Received March 10, 2022, accepted March 25, 2022, date of publication March 30, 2022, date of current version April 7, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3163380

Comparison of Deep Learning Techniques for Classification of the Insects in Order Level With Mobile Software Application

DURMUS OZDEMIR[1] **AND MUSA SELMAN KUNDURACI**[2]
[1]Department of Computer Engineering, Kutahya Dumlupinar University, 43100 Kutahya, Turkey
[2]Ministry of National Education, 16050 Bursa Turkey

Corresponding author: Durmus Ozdemir (durmus.ozdemir@dpu.edu.tr)

ABSTRACT Insects are a class of the arthropod branch and the most crowded animal group in terms of species and taxonomy. Due to destruction and forest fires, some insect species could go extinct without being detected. Identifying new insects and having knowledge about insects in terms of biodiversity will contribute positively to the studies carried out, especially in entomology, agriculture, the pharmaceutical industry, medicine, robotics, and other branches. In this study, we produced a mobile-based decision support software with a deep learning model to classify and detect insects at the order level. We also presented the comparative analysis results of SSD MobileNET, YoloV4, and Faster R-CNN InceptionV3 deep learning methods and adapting processes for order-level insect classification. Our approach studies the suitability of existing models towards such an objective, and we conclude that Faster R-CNN InceptionV3 performs the best at classifying and detecting insects at the order level. In addition, we shared 25820 training and 1500 test data in the kaggle database in order to contribute studies to be carried out in this area. As a result, we believe that this research will be beneficial to entomologists, naturalists, and other researchers in related fields.

INDEX TERMS Artificial intelligence, computers and information processing, insect classification.

I. INTRODUCTION

Entomology is a branch of zoology that includes scientific studies focusing on insect-related issues [1]. Due to the high insect populations, we found that more extensive research is needed for the order level of insects. In recent years, the desire to prevent insects from harming plants, animals, farmland, and people has been a reason for the increase in entomology studies [2], [3]. Also, entomology studies are essential because they offer new horizons and benefits to fields inspired by insects and nature, such as chemistry, medicine, pharmaceuticals, engineering, etc. [4]. A third of the world's crop is plundered and destroyed by insects.

For this reason, commercial losses are experienced due to the loss of many products [5]. The rapid and accurate identification of insects is essential whereby the prevention of economic losses and its contribution to the field of entomology [6], [7]. Also, insects inspire scientists in robots, sensors technologies, mechanical structures, aerodynamics, and intelligent systems [8]. The estimated number of species in insects is 1.5 million on the earth, but the number of named and defined species is around 750 thousand [9]. However, rarely does a new species continue to be discovered and named by scientists [10]. Due to the destruction and forest fires, some insect species are destroyed undetected [11]. For these reasons, academic studies on insect detection are essential for showing biodiversity [12]. When classifying insects, one of the essential criteria is determining the order level to which they belong. It is not possible to differentiate the type without determining the order level. As a result of scientific studies, 32 insect orders have been identified in nature. The newest insect orders were found in 2002. There are approximately 21 different criteria in order to determine, from the number of wings, body shape, number of feet, head shape [13]. In insect research using the traditional approaches, excessive time is needed due to many criteria to avoid some misdiagnoses [14]. When the literature was searched for insect detection at the order level, no decision support software, database, or program that can classify at the order level was found. Furthermore, it has been noted that there is no systematic deep learning comparison analysis that recognizes and classifies insect classification at the order level. Studies state that deep learning models, artificial intelligence, or machine learning algorithms are needed to classify and identify insects

The associate editor coordinating the review of this manuscript and approving it for publication was Khursheed Aurangzeb.

accurately and effectively [15], [16]. The major contributions of this paper can be summarized as follows: 1) A classification process covering all insect order levels was performed. 2) Comparative performance analysis of deep learning techniques for insect classification problem was presented. 3) Mobile-based decision support software for insect detection was presented. 4) An extensive database containing insect images that future studies can use was shared.

II. RELATED WORKS

Deep learning methods are used for education, e-commerce, industrial application, images, and processes in many areas where various datasets need to be analyzed [17]. When studies on object recognition with deep learning methods through images are examined, Krizhevsky *et al.*'s AlexNet study stand out. AlexNet won the ILSVRC, an Image classification competition, in 2012 [18]. Park *et al.* developed an insect classification application using the Squeeze-and-Excitation Networks module. In the application called SERAN, 34525 images for 123 classrooms were used in training. The developed algorithm is more successful than similar classification algorithms [19]. Deep Learning studies on object recognition gained speed with AlexNet. Previously, object recognition studies used classical image processing libraries [20]. There are studies in the literature stating that deep learning methods can be more efficient in insect classification and detection and that there is a need for studies using these methods [15], [21], [22].

The algorithm, called AdaBoost, carried out the classification of insects stored in granaries using Artificial Neural Network methods. The developed algorithm was compared with standard neural network methods. According to the experimental results, a significant improvement was obtained in the efficiency and classification accuracy of the new method [23]. In addition, a study called Automated Bee Identification System (ABIS), based on another convolutional neural network, observed mobile field investigations, including species identification of live bees in the field [24]. Lim *et al.* developed an algorithm that classifies butterflies and ladybugs. They used image processing techniques and classified according to the color, size, and position parameters [25]. They used OpenCV (a Computer Vision library) to develop the algorithm and introduced the 14 features of some insects to training with machine learning algorithms, and realized the identification of these insects. Lim *et al.* developed an application using CNN architecture to classify seven forest insects. In their application, firstly, the forest beetle is classified in the classifier on the web-based server then the result is sent to the application after. They study with common insect species, and also some of them are in the same insect order, and 22877 in total for training, 3861 for approval, and 2984 for the test were used [26]. For this reason, it is different from our study. In our study, insects are classified at the order level, and classification is made on rare insect orders [22].

Silva *et al.* used Euclidean distance and Dynamic Time Warping (DTW) classification methods for agricultural pests. They collected insect flight information and classified the insects by their genus with a laser light sensor. Moreover, owing to the audio recognition methods effectively collected features containing sufficient information for species identification provided real-time estimates [27].

Batista *et al.* and Chen *et al.* implemented their studies using sensors to detect insects. They used optical sensors to record the ''sound'' of insect flight from meters away, with total invariance to wind noise and ambient sounds. They stated that recent advances in sensor technology are beginning to come forward in the new field of Computational Entomology will emerge. Also stated that they train their models efficiently using a Bayesian classification approach [28], [29]. Lim *et al.* performed impact analysis for insect classification using CNN. In the developed algorithm, 27 classes and 1300 images were cropped to 256×256 dimensions for training and filtered. The images were prepared for training with CNN architecture. Later, the insect classification was made with the trained model, and its performance was measured [30]. Huynh *et al.* proposed a CDNN model for insect classification based on Neural Network and Deep Learning approaches. First, insect images were collected based on the intense scale unchanged property transformation. The properties pouch was used for image display as feature vectors. Finally, these feature vectors are trained and classified using the CDNN model based on the Deep Neural Network. This approach was developed to detect brown leaf flea and ladybug that damage rice in Mekong Delta, where rice is produced [31]. Xia *et al.* carried out insect classification research using VGG19 classification architecture developed from CNN. The study used 540 photographs obtained over the internet for 24 different species in deep learning. In the data increase phase, this number was increased to 4800. When the selected species are examined, it is seen that there are insects with the same order [32]. Buschbacher *et al.* proposed an Automated Bee Identification System (DeepABIS) that allows mobile field operations to be carried out on site, including the detection of live bee species. Using mobile smartphones and a cloud-based information gathering and communication network, DeepABIS provides participant detection scenarios. DeepABIS is flexible and transferable to other taxa, i.e., butterflies, flies, etc. [24]. Valan *et al.* carried out an SVM for the identification of insects. Two challenging tasks were tested in the study. Firstly, 884 face images from 11 families of the Diptera insect orders were found 96 percent accurate, and 2936 images from 14 families of the Coleoptera insect orders were 90 percent accurate. In the second task, a decision support system was created, which provided 96 percent accuracy in 339 images of three species of Colxytera genus Oxythyrea and 98.6 percent accuracy in Plecoptera larvae species in another dataset [33].

Khalifa *et al.* used an insect pest dataset containing 102 sub-classes. The chosen deep transfer learning models were AlexNet, GoogleNet, and Squeez-Net. These models were chosen based on their limited number of layers based on their architectures, reflecting the models' complexity, memory, and time. In order to make the models more robust and resolve the overfitting problem, data augmentation techniques were used by increasing the images of the dataset up to 4 times more than the original images. To prove the effectiveness of the selected models, the testing accuracy and performance metrics, such as the precision, recall, and F1 score, were measured [34].

When we examined the studies in the literature, we saw that the studies were generally carried out on insect species that are easy to distinguish and only for classification at the team level [35]. For this reason, the analyzes made in our study were evaluated on a comprehensive data set for 32 different order levels and all insect species. In addition, the application was carried out by comparing the performances of three different deep learning methods. Insect pests are one of the main factors affecting agricultural product production. With the development of computer algorithms and artificial intelligence, accurate and rapid identification of insect pests early can help prevent economic losses in the short and long term.

This study presents a comparative analysis of deep learning models for identifying and classifying insects at the order level. Our research, presented the adaptation processes, fine-tuning stages, and analysis results of the SSD MobileNET, YoloV4, and Faster R-CNN InceptionV3 deep learning methods for order-level insect classification. According to developed software, insect classification can be made easier and faster. The developed software is expected to be actively used in academic studies in Entomology. In addition, people interested in insects but do not have enough knowledge about entomology will be able to learn on their own which insect belongs to which order level by using the software we recommend. In the literature review and also preliminary interviews with researchers working on Entomology, it was seen that there was no comprehensive software study that detects and classifies at the order level in insect classification. Entomology researchers and students working in this field can be used the mobile decision support system we prepared in our study to classify insects in nature. In addition, the developed software will ensure the identification of pests and beneficial insects if actively used in agriculture. This will make agriculture more productive, thus contributing to the country's economy.

III. MATERIALS AND METHODS

A. DATA PREPROCESSING AND AUGMENTATION

Insects are a class of the arthropods branch and are the most populated animal group in terms of species. Thirty-two insect orders depend on the insect class. The datasets used in this study are insect images at the order level. When classifying insects, one of the essential criteria is determining the order to which they belong. It is not possible to determine the type without determining the order level. As a result of scientific studies, 32 insect orders were determined in nature. There are approximately 21 base criteria in insect order determination, from the number of wings to the body shape, from the number of feet to the head shape. Through these criteria, which are referred to as keys, entomologists can classify. These criteria are given in Table-1 below [36].

TABLE 1. Criterias for insect order determination.

Properties		
Number of Wings	Hind legs	Tentacles
Wing Shape	Tarsomere number	Body areas
Wing Pattern	Pretarsus	Body Shape
Wing evolution	Head Shape	Pronotum shape
Wing Vein	Antennas	Abdominal apex
Wing Base	Compound eyes	Abdominal base
Fore Legs	Mouthparts	Cerci

According to the values of these criteria, the order to which the insect belongs can be determined. Each criterion has many values; insect scientists evaluate their decisions based on these criteria and determine that the insect belongs to one of the 32 insect orders [13]. If these evaluated criteria do not meet the previous criteria and include different evaluation criteria, this may mean that a new order of insects has been discovered. For instance, the last discovered Mantophasmodea insect order was found in this way in 2002. Doubtless, this is a scarce situation [37]. With the software developed in our study, the classification process for all types of insect orders was made. Sample images of insect orders used for classification are shown in Figure 1.

Insect orders such as Coleoptera and Hymenoptera have a broad subclass compared to other Insect orders. The images of these insect orders are shown by selecting only one of the lower insect classes [1]. However, all these subclasses were used in the training and testing phases. In the software developed in this study, these learning criteria are provided with deep learning methods such as Faster R-CNN InceptionV3, SSD MobileNET, and YoloV4 methods. The Insect dataset of 25820 training images and 1500 test images are shared in the Kaggle database (https://www.kaggle.com/selmankundurac/insect-order-dataset) for aimed to contribute to the literature.

B. DEEP LEARNING FRAMEWORK: TENSORFLOW

Tensorflow is a free and open-source software library developed by Google Brain Team that can be used in artificial neural networks, deep learning, and genetic algorithms. It is used for both research and production at Google. Developed basically with the Python programming language, this library supports programming languages such as C ++, Java, Javascript, and R. It can be integrated into two ways, CPU and GPU. The CPU model uses the processor to process data, while the GPU model uses a graphics card processor [38]. Tensorflow is used for the Deep Learning process using an object detection library in this study.

FIGURE 1. The insect orders used for classification.

C. OPENCV LIBRARY AND DEEP LEARNING WITH PYTHON

OpenCV library contains more than 2500 algorithms such as classical image processing, computer vision, and machine learning. Many operations could be done with these methods, such as face recognition, object recognition, classification, tracking of objects, and extracting 3-dimensional models of objects in real-time or through offline systems. OpenCV has more than 47 thousand communities and more than 18 million downloads. The library is used by companies, public institutions, and research groups [39].

In this study, the OpenCV library was integrated with Tensorflow and used in image processing with Python programming language commands. Python is an object-oriented, interpreted, high-level programming language with its dynamic schema and modular structure that supports all kinds of data entry and class structures. Because it is platform-independent, it can be used in Unix, Linux, Mac, Windows, Amiga, Symbian operating systems. Also, the most crucial advantage that distinguishes Phyton from other programming languages is that it supports web applications, user interface applications, mobile applications, system applications, and databases. Python programming language was used in this study because the artificial intelligence and deep learning libraries are vibrant and easily adaptable.

D. FASTER R-CNN INCEPTIONV3, SSD AND YOLOV4 MODELS

Faster R-CNN is an artificial neural network algorithm created by combining RPN (Zone Proposal Algorithm) models [40]. This algorithm passes the entrance image through the conventional neural networks, and the feature map is drawn [21]. In this study, we used the InceptionV3 model from Faster R-CNN architecture. In Figure-2 general architecture and process of Faster R-CNN is presented [41]. In this stage, RPN is created, and region recommendations are made over this network. After the determined Network regions are reshaped, they are passed through their fully connected layers, and the classification process is performed. In this way, a faster estimation time is obtained.

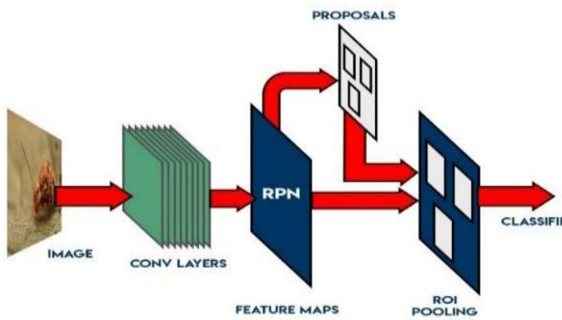

FIGURE 2. The architecture of the Faster R-CNN model.

SSD is a single-shot detector and can be used in real-time for object detection. It has no delegated region proposal network and predicts the boundary boxes and the classes directly from feature maps in one single pass. SSD introduces a few enhancements, including multi-scale functionality and default boxes, to retrieve the decrease in precision. To improve the SSD's performance, we added small convolutional filters, separate filters are used to handle the difference in aspect ratio in images, and multi-scale feature maps are used for object recognition [42]. In Figure-3 architecture of SSD is presented.

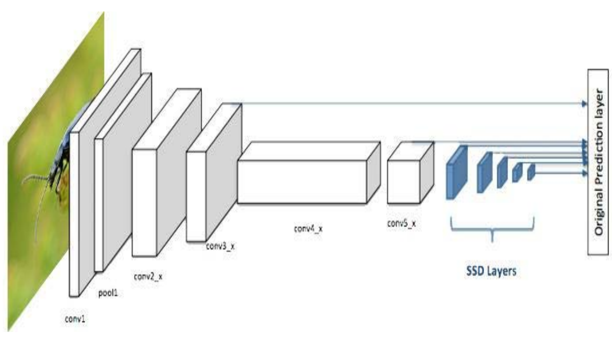

FIGURE 3. The architecture of the SSD model.

Another model we used for comparison in our study is the YOLOv4. YOLO (You Only Look Once) is an object detection algorithm using convolutional neural networks.

YOLO got this name because it can detect objects very quickly [43]. The YOLO model surrounds the objects it detects with a bounding box on the images. YOLO divides the input image into NxN grids. Each grid determines whether or not an object is in it, as well as the location of its center point within its area. Deciding that the object has a center point, the grid determines the object by finding the class, height, and width of that object and drawing a bounding box around that object. In the model created for this study, Yolov4 was used, and the training process was carried out with the TensorFlow library. The model file obtained after the training was converted into a tflite file and adapted to work in the mobile application. In Figure-4 architecture of YOLO is presented [44].

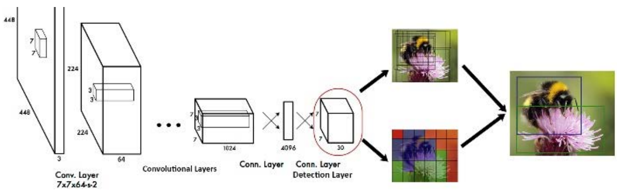

FIGURE 4. **The architecture of YOLO model.**

We selected the YOLOv4 model due to its good detection speed and accuracy in real-time applications and compared their accuracy and speed to investigate which algorithm performs best for insect detection in order level.

IV. PROPOSED APPROACH AND IMPLEMENTATION

In order to group insects at the order level with deep learning, the distinguishing features in the literature were determined, and the insect images in the training dataset were labeled. The images collected from search engines and databases such as ImageNet were subjected to image pre-processing such as light balance and image size adjustment. After the suitable images for training were tagged with labeling software, the methods created the datasets. The insect images put into training must reflect the distinctive features as clearly as possible for the classification system to work efficiently. Otherwise, the training process will be repeated many times since each image used in training will cause erroneous results.

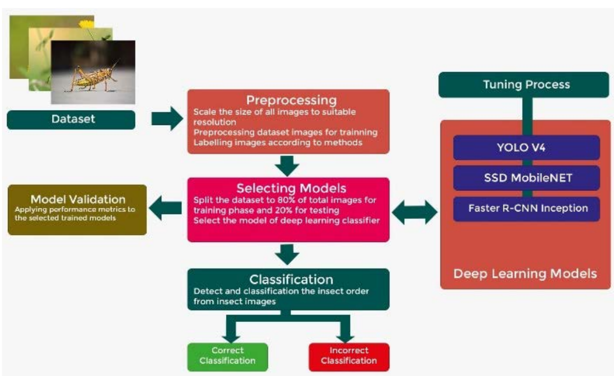

FIGURE 5. **The implementation process.**

In this study, the faster R-CNN-based InceptionV3, Yolo V4, and SSD MobilNet were compared and analyzed with deep learning and object classification methods. The Faster R-CNN-based Inception V3 approach is suitable for classification after about 120000 epochs, SSD MobileNET about 100000 epochs, and 100 epochs for YoloV4. In our study, out of 32 insect orders enrolled in training for classification, the errors in the datasets of insect orders with a low classification rate were corrected, reconstructed, and retrained. An improperly trained order will cause errors both in the classification of that insect order and other insect orders. For this reason, it is of great importance that the images collected for the insect orders belong to that insect order, that the image reflects the characteristic feature of the insect order, and that it is images from as many angles as possible.

A. EXPERIMENTAL SETTINGS

In object detection and classification, the size and characteristics of objects are significant for the correct result. In insects, features such as legs, antennae, wing structure, and number are used to distinguish the insect. Therefore, Anchor Boxes' width and height settings were changed in the training configuration. The size of the images should not be too low resolution or too high resolution because Distinctive features of the insect are lost in very low-resolution images. When the resolution is low, the error rate can be high. If the images are in too high resolution, the training time is prolonged, and it causes performance loss. Generally, the images are arranged between 720×720 pixels and 250×250 pixels using additional software. Trained Image Count (TIC) values belong to insect orders presented in Table-2. This table shows the number of TIC of tagged images used in training to classify each Insect order. Initially, a total of 3304 insect images were collected Image Count (IC), with a certain number of each class. Later, these images were extracted to 25820 images using Data Augmentation methods. The number of images after Data Augmentation is shown in the last column of the table (Augmented Image Count = AIC). As shown in Table 2, the number of images is low because the images of insect orders such as Zoraptera and Phthiraptera are low. Insect orders such as Hymenoptera, Lepidoptera, on the other hand, have more images since they are the more common insect orders. In order to reduce this imbalance and increase the success rate of classes with low image numbers, data augmentation processes were carried out at different rates. This method increases the success of classification, especially for rare insect species. For example, since Zoraptera is a rare insect order, approximately 25 photographs were obtained in the first stage. Subsequently, this number was increased to 900 with Data Augmentation. These images were shared in the kaggle database.

The study has been preferred for the Windows 10 operating system, the python programming language, Android 10 version for mobile application, and the anaconda platform for the deep learning environment. With the Tensorflow GPU version, additional libraries such as OpenCV, pillow, matplotlib have been installed in the working environment

TABLE 2. TIC values of insect order.

No	Insect Order	IC	AIC	No	Insect Order	IC	AIC
1	Archaeognatha	72	592	**17**	Mecoptera	102	816
2	Blattodea	99	896	**18**	Megaloptera	99	792
3	Coleoptera	124	992	**19**	Neuroptera	101	808
4	Collembola	96	768	**20**	Odonata	105	840
5	Dermaptera	100	800	**21**	orthoptera	104	832
6	Diplura	100	800	**22**	Phasmatodea	95	760
7	Diptera	100	800	**23**	Phthiraptera	21	720
8	Embioptera	60	480	**24**	Plecoptera	95	760
9	Ephemeroptera	100	800	**25**	Protura	101	808
10	Grylloblatodea	96	768	**26**	Psocoptera	95	760
11	Hemiptera	100	800	**27**	Siphonaptera	68	544
12	Hymenoptera	319	1752	**28**	Strepsiptera	97	776
13	Isoptera	125	1000	**29**	Thysanoptera	85	680
14	Lepidoptera	137	1096	**30**	Thysanura	81	648
15	Mantodea	102	816	**31**	Trichoptera	92	736
16	Mantophasmatodea	60	480	**32**	Zoraptera	21	900

of this platform. After the installation phase, the necessary coding and adjustments were made, and the training started.

B. LOSS FUNCTION

A loss value occurs after each epoch in training. This loss value calculates how far the predicted value is from the actual value, and while it is initially high, it decreases over time. If this loss value does not decrease or increases gradually, the training is terminated. This is the optimum value in the training epoch. The loss function value decreased further as the number of epochs increased.

$$L = \frac{1}{N_{cls}} \sum_{t} L_{cis}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_{i} p_i^* L_{reg}(t_i, t_i^*) \quad (1)$$

where L, L_{cls} and L_{reg} are the joint loss, classification loss, and regression loss of the border; N_{cls} and N_{reg} are the numbers of categories and boxes; λ and i represent the weight coefficient and the selected anchor box index; p_i and p_i^* represent the probability that candidate box i is the object, the value of the label (if the candidate box is a positive label, $p_i^* = 1$, otherwise $p_i^* = 0$); t_i^* is the predicted offset of the anchor box, and t_i^* is the offset between the anchor box and the actual object box. Also, in Figure-6, the graphic showing the relationship between the number of epochs and the loss value of the training is shown below. As seen in the graphic, the loss values, which were high at the beginning, dropped below 0.5 when the number of epochs reached 60000, and the training was terminated because it did not decrease more than 120000 epochs.

C. THE DESIGNED ANDROID APPLICATION

The mobile application-based decision support system presented in this study has been developed on the android studio platform, with Java and Kotlin languages, for android users. The mobile application has been developed by targeting the Android 10 version, and in order for it to work, the device must have a minimum Android 7.1 version. The interface of

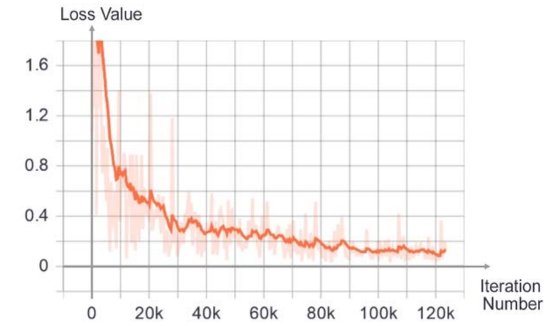

FIGURE 6. The number of epochs and loss value.

FIGURE 7. The deep learning-based mobile software interface.

the mobile application is shown in Figure 6. By pressing the button on the interface image, the user can select the insect image by taking a photo from the device's image gallery or the camera. Then, the comparison is made with the deep learning file embedded in the program, and the classification process is performed. After the classification, insect team name and prediction percentage are shown to the user, and the after the user is directed to the information page about the relevant insect. The user is given information about the insect team on the information page. SQLite, a local database, was used to store insect information. A tflite file was created for the training result of each model in this study. These files are integrated into the mobile application, and the required classifier class is written with the kotlin programming language. Because both the local database is used and the model tflite files are included in the program, users can classify and get information about the bug without needing an internet connection.

V. EXPERIMENTAL RESULTS

In order to measure the classification success of the proposed approach in different methods, test images were created with a certain number of images from each class.

TABLE 3. Precision, recall (or correct rate) and F1 score values.

	Faster R-CNN			SSD Model			YOLOv4 Model			*Total Number of Tests*
Insect Order	***P***	***R***	***F1***	***P***	***R***	***F1***	***P***	***R***	***F1***	
Archaeognatha	**0,90**	0,71	**0,80**	0,86	0,46	0,60	0,81	0,74	0,77	52
Blattodea	**0,89**	0,83	**0,86**	0,87	0,68	0,76	0,88	0,80	0,84	40
Coleoptera	**0,84**	0,94	**0,89**	0,66	0,76	0,70	0,80	0,90	0,85	33
Collembola	**0,71**	0,96	**0,81**	0,53	0,68	0,60	0,70	0,92	0,79	25
Dermaptera	**0,93**	0,61	**0,74**	0,90	0,42	0,57	0,90	0,60	0,72	62
Diplura	0,63	0,92	**0,75**	0,53	0,82	0,64	0,62	0,90	0,73	39
Diptera	0,77	0,83	**0,80**	0,51	0,71	0,60	0,72	0,80	0,76	52
Embioptera	0,74	0,74	**0,74**	0,50	0,49	0,49	0,71	0,72	0,71	35
Ephemeroptera	0,87	0,68	**0,76**	0,82	0,55	0,66	0,85	0,69	**0,76**	107
Grylloblatodea	0,73	0,86	**0,79**	0,59	0,77	0,67	0,71	0,80	0,75	22
Hemiptera	0,85	0,92	**0,89**	0,73	0,75	0,74	0,81	0,88	0,84	76
Hymenoptera	0,68	0,87	**0,76**	0,40	0,78	0,53	0,63	0,82	0,71	37
Isoptera	0,95	0,78	**0,86**	0,91	0,69	0,78	0,80	0,79	0,79	73
Lepidoptera	0,93	0,88	**0,90**	0,82	0,72	0,77	0,82	0,81	0,81	64
Mantodea	0,83	0,92	**0,87**	0,75	0,88	0,81	0,81	0,89	0,85	85
Mantophasmotodea	0,75	0,65	0,70	0,56	0,39	0,46	0,72	0,73	**0,72**	23
Mecoptera	0,78	0,83	**0,80**	0,77	0,77	0,77	0,74	0,80	0,77	47
Megaloptera	0,77	0,80	**0,79**	0,54	0,67	0,60	0,77	0,79	0,78	30
Neuroptera	0,79	0,82	**0,81**	0,59	0,61	0,60	0,78	0,80	0,79	33
Odonata	0,79	0,76	**0,78**	0,66	0,66	0,66	0,79	0,75	0,77	55
Orthgptrea	0,93	0,73	**0,82**	0,60	0,67	0,63	0,93	0,70	0,80	63
Phasmatodea	0,69	0,73	**0,71**	0,58	0,75	0,65	0,66	0,71	0,68	20
Phthiraptera	0,67	0,90	**0,77**	0,57	0,69	0,62	0,62	0,89	0,73	71
Plecoptera	0,71	0,83	**0,77**	0,78	0,97	0,87	0,69	0,82	0,75	30
Protura	0,85	0,97	**0,91**	0,58	0,73	0,65	0,81	0,89	0,85	41
Psocoptera	0,74	0,88	**0,80**	0,77	0,39	0,51	0,71	0,92	**0,80**	26
Siphonaptera	0,91	0,73	**0,81**	0,75	0,94	0,83	0,90	0,71	0,79	32
Strepsiptera	0,81	0,94	**0,87**	0,68	0,51	0,58	0,80	0,90	0,85	55
Thysanoptera	0,88	0,78	**0,83**	0,48	0,74	0,58	0,85	0,76	0,80	31
Thysanura	0,68	0,81	**0,74**	0,73	0,75	0,74	0,67	0,80	0,72	44
Trichoptera	0,90	0,80	**0,84**	0,83	0,60	0,69	0,87	0,79	0,83	42
Zoreptera	0,91	0,76	**0,83**	0,89	0,56	0,69	0,91	0,75	0,82	55
Accuracy			**0,808**			0,669			0,778	1500
Macro AVG	0,806	0,817	**0,805**	0,679	0,673	0,658	0,774	0,799	0,786	1500
Weighted AVG	0,821	0,808	**0,808**	0,705	0,669	0,669	0,787	0,795	0,791	1500

Almost 1500 insect images in total have been tested in three different classification methods. The classification value for each mage was first transferred to text files, and then the results of all classes were transferred to the confusion matrix numerically. The highest values with IoU (Intersection over Union) value IoU >= 0.50 and above were accepted as successful classification values in the tests. The most common performance measurements in the field of Deep Learning are used and presented respectively; the Test Accuracy Eq.(1), Precision Eq.(2), Recall Eq.(3), and F1 Score Eq.(4) [45]. The confusion matrix should be created to obtain TP, TN, FP, and FN values. The TP value indicates the number of correct predictions, the FN value indicates the number of false predictions, FP indicates the number of classes in other classes, and the TN value represents the total number that class is not related to. After these values are obtained, Test Accuracy, Precision, Recall, and F1 scores are calculated for each insect order. The formulas used for the calculation are given below. The F1 Score (4) obtained at the end of the measures the study's success. A comparison of the Inception V3, SSD, and YOLOv4 models is given in Table 3. The number of insect images tested in the "total number of tests" column in Table-3 and the classification success for each class in the other columns are presented by obtaining Sensitivity, Recall, and F1 Scores.

$$Testing\ Accuracy = \frac{TN + TP}{(TN + TP + FN + FP)} \tag{2}$$

$$Precision = \frac{TP}{(TP + FP)} \tag{3}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{4}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

Besides, each model's accuracy, macro average, and weighted average results were calculated. In Table-3, the highest F1 scores of the three models were shown in bold. The F1 score of the Inception V3 model was higher than the other models in almost all classes. The Inception V3 model was the most successful classification with the highest F1 score was in the "Protura" class with 0.91. The lowest F1 score in the Inception V3 model was in the Mantophasmotodea class with 0.70. Since Mantophasmotodea was a rare insect order, the F1 score may have been low.

Precision is the rate at which a class appears in the results of other classes as a result of classification. If it is close to 1, that class can be said to be less in the results of other classes, and if it is closer to 0, it can be more. According to this definition, the Faster R-CNN-based Inception V3 model's Precision value is 0.67 in the most common insect group, called Phthiraptera order. The lowest Precision value was Hymenoptera with 0.40 in the SSD Model and Phthiraptera and Diplura with 0.62 in the YOLO V4 Model. The Recall value is obtained by correct classification results divided by the number of results that should have been returned. The recall value results of the Inception V3 model were Collembola, which had the highest score of 0,96, and Dermaptera as the lowest score of 0,61. In the SSD Model, the highest Recall value belongs to the Plecoptera class with 0.97, while the lowest Recall value belongs to the Mantophasmotodea and Psocoptera classes with 0.39. In the YOLO Model, the highest Recall value belongs to the Collembola and Psocoptera classes with 0,92, and the lowest Recall value belongs to the Dermaptera class with 0,60.

TABLE 4. Comparision of methods.

Models	Accuracy	Training Time (Windows 10)	Classification Time (Windows 10)	Classification Time (Android 10)
Faster R-CNN Inception V3	**0,808**	6,9h	27 ms	85 ms
SSD MobileNET	0,669	5,1h	34 ms	92 ms
Yolo V4	0,778	9,2h	17 ms	81 ms

Table 4 presents the accuracy values of the three methods, training cost time, and how long each test image was classified. The fourth column in Table-4 shows the classification times of the three models used in the testing phase of an image in the Windows 10 operating system. The Faster R-CNN-based Inception V3 model performed classification in 27 milliseconds, SSD MobilNET in 34 milliseconds, and Yolo V4 in 17 milliseconds. Accordingly, the fastest model in the Windows 10 operating system was Yolo V4. This measurement was made on a computer with an Intel Core i7 8700 3.2 GHz processor, 32 GB RAM, 4 GB, 4 GB Nvidia 1050ti, and a 1 TB SSD hard disk. The fifth column shows the comparison and classification cost times on an Android device. When the classification times on a smartphone with 512 GB RAM and Android 10 are examined, it is seen that it is 85 ms in the Faster R-CNN based approach, 82 ms in the SSD MobileNET, and 81 ms in the Yolo V4 model. When the models are compared, the fastest method is again the YoloV4 model. Classification times may vary depending on the hardware and software features of the device. According to accuracy values, the Faster R-CNN-based Inception V3 model 0,808 has achieved a more successful result than other models. SSD MobileNET has failed compared to other models. In the images below (Figure 7), the classification results in the images of the Ephemeroptera and Plecoptera insect orders can be seen. These results show an Ephemeroptera insect order, which is correctly detected with 99 percent, in Figure 7(a). In Figure 7(a), the characteristic features of the Ephemeroptera insect order are fully reflected. In the image, however, the insect appears clearly. For this reason, the correct classification was made with a high rate of 99 percent. Another classification image of the same insect order is shown in Figure 7(b). Here, the insect is not as clear as in the other image. The software classifies, albeit at very low percentages. Nevertheless, low rates are not shown as a result of classification. An incorrectly detected Ephemeroptera insect order in Figure 7(c) is seen. The software has mixed the Plecoptera insect order with the Ephemeroptera insect order. The correct classification result for the mixed species can be seen in Figure 7(d). In Figure 7(c), the software made the wrong classification because the characteristics of the insect order are similar to the Plecoptera insect order. The distinguishing criteria must be evident in the insect image for a successful classification when the software results are examined. The mistakes can be eliminated by increasing the number of training transactions, but the prominent presence of the insect in the image is essential for successful classification.

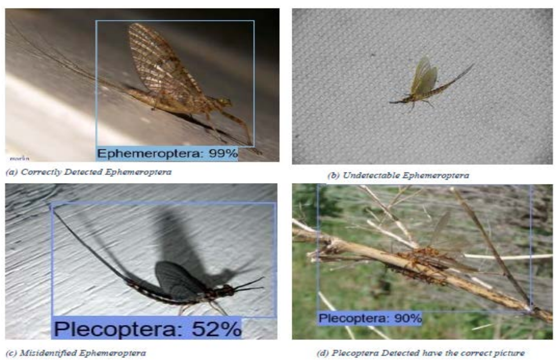

FIGURE 8. Ephemeroptera and plecoptera insect orders.

VI. DISCUSSION AND CONCLUSION

When the test results of the compared deep learning models were evaluated in general, the insects were successfully detected and classified in order level. The Faster R-CNN-based Inception V3 model was most successful in correctly identifying and classifying insects at the order level. In terms of test time, the Yolo V4 model has an advantage over other models because it gave the best results on both Android and Windows operating systems. Although we concluded in this study that the Faster R-CNN model is more successful than the other models we compared, some studies in the literature show that the accuracy value is high in various models [46]. We foresee that this is due to differences in dataset structure, problem type, and fine-tuning training stages.

We recommend the Faster R-CNN-based Inception V3 model because it gives successful results in terms of accuracy in tests and is better than other methods even in low-quality photos with blurry and low brightness. There are also insect images that all three models used could not detect. Image-related causes of this situation are insufficient light, the insect being too small in the image, the insect's image not being clear, and a different object in the image that makes it difficult to detect an object. Another situation is that the models cannot detect it because the insect photograph cannot reflect the characteristic features of the insect. Not only for insect detection but also for all objects to be detected, the characteristic features of that object should be reflected in the image. The findings of this study demonstrated the performance of deep learning models and how the models produced results in a real-world problem such as determining the order level of insects using insect images. It is anticipated that the research results and the mobile application presented will contribute to researchers working in the fields of insects.The insect data set we created from 25820 images has been shared publicly in the kaggle environment. We aim to contribute to the literature as a primary resource for related studies in order-level insect classification and detection processes in future studies. Researchers can perform different studies by developing this data set or testing it with different algorithms and deep learning models.

Researchers who want to do insect detection using Deep Learning methods can achieve more successful results by reducing the number of classes to less and using more datasets in training. More comprehensive results can be obtained for future studies by expanding the dataset, especially for rare insects, and by working with entomologists to enrich the data sets. In addition, due to the easy use of the model and mobile software proposed in the study, people who are not directly related to entomology but work in fields such as agriculture will detect insects that may damage agricultural products using this software. As a result, the proposed deep learning-based mobile software will detect harmful insects, increase agricultural productivity, and indirectly contribute to the national economy. In addition, the proposed software is expected to contribute positively to the decision-making processes of Entomology experts.

ACKNOWLEDGMENT

The authors would like to thank Dr. Yakup Senyuz for his suggestions and comments on their study.

REFERENCES

[1] D. F. Cheng, K. M. Wu, Z. Tian, L. P. Wen, and Z. R. Shen, "Acquisition and analysis of migration data from the digitised display of a scanning entomological radar," *Comput. Electron. Agricult.*, vol. 35, nos. 2–3, pp. 63–75, Aug. 2002.

[2] M. Martineau, D. Conte, R. Raveaux, I. Arnault, D. Munier, and G. Venturini, "A survey on image-based insect classification," *Pattern Recognit.*, vol. 65, pp. 273–284, May 2017.

[3] K. Thenmozhi and U. S. Reddy, "Crop pest classification based on deep convolutional neural network and transfer learning," *Comput. Electron. Agricult.*, vol. 164, Sep. 2019, Art. no. 104906.

[4] J. Mlcek, M. Borkovcova, O. Rop, and M. Bednarova, "Biologically active substances of edible insects and their use in agriculture, veterinary and human medicine," *J. Cent. Eur. Agric.*, vol. 15, no. 4, pp. 1–13, 2014.

[5] C. M. Oliveira, A. M. Auad, S. M. Mendes, and M. R. Frizzas, "Crop losses and the economic impact of insect pests on Brazilian agriculture," *Crop Protection*, vol. 56, pp. 50–54, Feb. 2014.

[6] C. Xie, J. Zhang, R. Li, J. Li, P. Hong, and J. Xia, "Automatic classification for field crop insects via multiple-task sparse representation and multiple-kernel learning," *Comput. Electron. Agricult.*, vol. 119, pp. 123–132, Nov. 2015.

[7] Y. Senyuz, "The insects: An outline of entomology/Böcekler: Entomolojinin ana hatları," in *The Insects: An Outline of Entomology*, S. Candan, H. Koç, and A. Gök, Eds. Ankara, Turkey: Nobel Yayın Dağıtım, 2012.

[8] J. R. Serres and S. Viollet, "Insect-inspired vision for autonomous vehicles," *Current Opinion Insect Sci.*, vol. 30, pp. 46–51, Dec. 2018.

[9] R. Fox, C. A. Harrower, J. R. Bell, C. R. Shortall, I. Middlebrook, and R. J. Wilson, "Insect population trends and the IUCN red list process," *J. Insect Conservation*, vol. 23, no. 2, pp. 269–278, Apr. 2019.

[10] V. Ngô-Muller, R. Garrouste, and A. Nel, "Small but important: A piece of mid-cretaceous burmese amber with a new genus and two new insect species (odonata: Burmaphlebiidae & 'psocoptera': Compsocidae)," *Cretaceous Res.*, vol. 110, Jun. 2020, Art. no. 104405.

[11] M. A. Krawchuk, G. W. Meigs, J. M. Cartwright, J. D. Coop, R. Davis, A. Holz, C. Kolden, and A. J. Meddens, "Disturbance refugia within mosaics of forest fire, drought, and insect outbreaks," *Frontiers Ecol. Environ.*, vol. 18, no. 5, pp. 235–244, Jun. 2020.

[12] S. Kaloudis, D. Anastopoulos, C. P. Yialouris, N. A. Lorentzos, and A. B. Sideridis, "Insect identification expert system for forest protection," *Expert Syst. Appl.*, vol. 28, no. 3, pp. 445–452, Apr. 2005.

[13] P. J. Gullan and P. S. Cranston, *The Insects: An Outline of Entomology*. Hoboken, NJ, USA: Wiley, 2014.

[14] Y. S. Chang and E. S. Jun, "An intelligent insect search system based on observation of the insect's structure," *Expert Syst. Appl.*, vol. 42, no. 6, pp. 2975–2984, Apr. 2015.

[15] D. I. Patrício and R. Rieder, "Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review," *Comput. Electron. Agricult.*, vol. 153, pp. 69–81, Oct. 2018.

[16] M. Tuda and A. I. Luna-Maldonado, "Image-based insect species and gender classification by trained supervised machine learning algorithms," *Ecol. Informat.*, vol. 60, Nov. 2020, Art. no. 101135.

[17] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 2, pp. 84–90, Jun. 2017.

[19] Y. J. Park, G. Tuxworth, and J. Zhou, "Insect classification using squeeze-and-excitation and attention Modules–a benchmark study," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 3437–3441.

[20] H. S. Maghdid, A. T. Asaad, K. Z. Ghafoor, A. S. Sadiq, and M. K. Khan, "Diagnosing COVID-19 pneumonia from X-ray and CT images using deep learning and transfer learning algorithms," *Proc. SPIE*, vol. 11734, Apr. 2021, Art. no. 117340E.

[21] H. Jiang, C. Zhang, Y. Qiao, Z. Zhang, W. Zhang, and C. Song, "CNN feature based graph convolutional network for weed and crop recognition in smart farming," *Comput. Electron. Agricult.*, vol. 174, Jul. 2020, Art. no. 105450.

[22] S. Lim, S. Kim, S. Park, and D. Kim, "Development of application for forest insect classification using CNN," in *Proc. 15th Int. Conf. Control, Autom., Robot. Vis. (ICARCV)*, Nov. 2018, pp. 1128–1131.

[23] H. Zhang, Q. Huo, and W. Ding, "The application of AdaBoost-neural network in storedproduct insect classification," in *Proc. IEEE Int. Symp. IT Med. Educ.*, Dec. 2008, pp. 973–976.

[24] K. Buschbacher, D. Ahrens, M. Espeland, and V. Steinhage, "Image-based species identification of wild bees using convolutional neural networks," *Ecol. Informat.*, vol. 55, Jan. 2020, Art. no. 101017.

[25] J. Lim, J. Cho, T. Nam, and S. Kim, "Development of a classification algorithm for butterflies and ladybugs," in *Proc. IEEE Region 10th Conf. (TENCON)*, Nov. 2006, pp. 1–3.

[26] H. Yang, W. Liu, K. Xing, J. Qiao, X. Wang, L. Gao, and Z. Shen, "Research on insect identification based on pattern recognition technology," in *Proc. 6th Int. Conf. Natural Comput.*, Aug. 2010, pp. 545–548.

[27] D. F. Silva, V. M. A. Souza, D. P. W. Ellis, E. J. Keogh, and G. E. A. P. A. Batista, "Exploring low cost laser sensors to identify flying insect species," *J. Intell. Robotic Syst.*, vol. 80, no. 1, pp. 313–330, Dec. 2015.

[28] G. E. Batista, E. J. Keogh, A. Mafra-Neto, and E. Rowton, "SIGKDD demo: Sensors and software to allow computational entomology, an emerging application of data mining," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2011, pp. 761–764.

[29] Y. Chen, A. Why, G. Batista, A. Mafra-Neto, and E. Keogh, "Flying insect classification with inexpensive sensors," *J. Insect Behav.*, vol. 27, no. 5, pp. 657–677, Sep. 2014.

[30] S. Lim, S. Kim, and D. Kim, "Performance effect analysis for insect classification using convolutional neural network," in *Proc. 7th IEEE Int. Conf. Control Syst., Comput. Eng. (ICCSCE)*, Nov. 2017, pp. 210–215.

[31] H. X. Huynh, D. B. Lam, T. Van Ho, D. T. Le, and L. M. Le, "CDNN model for insect classification based on deep neural network approach," in *Context-Aware Systems and Applications, and Nature of Computation and Communication* (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering), vol. 298, P. Vinh and A. Rakib, Eds. Cham, Switzerland: Springer, 2019, doi: 10.1007/978-3-030-34365-1_10.

[32] D. Xia, P. Chen, B. Wang, J. Zhang, and C. Xie, "Insect detection and classification based on an improved convolutional neural network," *Sensors*, vol. 18, no. 12, p. 4169, 2018.

[33] M. Valan, K. Makonyi, A. Maki, D. Vondráček, and F. Ronquist, "Automated taxonomic identification of insects with expert-level accuracy using effective feature transfer from convolutional networks," *Systematic Biol.*, vol. 68, no. 6, pp. 876–895, Nov. 2019.

[34] N. E. M. Khalifa, M. Loey, and M. H. N. Taha, "Insect pests recognition based on deep transfer learning models," *J. Theor. Appl. Inf. Technol.*, vol. 98, no. 1, pp. 60–68, 2020.

[35] Y. Kaya, L. Kayci, and M. Uyar, "Automatic identification of butterfly species based on local binary patterns and artificial neural network," *Appl. Soft Comput.*, vol. 28, pp. 132–137, Mar. 2015.

[36] V. Clausnitzer, V. J. Kalkman, M. Ram, B. Collen, J. E. M. Baillie, M. Bedjanič, W. R. T. Darwall, K.-D.-B. Dijkstra, R. Dow, J. Hawking, H. Karube, E. Malikova, D. Paulson, K. Schütte, F. Suhling, R. J. Villanueva, N. von Ellenrieder, and K. Wilson, "Odonata enter the biodiversity crisis debate: The first global assessment of an insect group," *Biol. Conservation*, vol. 142, no. 8, pp. 1864–1869, Aug. 2009.

[37] S. Roth, J. Molina, and R. Predel, "Biodiversity, ecology, and behavior of the recently discovered insect order mantophasmatodea," *Frontiers Zool.*, vol. 11, no. 1, p. 70, Dec. 2014.

[38] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, and M. Kudlur, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operating Syst. Design Implement. (OSDI)*, 2016, pp. 265–283.

[39] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision With the OpenCV Library*. Sebastopol, CA, USA: O'Reilly's, 2008.

[40] L. Jiao, S. Dong, S. Zhang, C. Xie, and H. Wang, "AF-RCNN: An anchor-free convolutional neural network for multi-categories agricultural pest detection," *Comput. Electron. Agricult.*, vol. 174, Jul. 2020, Art. no. 105522.

[41] W. Yang, Z. Li, C. Wang, and J. Li, "A multi-task faster R-CNN method for 3D vehicle detection based on a single image," *Appl. Soft Comput.*, vol. 95, Oct. 2020, Art. no. 106533.

[42] S. Gu, L. Ding, Y. Yang, and X. Chen, "A new deep learning method based on AlexNet model and SSD model for tennis ball recognition," in *Proc. IEEE 10th Int. Workshop Comput. Intell. Appl. (IWCIA)*, Nov. 2017, pp. 159–164.

[43] U. Nepal and H. Eslamiat, "Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs," *Sensors*, vol. 22, no. 2, p. 464, Jan. 2022.

[44] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[45] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and F-score, with implication for evaluation," in *Proc. Eur. Conf. Inf. Retr.*, 2005, pp. 345–359.

[46] M. Li, Z. Zhang, L. Lei, X. Wang, and X. Guo, "Agricultural greenhouses detection in high-resolution satellite images based on convolutional neural networks: Comparison of faster R-CNN, YOLO v3 and SSD," *Sensors*, vol. 20, no. 17, p. 4938, Aug. 2020.

DURMUS OZDEMIR received the B.Sc. degree in computer engineering from the European University of Lefke, in 2004, Turkish Republic of Northern Cyprus, the M.S. degree in computer engineering from Karadeniz Technical University, in 2009, Trabzon, Turkey, and the Ph.D. degree from the Computer Education and Instructional Technologies Department, Ataturk University, in April 2015. He has been working as an Assistant Professor with the Computer Engineering Department, Kutahya Dumlupinar University, since May 2016. His research interests include instructional technology integration, digital transformation, e-learning, distance education policies, optimization, and deep learning.

MUSA SELMAN KUNDURACI was born in Konya, in 1990. He received the graduate degree in computer systems teacher from the Technical Education Faculty and in computer engineering from the Engineering Faculty, Konya Selcuk University, in 2013 and 2015, respectively, and the master's degree in computer engineering from Kutahya Dumlupınar University, in 2020. Since 2015, he has been working as a Teacher with the Technical High School, Ministry of National Education.