detection and classification," *IEEE Journal*, pp. 45301–45312, 2019.

[6] Halyomorpha Halys https://www.marylandbiodiversity.com/view/6416 (Accessed June 2021).

[7] https://github.com/openvinotoolkit/cvat (Accessed June 2021).

[8] Annotation Best Practices (Accessed June 2021 - https://nanonets.github.io/tutorials-page/docs/annotate.

[9] Data Collection Best Practices (Accessed June 2021 - https://nanonets.github.io/tutorials-page/docs/data-collection).

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," https://arxiv.org/abs/1506.02640, May 2016.

[11] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," https://arxiv.org/abs/1911.09070v7, Jul. 2020.

[12] E. S. Olivas, J. D. M. Guerrero, M. Martinez-Sober, J. R. Magdalena-Benedito, and A. J. Serrano López, *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods and Techniques*, Information Science Reference, 2009.

[13] C. -Y. Wang, H. -Y. Mark Liao, Y. -H. Wu, P. -Y. Chen, J. -W. Hsieh, and I. -H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 1571–1580.

[14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," https://arxiv.org/abs/1704.04861, Apr. 2017.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," https://arxiv.org/abs/1801.04381, Mar. 2019.

[16] Y. Lee, S.-H. Lee, J. Yoo, and S. Kwon, "Efficient single-shot multi-object tracking for vehicles in traffic scenarios," *Sensors*, vol. 21, 6358, 2021.

[17] S. Ren, K. He, R. Girshick, and S. Jian, "Faster R-CNN: Towards real-time object detection with region proposal networks," https://arxiv.org/abs/1506.01497, Jan. 2016.

[18] Y. Zhou, S. Wen, D. Wang, J. Mu, and I. Richard, "Object detection in autonomous driving scenarios based on an improved Faster-RCNN," *Appl. Sci.*, vol. 11, 11630, 2021.

[19] X. Lu, X. Kang, S. Nishide, and F. Ren, "Object detection based on SSD-ResNet," 2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS), 2019, pp. 89–92.

[20] F. Furlán, E. Rubio, H. Sossa, and V. Ponce "CNN based detectors on planetary environments: a performance evaluation. front neurorobot," vol. 14, 590371, Oct. 2020.