Detection of Halyomorpha Halys Using Neural Networks*

A. Sava, L. Ichim, *Member, IEEE,* and D. Popescu, *Member, IEEE*

***Abstract* — The paper's goal was to create some neural networks-based models for the detection and classification of insects such as Halyomorpha Halys in ecological orchards, from acquired images in the trees. The detecting operations were performed using models from two of the most efficient deep learning families in this area: R-CNN and YOLO. Using the proposed models, (Faster R-CNN, YOLOv5-s, YOLOv5-m, and YOLOv5-l) to early detection of harmful insects, a real contribution to anticipating damage in orchards is possible. The dataset is composed of images taken from the Maryland Biodiversity dataset. All training and testing operations were performed with the help of GPU processors provided by Google, the resulting models being saved on Google Drive Cloud. The images were evaluated from the detection and the classification perspective based on specific metrics such as precision, recall, and mAP. The best results were obtained for YOLOv5-m.**

I. Introduction

Insect monitoring is an essential action in maintaining and developing agricultural crops. The information about the presence and abundance of pests and taking appropriate action in a timely manner are necessary to reduce potentially damaging infestations. Pest monitoring is usually carried out by methods involving active human analysis, which takes time and financial resources for farmers as it requires direct inspection of crops. On the other hand, the use of traps or the use of chemicals to prevent possible invasions has no effect on many types of insects like Halyomorpha Halys (HH) and can cause negative effects on crops, leading to increased damage [1]. The paper's objective is to create an automatic method for locating and detecting HH insects from images received from UAVs by using an artificial intelligence model.

In recent years, machine learning and especially convolutional neural networks have proven to be a useful tool in insect monitoring, having promising results in multiple tasks such as detection, semantic segmentation, and classification. This type of network uses the idea of the artificial neuron, which simulates how neurons in the human brain function. Detecting objects of interest can be useful when it is used correctly, depending on the area in which it is used. Considering that, the expected results come from an optimal implementation of actual solutions according to the user needs being obtained with the help of efficient and stable architectures. The task of detecting and locating objects in images is a complex process whose behavior has been studied in various research conducted in this area.

HH is a member of the Hemiptera insect family, which contains about 82,000 species and is the largest hemimetabolous insect family. All these insects have similar anatomy, diversifying through a wide range of different sources of food. HH, also known as the Brown Marmorated Stink Bug, is native to Asia (China, Taiwan, Korea, and Japan) and has become a major global pest in recent decades, mainly due to its ability to colonize outside its native area [2]. The high spreading capacity (at least 170 plant species) and the ability to multiply with other endemic species have increased the number of insects in this family. For example, the damage caused by HH has increased considerably in recent years, resulting in significant losses, particularly in agricultural crops, orchards, grapes, ornamentals, vegetables, and seed crops [3]. As HH continues to expand, it becomes a growing threat to agriculture. HH is also an alarming pest well known for its invasion of human settlements such as houses, schools, and other indoor spaces, in large numbers, especially in winter.

Thanks to the advance in technology there are modern methods that could help better detect and observe the invasive insect species. UAVs equipped with cameras can be automatically driven on a designed route in the monitored area aiming to capture targeted images and gather crop data. It is not a new concept, as scientists have continuously searched for ways to improve the robots equipped with video cameras that can be used in all sorts of activities including agriculture. Although the agricultural sector implies a series of multiple factors to be considered when trying to take measurements, like wind speed, the ground type, the intensity of the light, the weather conditions, the crop placement, etc., the robots have been more and more often used in agricultural research because the information that they collected has helped the researchers to find solutions to several problems [4], [5].

Using UAVs for the acquisition of the entire crop area, with high resolution but also with low costs, makes the detection of HH insects an important area of analysis. However, the large amount of data that is required in the training and validation

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

A.Sava is with the University POLITEHNICA of Bucharest, Romania (e-mail: sava.andrei.octavian@gmail.com).

L. Ichim, is with the University POLITEHNICA of Bucharest, Romania (e-mail: loretta.ichim@upb.ro).

D. Popescu is with the University POLITEHNICA of Bucharest, Romania (corresponding author; e-mail: dan.popescu@upb.ro).

.

phases is a time-consuming process because manual labeling of data but is necessary for these phases.

The solution proposed in this paper for the task of HH detection is based on object recognition algorithms, including the R-CNN (Regions with Convolutional Neural Networks) series [6] and the YOLO (You Only Look Once) series [7]. The YOLO series is superior in terms of speed, being the better option in practical scenarios because R-CNN can't meet the real-time performance of object detection. The advantage of the R-CNN series is that it is superior in detecting target objects requiring higher accuracy.

II. Materials and Methods

A. Neural Networks Used

Models from two families of convolutional meural networks are used for HH detection: R-CNN (Regions with Convolutional Neural Networks) and YOLO (You Only Look Once).

Released in 2014, R-CNN [6] was one of the first networks to provide good results for locating, detecting, and segmenting objects in images. The results were obtained from the popular databases VOC-2012 [7] and ILSVRC-2013 [8]. To get accurate results, the authors composed an architecture with three modules:

(1) Module for selective search of region extraction.

(2) Module for feature extraction using machine learning techniques such as convolutional neural networks.

(3) Module for classification, marking the region of interest into bounding boxes.

The regions proposed for study are automatically chosen using an artificial vision technique (selective search). The choice of this algorithm can be modified due to the flexibility of the model. The neural network used to extract features is AlexNet [9]. AlexNet consists of a total of eight layers, of which five are convolutional layers, some with "max-pooling" layers, and the remaining are fully connected layers. Thus, two fully connected layers (FC) are used for selection, followed by a "softmax" layer at the end [10]. When apply the dataset, the parameters for AlexNet must have input values of the size 224 × 224 × 3. This network marks the beginning of the evolution of the image segmentation field, winning in 2012 the ImageNet Large-Scale Visual Recognition Challenge-ILSVRC competition with an accuracy of 84.6%. The output, in the case of R-CNN, is represented by a vector of 4,096 elements that describe the content of the image.

Fast R-CNN, representing the improvement of R-CNN by implementing new machine learning techniques and innovations to increase speed and accuracy, obtained 9 times faster speed in the training phase and 213 times faster speed in the testing phase compared to the first version of R-CNN. Moreover, comparing the network with SPPnet (Spatial Pyramid Pooling) [11], a network built on having, like R-CNN, the three main networks put into a multi-stage architecture, the training time is 3 times shorter and the test time 10 times shorter. The main difference between Fast R-CNN and R-CNN is the removal of the unnecessary CNN vectors for each proposed region and the addition of a single CNN vector, passed through the entire image, containing common features of all regions proposed for analysis [12].

For the Faster R-CNN architecture ([13], [14]) (Fig. 1), the authors chose to replace the selective search used in the first module of the network with a new algorithm for proposing regions of interest in input images called Region Proposal Network (RPN). The second module of this architecture is the Fast R-CNN detector, which locates and classifies target objects in images. The architecture starts by sending the input image from the dataset to a main convolutional network (called the "backbone"). Usually, the output of the backbone network has a much smaller H × W size than the input image. For each value on the output feature map, the model must learn if it has a matching object in the original image, at the appropriate size and location. The feature map is 40 × 60 for a 600 × 1000 image. This leads to a total of 21,600 anchor boxes proposed for analysis, 9 for each pixel in the feature map generated by the backbone. During training, all the bounding boxes that exceed the edge of the initial image are ignored. In the end, an average of 6000 bounding boxes results that contribute to calculating the network loss [13].

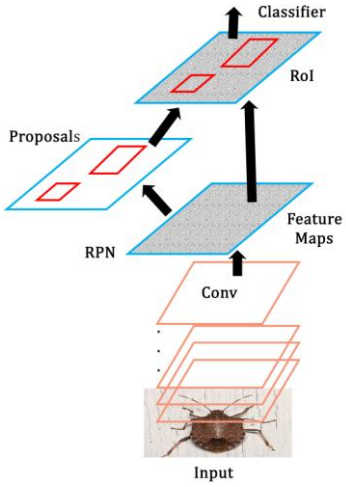

Figure 1. Faster R-CNN architecture.

Originally released in 2015, the You Only Look Once (YOLO) network quickly caught the attention of researchers in the field of computer vision because of the innovation brought by the authors in locating and detecting objects in images. R-CNN networks use Regions Proposal Networks (RPNs). Following the steps above, R-CNN networks apply a classifier to the regions proposed by the RPN network as belonging to the class (s), and then apply post-processing methods to reduce duplicates and refine the bounding boxes. The optimization problems due to the multiple networks required to obtain the location and classification in the R-CNN family motivated the authors to develop a single network, composed of all the previous stages. Having an input image that with or without

target objects, after a single convolutional neural network with multiple convolutional layers, the system produces vectors corresponding to each object that is detected in the image. The main improvement of YOLO is that all object characteristics and predictions are calculated simultaneously [15]. Shortly after the release of YOLOv4 [16], a new version was released by Ultralystics LLC [17]. They are known for implementing a YOLO architecture, written by A. Bochkovsky in a system that uses mostly C as a programming language, in PyTorch, one of the main libraries used in Python to develop artificial intelligence architectures. Thus, the main advantage of YOLOv5 [17] is the much easier integration with IoT devices due to the use of Python as a programming language. The main purpose of the YOLO architecture is to directly return the localization of the target objects and the class corresponding to each detection, starting from the initial image used as input to the neural network. YOLOv5 architecture is shown in Fig. 2.

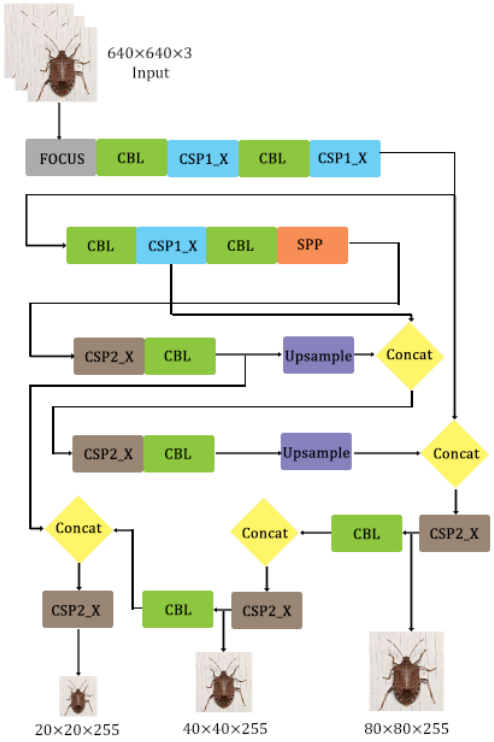

Figure 2. YOLOv5-m architecture.

The main YOLO architecture is composed of three parts [18]:

(1) Feature Extraction: Also called the backbone, it is a convolutional neural network that extracts the main features from the input images. The major change compared with YOLOv4 is that the module starts with a new structure, called Focus, which builds a feature map of size 320 × 320 × k, where k is chosen based on the model.

(2) Feature Aggregation: This module, called the neck, processes the backbone output, creating semantic features and, in other words, preserving features that contain important details that would be lost due to sequences of convolutional operations. YOLOv5 uses the FPN-PAN [19] architecture. The main idea of using this architecture is to generate pyramidal layers of features, improving the detection of objects at different scales and increasing the detection of the same object, regardless of scale and size. This structure also helps to propagate low-level features into future layers.

(3) Prediction: Also called the head, this part makes the prediction, generates the bounding box, and classifies the contents of the box into a category according to the class learned in the learning phase.

Transfer learning is commonly used in an optimization or classification problem when it is required to achieve high accuracy in a limited time frame. In this research on bug classification, transfer learning is suitable if it is used on a small dataset of images, in our case, a few hundred images. As deep learning requires a huge amount of data for training, transfer learning can use this pattern to train the pre-trained network on a small dataset. For the training and testing of artificial intelligence models in the field of artificial vision, the number of parameters is extremely high, so high-power data processing is needed. Common computers are an inefficient alternative in terms of training time. Thus, to obtain good results, it is necessary to have a platform on which to carry out the training and testing processes of the models. Google Colab is a web development system provided by Google to support the efforts of artificial intelligence researchers. The platform provides free access to the Graphics Processing Unit (GPU), but also the Tensor Processing Unit (TPU). Moreover, throughout the development process, Python was used as the core programming language. The first step was to preprocess the data for the detection and classification of HH insects in the dataset images. The data processing consists of preparing the images for training, respecting the requirements of the architecture used: creating XML files for the R-CNN series and TXT files for the YOLO series, each file having information about bounding boxes, all files being used for training and validating.

B. Dataset Used

A large dataset is required to successfully train the proposed neural networks. The images we have used for learning and validation have been taken from the Maryland Biodiversity dataset [20]. The total number of images in the dataset is 700. The dataset contains images of HH in different poses, from different distances, and at different stages of evolution. The images have different resolutions and are saved in JPG format. Examples of images can be seen in Fig. 3.

(a) (b) (c) (d)

Figure 3. Examples of HH: (a) and (b) adult; (c) and (d) nymph.

C. Evaluation Metrics

For the evaluation of the HH detection and classification models, metrics (statistic indicators) like precision, recall, and mean average precision are introduced.

Precision (P) is the number of instances detected correctly relative to the total number of instances detected by the algorithm. This index is calculated using the formula (1).

$$P = \frac{TP}{TP + FP} \tag{1}$$

Recall (R) represents the number of instances detected correctly in relation to the actual number of instances that had to be detected by the algorithm. This index is calculated using the formula (2).

$$R = \frac{TP}{TP + FN} \tag{2}$$

Average Precision (AP) is the area under the precision-recall curve (PR curve), where the X-axis is the recall and the y-axis is the precision. AP is a way to summarize the precision-recall curve into a single value representing the average of all precisions. This index is calculated using the formula (3). Mean Average Precision (mAP) is the average of AP and is calculated using the formula (3), which represents the AP of class and represents the number of classes.

$$AP = \int_0^1 P(R)dR \tag{3}$$

$$mAP = \frac{1}{n}\sum_{n=1}^{k=n} AP_k \tag{4}$$

III. Experimental Results and Discussions

To create the training and validation data set, initially having a relative small number of images (700), the images were augmented, this consisting of rotating 90°, 180°, and 270°, respectively, thereby forming three new images. All images used in these two phases must have the bounding box enclosing the HH insect, the coordinates being saved according to the architecture used: either as a text file (in the case of the YOLO series) or an XML file (in the case of the R-CNN series).

All simulated models were analyzed using the following constant hyper-parameters, presented in Table I where SGD is the Stochastic Gradient Descent Optimizer.

TABLE I. The Constant Hyper-Parameters Used.

Input image size	Epochs	Dataset	Learning rate	Optimizer
640	30	Maryland	0.01	SGD

In order to analyze HH insect recognition using artificial vision and machine learning techniques, the evaluation metrics proposed were applied to the Faster R-CNN, YOLOv5-s, YOLOv5-m, and YOLOv5-l models trained in the experiments. The convergence curves of the loss function for models are represented in Fig. 4.

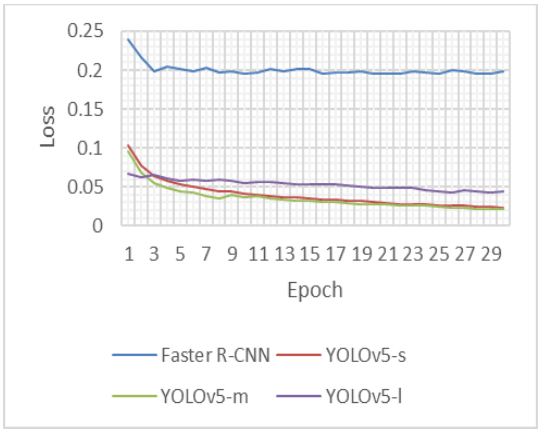

Figure 4. Training loss function convergence.

To analyze the performance in locating and recognizing the HH insect of the proposed Faster R-CNN, YOLOv5-s, YOLOv5-m, and YOLOv5-l models, P (Fig. 5) and R (Fig. 6) curves, depending on the number of epochs are presented.

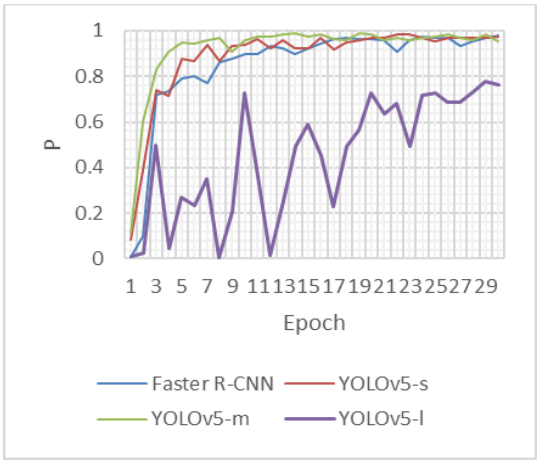

Figure 5. Precision plot.

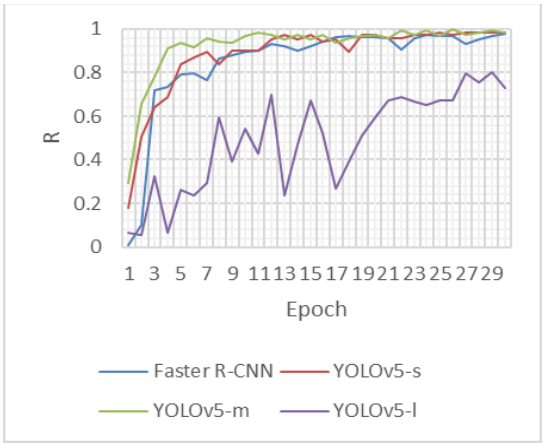

Figure 6. Recall plot.

In order to analyze the performance of the neural networks used, the validation was performed on 123 images from the dataset [20], other from the learning phase. The performance metrics values are presented in Table II. With bold were highlighted the higher values of the performances.

TABLE II. THE PERFORMANCES OF THE NEURAL NETWORKS USED.

Model	Precision (%)	Recall (%)	mAP (%)	Model size (MB)	Training time (h)	Testing time (s)
FR-CNN	87.1	88.0	89.1%	158	1.416	6.0
YOLOv5-s	88.3	88.3	89.4	**13.8**	**0.833**	**0.3**
YOLOv5-m	**95.3**	**98.4**	**99.2**	40.3	1.913	0.8
YOLOv5-l	73.8	75.4	77.0	88.6	3.207	1.6

An example of the detection results by the four neural networks applied to four images (Image 1, Image 2, Image 3, and Image 4) are shown in Fig. 7. The detection are corrected, except the Image 4 for YOLOv5-l.

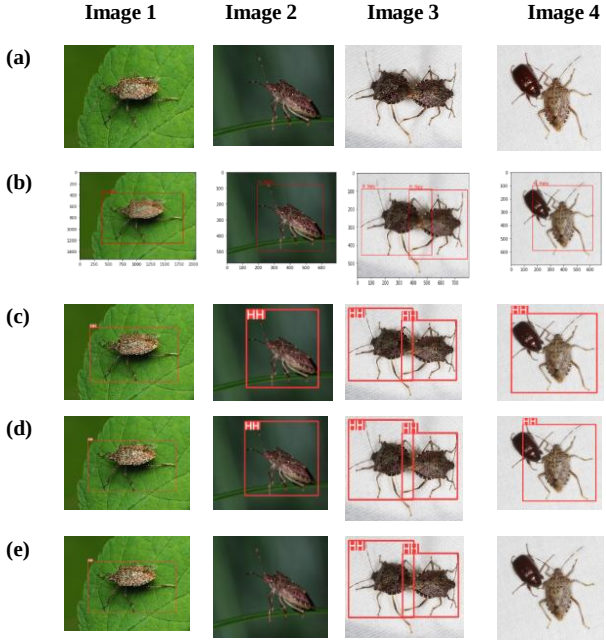

Figure 7. Experimental results. (a) Original images, (b) Detection results for Faster R-CNN, (c) Detection results for YOLOv5-s, (d) Detection results for YOLOv5-m, and (e) Detection results forYOLOv5-l.

Based on the experimental results, the following observation can be made:

(1) In Fig. 4, the loss function of the Faster R-CNN converges the slowest to the minimum value, followed by YOLOV5-l, YOLOv5-s, and YOLOv5-m. The minimum value of the Faster R-CNN loss function is the largest minimum value of all experiments.

(2) In Fig. 4, YOLOv5-l converges the slowest to the minimum value compared with YOLOv5-s and YOLOv5-m, trending to stabilize at a higher value.

(3) In Table II, the best results in terms of evaluation metrics were found in YOLOv5-m: precision 95.3%, recall 98.4%, and mAP 99.2%.

(4) In Table II, it can be seen that the average size of the YOLOv5 series is smaller than the R-CNN series. Even in the largest setup (YOLOv5-l with 88.6 MB), the YOLOv5 is smaller than the Faster-RCNN (158 MB).

(5) It can be seen in Table II that increasing depth and width in YOLOv5 does not guarantee better results. For every detection task, the model needs to be chosen according to a number of classes, difficulty, or processing power.

(6) In Fig. 7, the results of four corner cases are shown: H. Halys with a size greater than 30% of the image (Image 1), H. Halys on its side (Image 2), multiple H. Halys in the same image (Image 3), and H. Halys with another insert type in the same image (Image 4). In terms of detection, Faster R-CNN, and YOLOv5-m detected correctly all H. Halys (using a confidence score threshold of 50%). YOLOv5-s were detected correctly in cases of Image 1, Image 2, and Image 3. YOLOv5-l didn't perform the right detection for the four corner cases. It can also be seen in Fig. 6 that, overall, YOLOv5-m is the most precise method.

In the case of YOLOv5, we found a balance in terms of scale. It was shown that a larger network in terms of width and depth might not give the best results in HH detection because of the number of classes and dataset size, converting the large architecture (YOLOv5-l) into an inefficient network in terms of the time of training and results. Even if the results obtained are highly accurate, they can be improved in terms of both localization and classification by refining the hyper-parameters for YOLOv5-m, increasing training epochs, and adding more images.

IV. CONCLUSION

In this paper, neural network-based techniques were applied to HH detection. From R-CNN and YOLOv5 families, four models were studied. A dataset containing HH in different sizes and levels of evolution (adult or nymph) was collected from the Maryland website [20]. The images were labeled according to the requirements of the architecture: creating a TXT or XML containing the class information and bounding box coordinates. The model YOLOv5-m obtained the best results for the test dataset like the precision of 95.3% and the recall of 98.4%. Also, in terms of bounding boxes, YOLOv5-m obtained the best localization predictions. Results were positively influenced using GPU graphics processors made available through the Google Colab platform. As a feature work we intend to create a collective intelligence block based on multiple neural networks, selected by individual performances, to improve the statistic performances for detection and classification of harmful insects.

REFERENCES

[1] Rosalba Calvini, Veronica Ferrari, Lara Maistrello and Alessandro Ulrici, "Monitoring of insect pests in crop fields using spectral imaging", in the 20th International Conference on NIR 2021 Beijing.

[2] Sparks, M.E., Bansal, R., Benoit, J.B. et al. Brown marmorated stink bug, Halyomorpha halys (Stål), genome: putative underpinnings of polyphagy, insecticide resistance potential and biology of a top worldwide pest. BMC Genomics 21, 227 (2020). https://doi.org/10.1186/s12864-020-6510-7

[3] Leskey, T. C., & Nielsen, A. L. (2018). Impact of the Invasive Brown Marmorated Stink Bug in North America and Europe: History, Biology, Ecology, and Management. Annual Review of Entomology, 63(1), 599–618. https://doi.org/10.1146/annurev-ento-020117-043226

[4] M. Bergerman et al., "Robot Farmers: Autonomous Orchard Vehicles Help Tree Fruit Production," in IEEE Robotics & Automation Magazine, vol. 22, no. 1, pp. 54-63, March 2015, doi: 10.1109/MRA.2014.2369292.

[5] Zhang, Q., Karkee, M., & Tabb, A. (2019). The Use of Agricultural Robots in Orchard Management. CoRR, abs/1907.13114. http://arxiv.org/abs/1907.13114

[6] Girshick, R. B., Donahue, J., Darrell, T., & Malik, J. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. CoRR, abs/1311.2524. http://arxiv.org/abs/1311.2524

[7] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (n.d.). The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[8] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 115(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

[9] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet Classification with Deep Convolutional Neural Networks. Commun. ACM, 60(6), 84–90. https://doi.org/10.1145/3065386

[10] Alom, Md. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Essen, B. C. V., Awwal, A. A. S., & Asari, V. K. (2018). The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches. CoRR, abs/1803.01164. http://arxiv.org/abs/1803.01164

[11] He, K., Zhang, X., Ren, S., & Sun, J. (2014). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. CoRR, abs/1406.4729. http://arxiv.org/abs/1406.4729

[12] Girshick, R. B. (2015). Fast R-CNN. CoRR, abs/1504.08083. http://arxiv.org/abs/1504.08083

[13] Ananth, S. (2019). Faster R-CNN for object detection, Towards Data Science, https://towardsdatascience.com/faster-r-cnn-for-object-detection-a-technical-summary-474c5b857b46

[14] Khazri, A. (2019). "Faster RCNN object detection," https://towardsdatascience.com/faster-rcnn-object-detection-f865e5ed7fc4

[15] Redmon, J., & Farhadi, A. (2016). YOLO9000: Better, Faster, Stronger. CoRR, abs/1612.08242. http://arxiv.org/abs/1612.08242

[16] Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. CoRR, abs/2004.10934. https://arxiv.org/abs/2004.10934

[17] Jocher, G., Stoken, A., Borovec, J., and et al. (2020). ultralytics/yolov5: v3.1 - Bug fixes and performance improvements, https://zenodo.org/record/4154370/export/dcat#.YdiA98lBxaQ

[18] Yao, J., Qi, J., Zhang, J., Shao, H., Yang, J., & Li, X. (2021). A Real-Time Detection Algorithm for Kiwifruit Defects Based on YOLOv5. Electronics, 10(14). https://doi.org/10.3390/electronics10141711

[19] Kirillov, A., Girshick, R. B., He, K., & Dollár, P. (2019). Panoptic Feature Pyramid Networks. CoRR, abs/1901.02446. http://arxiv.org/abs/1901.02446

[20] Maryland Biodiversity Project, https://www.marylandbiodiversity.com/ (accessed on December 2021).