Proc. of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME 2023)
19-21 July 2023, Tenerife, Canary Islands, Spain

2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME) | 979-8-3503-2297-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICECCME57830.2023.10252700

Brown Marmorated Stink Bug Detection Using Multiple Neural Networks

Andrei Vicol
Faculty of Automatic Control and Computer Science
University POLITEHNICA of Bucharest
Bucharest, Romania
vicol.andrei2871@gmail.com

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

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

***Abstract*—The identification and management of pest infestations in a timely manner have been an ever-pressing issue, more so than ever now, when new pests have arrived from around the world and governments are encouraging farmers to use fewer and fewer pesticides. The use of drones and artificial intelligence in the monitoring of orchards has taken a strong rise in recent years due to the advantages offered by efficiency, coverage, and price. A particularly dangerous insect in fruit production is the brown marmorated stink bug (BMSB). The paper seeks to look at a performance comparison in BMSB detection between three states of the art models of image detection and classification: MobileNetV2, Xception, and EfficientNet. Each network model was slightly modified being thus adapted for this application and the results obtained had high accuracy, between 97% and 99%.**

Keywords—convolutional neural networks, image processing, insect detection, orchard monitoring

I. Introduction

In the case of monitoring agricultural crops and, in particular, orchards, methods and algorithms for the detection and evaluation of harmful insects have improved automation processes and provided governments or agricultural landowners with particularly useful information for appropriate actions. The automatic detection and monitoring of pests in agricultural crops with the help of remotely controlled image acquisition and processing systems based on neural networks represent a new trend in this field [1], [2].

Halyomorpha Halys (HH) is the scientific name for the colloquially named brown marmorated stink bug (BMSB), which has been shown to be a major problem for orchards all around Europe and North America [3], [4]. It has been accidentally introduced in Europe and the United States starting in the late 1990s where it has become a major threat to the health of fruits and trees, costing farmers upwards of tens of millions of euros in damages every season. Having been introduced from completely foreign ecosystems, BMSB presents itself with no natural enemies to limit its spread, thus continuing its territorial expansion to ever more states, both in North America and Europe.

Considering the financial losses this pest incurs, containing and eliminating it from orchards and other types of plantations has become a pressing need, one that promises to reduce and aims to eliminate the need for pesticides, as well as prevent any corresponding crop losses. Traditional insecticide-applying methods have proven to be slow and unable to keep up with the insect's spreading patterns, thus inspiring farmers and researchers to investigate novel ways of tackling this task.

To combat the spreading of pests effectively and accurately between crops, UAVs have been used in conjunction with machine learning algorithms to identify infested individuals based on images obtained from cameras mounted as payloads on multirotor or fixed-wing drones. As each type of crop requires specific approaches to warrant a successful identification task, different types of imaging techniques and machine-learning algorithms must be employed [5]. More specifically, for the task of identifying BMSB specimens on trees from orchards, multirotor drones must be equipped with cameras and an image classification algorithm ought to be chosen, developed, and trained.

Scientific works dealing with the detection and classification of insects from images are as old as the craft itself, but most of the time the imagery used is very specialized in its quality and origin. More specifically, many works frequently cited use images from known databases for their training steps; they are of great quality and will make it easy for good algorithms to extract features and make good predictions. Some examples of the most used public databases containing annotation insects are Maryland (https://www.marylandbiodiversity.com/) and IP102 (https://github.com/xpwu95/IP102). Based on these datasets, insects such as harmful bugs were classified with the help of neural networks or combinations of neural networks [6], [7].

While being great quality sources of images, taken by professionals to ensure the best visibility of specie-specific features, they are non-representative with respect to the real-world conditions that one might encounter in the field. The presence of intense sunlight, shading from nearby plants, or unfavorable angles makes for certain features to be less visible or for otherwise less evident features to be unrealistically exaggerated, thus bringing about supplementary confusion with respect to the species type. Even less apparent sources of error, such as cloud presence or plant-surface texture can cause unexpected shifts in light temperature or undesired reflection

and polarization of light. Furthermore, when considering the mission of using UAVs as the source of the images, supplementary hurdles, such as lesser proximity to specimens or a more vibrationally-intensive medium, must be taken into account as additional sources of noise, and possibly unusable images.

The growing popularity of machine learning among the public, coupled with a pressing need to obtain less chemically tainted produce while eliminating pests, has given rise to a new field of intelligent agriculture. Researchers and cultivators alike are investing more time and money than ever before in new technologies to help them fight pests and diseases without resourcing harmful chemicals.

One of these innovative approaches is the use of orchard or crop imagery to study and classify the insects most commonly affecting crops, which recently has also made use of drones and other UAV systems to acquire aerial imagery of the crops affected. The number of scientific resources regarding such studies is constantly increasing and the quality and accuracy of the developed methods are getting better and better.

To tackle this issue, one team of researchers developed a lightweight deep neural network called CPAFNet, built on a pre-existing CPAFNet architecture that was enhanced with the improvements in structure brought about by VGG16 and InceptionV3 [8]. This approach yielded 92.63% accuracy over 6000 iterative training steps, managing to surpass the VGG16 and InceptionV3 models.

Another approach to this task involved developing a fusing of high-accuracy models, namely ResNet50, AlexNet, VGG16, and InceptionResNetV2, into a computerized detection method to recognize three types of pests affecting citruses [9]. The team managed to get 99% accuracy, with the only limitation being the necessity to manually take all the necessary imagery. However, as the authors of the paper have mentioned, this opens up the door for using UAVs as a means of image acquisition.

Researchers making more intense use of UAVs have developed systems and methodologies of analysis by equipping multirotor UAVs with high-resolution cameras, taking a great many photos of the crop of interest, and using the images to train classification and segmentation models [5]. The images taken with the UAV were then used in conjunction with transfer learning and fine-tuning to obtain very good accuracies from InceptionV3, ResNet50, VGG16, VGG19, and Xception.

This current paper seeks to look at a performance comparison in BMSB detection between three states of the art models of image detection and classification: MobileNetV2, Xception, and EfficientNet. Each network model was slightly modified being thus adapted for this application and the results obtained had high accuracy, between 97% and 99%.

II. Materials and Methods

A. Dataset Used

The dataset used to train (by transfer learning) and test the proposed CNN (convolutional neural networks) models is comprised of images taken by a multirotor UAV in a pear orchard, as well as manually taken photos from the university's gardens. All images have been manually verified for their focus and split into two classes: one class for BMSB (class 'HH') and another one for other insects and plant environments (class 'Other'). The roughly prepared images have then been split into two directories, one corresponding to each class for further cleaning and augmentation. Pictured in Fig. 1, we have a selection of nine images pertaining to the 'HH' class. The selected images have already been cleaned and are readied for the learning process. The dataset contains images with BMSB both adults and nymphs.

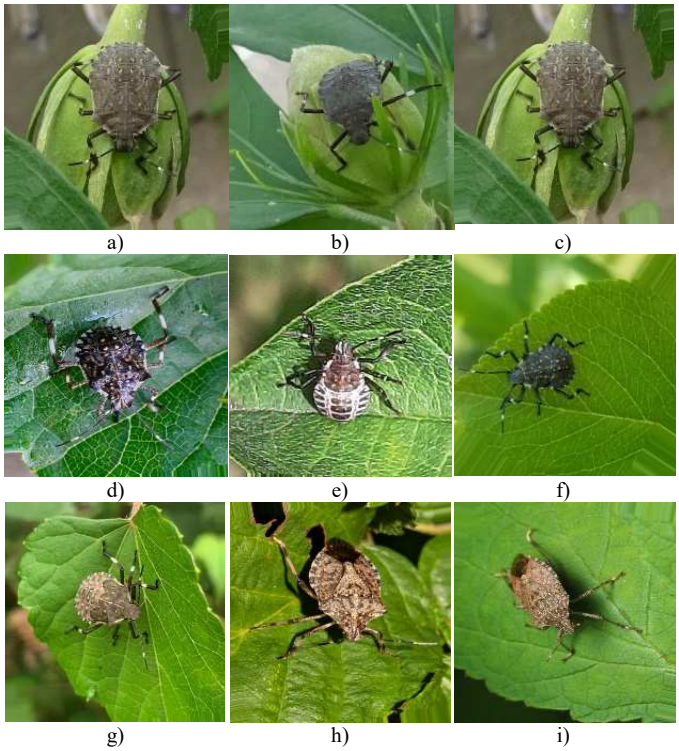

Fig. 1. Selection of class HH images.

Presented in Fig. 2, there is a nine-image collection of images belonging to the 'Other' class. Such images consist of vegetation and foliage corresponding to the usual environment of the BMSB insect, as well as other commonly encountered insects that might appear in real-world images acquired by drones in the field: Fig. 2 a, c, e, f, i – other insects; Fig. 2 b, d, g, h – background from the trees.

Splitting the dataset for different phases was done in the usual way, using 70% of the total images for training, 20% for validation, and the remainder 10% reserved for testing the model performance once training was completed. We considered a separate subset of images whose only purpose is to test the model performance, without having been used for training, to prevent the evaluation of the model from being skewed by previously learned features. Thus, from the total amount of 2000 sample images, 200 were set aside for accuracy, performance, and duration of obtaining predictions in the testing phase. Examples of test images are presented in Fig. 3: a, b, c – HH Class, adults or nymphs; d, f, e – Other Class.

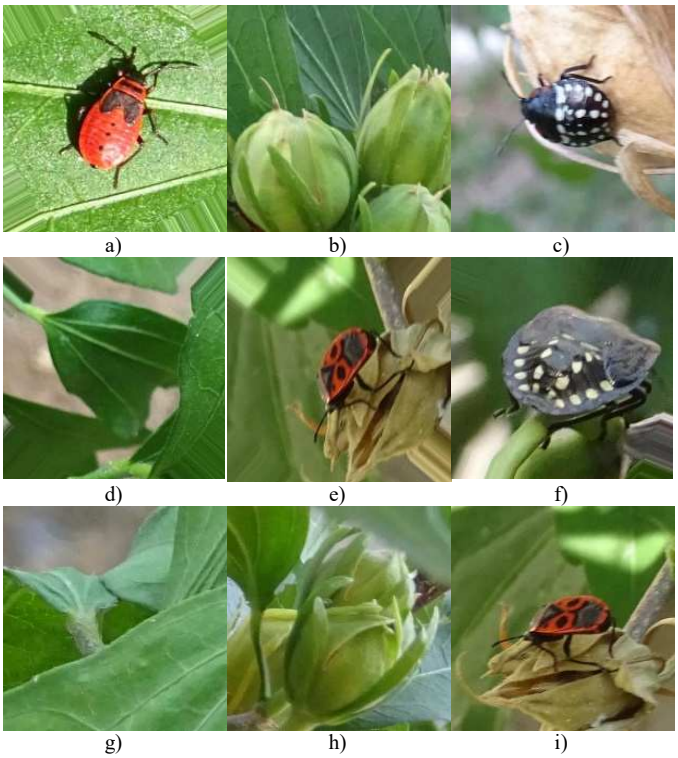

Fig. 2. Selection of Class Other images.

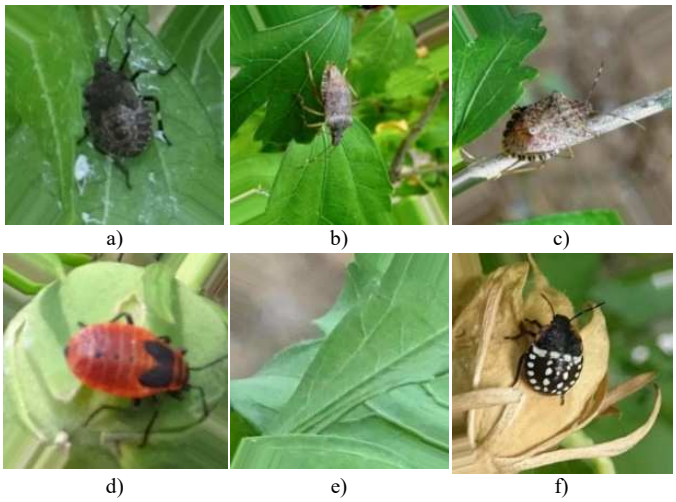

Fig. 3. Examples of test images.

B. Neural Networks Used

Three deep convolutional networks are tested and compared for HH detection: MobileNetV2, Xception, and EfficientNet. The need to select the most efficient neural networks for the detection and classification of HH will aim to create a multi-network system with a global decision, having superior detection performance of these insects in really difficult conditions (orchards).

Developed by engineers and scientists at Google, MobileNetV2 is a CNN that is comprised of 53 layers and was trained on the industry standard dataset ImageNet. Thanks to the extensive amount of image classes and information that ImageNet holds, MobileNetV2 can classify 1000 different image classes with the highest accuracy at the time of its inception [10]. Built on the grounds of MobileNetV1, this new model brings about improvements in performance, as well as a more robust architecture. With a solid performance base provided by the original MobileNetV1 model, the research team was able to create a more modern and better-performing model. The new model improves over the old one by centering its architecture around inverted residuals and linear bottlenecks. Inverted residuals allow the network to be more computationally efficient by reducing the size of the feature map while preserving accuracy. This is achieved by inverting the order of operations inside each computational block, starting with a 1×1 convolution followed by a 3×3 convolution to reduce the size of the feature map, ending with another 1×1 convolution to restore the shape of the tensor. Similarly, linear bottlenecks are blocks that make use of a linear projection layer for feature map reduction, a 3×3 convolution, and an output projection layer that increases the feature map size. This approach, together with the inverted residual blocks, helps make the model even more computationally efficient than its predecessor while offering state-of-the-art accuracy.

Like MobileNetV2, Xception is a machine learning model developed by Google and is a successor to InceptionV3, offering improvements in accuracy and a reduction in computational intensity. It is a deep learning model with modified depth separable convolutions that were used in the original iterations of the Inception models, replacing them with a modified version [11]. The new architecture takes a depth-wise separable convolution block, comprised of a depth-wise convolution followed by a pointwise convolution, and inverts the order of operations. This allows the model to perform a low-cost pointwise convolution at the input of the layer, following it with a depth-wise convolution after it, thus reducing the computational strain while maintaining high accuracy.

The EfficientNet B0 architecture [12] has an input layer and a series of repeated layers like the squeezed bottleneck, depth-wise separable convolution, and pointwise convolution. Each of these blocks is computationally efficient, reducing the number of parameters and increasing the model's capacity. This network model uses the scaling action of the architecture in three directions: depth, width, and resolution to optimize the performances.

C. Transfer Learning

The basic principle of transfer learning is that certain attributes acquired during the first task will be applicable to the second task, allowing the model to converge more quickly and efficiently. For the task at hand, transfer learning was used as the basis for furthering the learning and classification capabilities of pre-existing models, shortening the development duration as well as guaranteeing a well-tested network architecture. The models used are state-of-the-art in image classification tasks and have been trained on the ImageNet dataset which learns patterns and features for over 1000 different image classes.

D. Performance Metrics

To evaluate the performance of the models and obtain a clear assessment of each one's behavior, statistical indicators derived from the confusion matrix were used (Table I).

TABLE I. PERFORMANCE INDICATORS USED

Indicator	Formula
Accuracy	$ACC = \frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\frac{TP}{TP + FP}$
Dice Coefficient	$\frac{2 \cdot TP}{2 \cdot TP + FP + FN}$

The basic elements of the confusion matrix, marked in Table I are *TP* – true positive, *TN* – true negative, *FP* – false positive, and *FN* – false negative.

III. IMPLEMENTATION

A. Data Cleaning and Preparation

Before the images were ready for training the model, further processing was needed to ensure data cleanliness and correctness. The initially selected images were processed by custom software meant to allow the user to obtain a 512×512 image that, while still larger than what models require, retains enough usable detail to properly train the models.

To preserve the natural lighting and environmental conditions that one might find in an actual orchard, no color correction, histogram equalization, contrast enhancements, or sharpening filters were applied to the images. Instead, the only additional processing done to the initial dataset was an augmentation step to increase the size of the dataset and allow the model to learn from a more detailed set of images.

Augmentation was done in the form of morphological transformations to the images, applying a combination of mirroring, shearing, rotational, and flipping operations. To prevent overfitting the augmentation step would only produce one extra image per source image, doubling the size of the original collection of images without introducing many similar images. In the case of images that had a rotational transformation applied to them, filling in the missing data caused by rotating the original image was achieved using nearest interpolation.

B. Model Architectures

The bases for this work were Xception, MobileNetV2, and EfficientNet models which were augmented with additional layers to better train them on the working dataset. Common to the models is the input sequence which consists of an input layer that takes 224×224×3 images in 8-bit unsigned character format followed by a Rescaling layer that transforms those images into floating point format in the 0 to 1 range. The models have the 1000 neuron dense layer removed, as that was initially required for the 1000 classes that the ImageNet dataset was divided into. Instead, our models replace that layer with an output sequence specific to the model and which was found to have better accuracy and not lengthen the learning process unnecessarily.

In the case of the Xception model, its output sequence consists of a Global Average Pooling 2D layer, a Dropout layer to prevent overfitting by randomly dropping out 20% of the neurons, and a Dense layer with two neurons, pertaining to the two classes of images.

Table II presents the architecture of the model as complemented to better fit the dataset and allow for further feature learning.

For the MobileNetV2 model (Table III), the output sequence consists of a Global Average Pooling 2D layer, and a series of Dense layers, each with a decreasing number of parameters, all having a Dropout layer in between to reduce the chances of overfitting.

TABLE II. XCEPTION ADAPTED ARCHITECTURE

Input layer (224, 224, 3)	
Rescaling (224, 224, 3)	
Xception	
Global Average Pooling 2D	
Dropout (0.2)	
Dense (2)	
Total Parameters	20,863,529
Trainable Parameters	2,049

TABLE III. MOBILENETV2 ADAPTED ARCHITECTURE

Input layer (224, 224, 3)	
Rescaling	
MobileNetV2	
Global Average Pooling 2D	
Dense (512)	
Dropout (0.2)	
Dense (256)	
Dropout (0.2)	
Dense (64)	
Dropout (0.2)	
Dense (2)	
Total Parameters	3,061,762
Trainable Parameters	803,778

Compared with the Xception model, MobileNetV2 is larger in size, has more trainable parameters, and requires a slightly longer training period for each epoch. In total, both models were taught over 20 epochs.

The adaptation of the EfficientNet architecture to BMSB detection presented in Table IV is similar to those related to Xception and MobileNetV2 models.

TABLE IV. EFFICIENTNET ADAPTED ARCHITECTURE

Input layer (224, 224, 3)	
Rescaling Normalization	
EfficientNet B0, no pre-trained weights, classes=2	
Global Average Pooling 2D	
Dropout (0.5)	
Dense (2) + Softmax	
Total Parameters	4,335,998
Trainable Parameters	328,450

IV. Experimental Results and Discussions

A. Overview of Training Parameters and Performance

Having obtained a final dataset that was ready to be used as learning material for the models, training was commenced for each model with the same order of images for the dataset. The performance metrics used for both models are accuracy, precision, and the Dice coefficient.

Considering that the number of parameters and trainable parameters varies between the models, different results were obtained in terms of training time and time per epoch spent learning (Table V).

In Fig. 4 – Fig. 9 the accuracies and loss functions are presented for Xception, MobileNetV2, and EfficientNet models both for training and testing algorithms.

TABLE V. Training Parameters for Each Model

Model	Xception	MobileNetV2	EfficientNet
Total parameters	20 863 529	3 061 762	4 335 998
Trainable parameters	2 049	803 778	328 450
Training epochs	20	20	20
Time per epoch (s)	154	63	31
Total time (s)	3080	1260	620

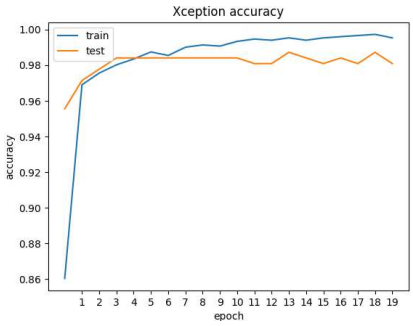

Fig. 4. Xception test and validation accuracy for 20 Epochs.

As can be seen in Fig. 4, Xception has a steadily increasing validation accuracy that seems to stabilize around 98%. While there might be a possibility to further increase the accuracy by increasing the number of epochs, the long training time per epoch, as can be seen in Table V, would not justify increasing the number of epochs.

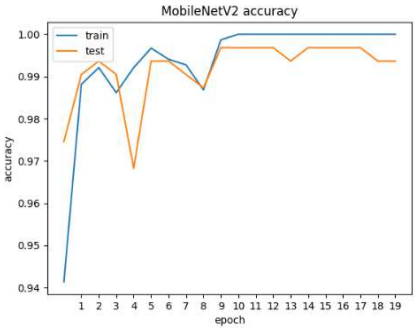

Fig. 5. MobileNetV2 test and validation accuracy.

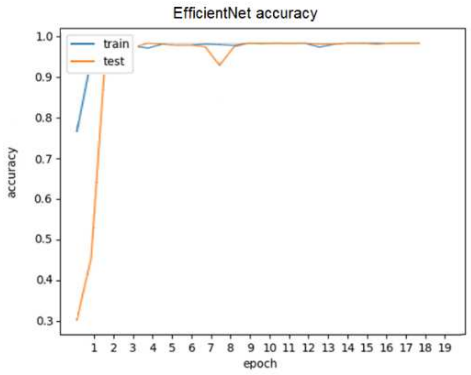

Fig. 6. EfficientNet training and testing loss.

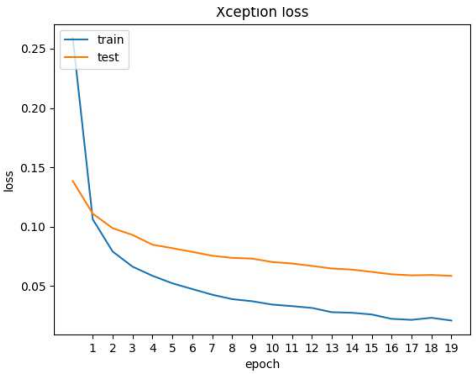

Fig. 7. Xception training and testing toss.

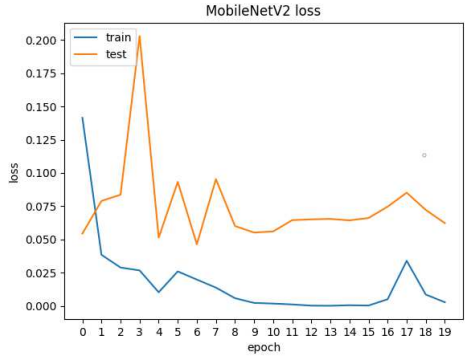

Fig. 8. MobileNetV2 training and testing loss.

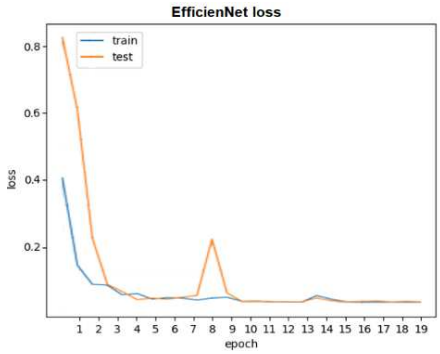

Fig. 9. EfficientNet training and testing loss.

Compared with its counterpart, MobileNetV2 (Fig. 5) has a higher validation accuracy and significantly shorter training times. Validation accuracy hovers around the 99% value with a slight variation between epochs.

Fig. 7, Fig. 8, and Fig. 9 present the loss functions for the three models, showing a decreasing trend with increases in the number of epochs, while MobileNetV2 has slightly lower loss values. Moreover, in the case of MobileNetV2 the loss function settles around values very close to zero, while in the case of Xception, its loss function settles around 0.07.

B. Overview of Testing Performance

For the testing phase, the models that were trained on the exact dataset, with the same order of images, were saved on local storage and prepared for future loadings. The exact same dataset was used for both models, consisting of 200 images from both classes. The accuracy, precision, and Dice coefficient were computed for the three models and are presented in Table VI and Table VII.

TABLE VI. TESTING RESULTS FOR CONFUSION MATRIX

	TP	**TN**	**FP**	**FN**
Xception	45	147	3	5
MobileNetV2	46	149	2	3
EfficientNet	46	148	3	3

TABLE VII. TESTING PERFORMANCES

CNN	**Xception**	**MobileNetV2**	**EfficientNet**
Accuracy	0.96	0.975	0.97
Precision	0.938	0.958	0.939
Dice Coefficient	0.918	0.948	0.939

Testing behavior is like the training and validation steps, with MobileNetV2 slightly outperforming Xception and EfficientNet in terms of accuracy, precision, and Dice coefficient.

V. CONCLUSIONS

Architecture adaptations were made for the proposed networks to suit the application, and transfer learning was used to learn the new model layers while preserving the large amount of pre-learned information that the original models had. The three models performed very well, with an accuracy between 96% and 97.5%. Regarding the training times, the values can be improved by using a more powerful computer. As future work, we will develop a system based on the fusion of the decisions of the best neural networks to increase the accuracy in detecting harmful insects.

ACKNOWLEDGMENT

This work was supported by HALY.ID project. HALY.ID is part of ERA-NET Co-fund 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] Apolo-Apolo Orly Enrique, Pérez-Ruiz Manuel, Martínez-Guanter Jorge, Valente João, A Cloud-Based Environment for Generating Yield Estimation Maps From Apple Orchards Using UAV Imagery and a Deep Learning Technique, Frontiers in Plant Science, volume 11, 2020, ISSN 1664-462X.

[2] Xu, C.; Yu, C.; Zhang, S.; Wang, X. Multi-Scale Convolution-Capsule Network for Crop Insect Pest Recognition. Electronics 2022, 11, 1630.

[3] M. E. Sparks., R. Bansal, J. B. Benoit, and 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, vol. 21, 227, March 2020.

[4] T. C. Leskey and A. L. Nielsen, "Impact of the invasive brown marmorated stink bug in North America and Europe: history, biology, ecology, and management," Annu Rev Entomol., vol. 63, pp. 599–618, 2018.

[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] Popkov, Alexander & Konstantinov, Fedor & Neimorovets, Vladimir & Solodovnikov, Alexey. (2022). Machine learning for expert-level image-based identification of very similar species in the hyperdiverse plant bug family Miridae (Hemiptera: Heteroptera). Systematic Entomology. 47. 10.1111/syen.12543.

[7] D. Popescu, T.L. Serghei, and L. Ichim: Dual Networks Based System for Detecting and Classifying Harmful Insects in Orchards, Proc. of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), 16-18 November 2022, Maldives

[8] J. Wang, Y. Li, H. Feng, L. Ren, X. Du, and J. Wu, "Common pests image recognition based on deep convolutional neural networks," Computers and Electronics in Agriculture, vol. 179, 105834, 2020.

[9] M. Khanramaki, E. A. Asli-Ardeh, and E. Kozegar, "Citrus pests classification using an ensemble of deep learning models," Computers and Electronics in Agriculture, vol. 186, 106192, 2021.

[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 4510–4520.

[11] F. Chollet, "Xception: deep learning with depthwise separable convolutions," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 1800–1807.

[12] M. Tan and Q. V. Le: "EfficientNet: rethinking model scaling for convolutional neural networks," Proceedings of the 36th International Conference on Machine Learning (ICML), 9-15 June 2019, Long Beach, California, USA, pp. 6105–6114.