5-8-2023

Using Deep Learning Model to Identify Iron Chlorosis in Plants

Munir Majdalawieh
Zayed University, munir.majdalawieh@zu.ac.ae

Shafaq Khan
University of Windsor

Md. T. Islam
University of Windsor

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 April 2023, accepted 30 April 2023, date of publication 8 May 2023, date of current version 17 May 2023.
Digital Object Identifier 10.1109/ACCESS.2023.3273607

RESEARCH ARTICLE

Using Deep Learning Model to Identify Iron Chlorosis in Plants

MUNIR MAJDALAWIEH[1], SHAFAQ KHAN[2], AND MD. T. ISLAM[2]
[1]College of Technological Innovation, Zayed University, Dubai, United Arab Emirates
[2]School of Computer Science, University of Windsor, Windsor, ON N9B 3P4, Canada
Corresponding author: Munir Majdalawieh (munir.majdalawieh@zu.ac.ae)

This work was supported by Zayed University under Grant R21064.

ABSTRACT Iron deficiency in plants causes iron chlorosis which frequently occurs in soils that are alkaline (pH greater than 7.0) and that contain lime. This deficiency turns affected plant leaves to yellow, or with brown edges in advanced stages. The goal of this research is to use the deep learning model to identify a nutrient deficiency in plant leaves and perform soil analysis to identify the cause of the deficiency. Two pre-trained deep learning models, Single Shot Detector (SSD) MobileNet v2 and EfficientDet D0, are used to complete this task via transfer learning. This research also contrasts the architecture and performance of the models at each stage and freezes the models for future use. Classification accuracy ranged from 93% to 98% for the SSD Mobilenet v2 model. Although this model took less time to process, its accuracy level was lower. While the EfficientDet D0 model required more processing time, it provided very high classification accuracy for the photos, ranging from 87% to 98.4%. These findings lead to the conclusion that both models are useful for real-time classifications, however, the EfficientDet D0 model may perform significantly better.

INDEX TERMS CNN, iron chlorosis, plant disease, transfer learning.

I. INTRODUCTION

For early recognition of nutrient deficiency Aleksandrov [1] proposed an Artificial Neural Networks (ANN) approach based on chlorophyll fluorescence data. By exposing a leaf to the light of a certain wavelength and calculating the quantity of light emitted at longer wavelengths in response, the fluorescence yield may be calculated [2]. It would be difficult sometimes to extract features by following this method because not all molecules are fluorescent. An experienced farmer would know if there were any nutrient deficiencies in the plants by having a look at the leaves. With the goal to make a model that would have a farmer's eye, scientists came up with a methodology that involves feature extraction using digital image processing [3], [4]. A study [5] developed a system that would identify a nutrient deficiency in fruit plants using a dataset with digital images of fruits.

A plant should not have too little or too much of iron as it is a micronutrient. If the plant receives too much iron, it becomes iron toxic. Classic symptoms of iron toxicity are leaf discoloration (bronzing) and a stunted root system [4]. The ability to identify iron deficiency requires a dedicated eye. We need to address the question 'How to train an AI model which will be able to identify iron chlorosis?'. So, the idea itself is interesting to train a model that can successfully classify the outcome. The results of this study will help farmers with a lack of knowledge regarding various plant diseases. Remote farming emphasizes remote robot control and remote field monitoring in addition to the use of information and communications technology (ICT) to increase production efficiency and cut expenses [6]. This research will also have a direct influence on the business of remote farmers. Remote farmers that cannot visit farms on a daily basis can depend solely on digital photos of plants to monitor the business.

Often, iron deficiency is not a direct result of a lack of iron in the soil. A variety of soil conditions may affect the availability of iron to plants. Conditions such as high soil pH, high soil moisture, low temperatures, high phosphorus, and high concentrations of competing elements, such as zinc, calcium, and manganese, may reduce the availability of iron

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

to plants. Therefore, applying iron without considering soil conditions might not help in correcting the deficiency. Our test is important because before doing Alexandrov's test we need to be sure which nutrient is actually missing in the soil. For example, in remote farming, if iron deficiency can be predicted in the soil prior to the soil test, then a short-range soil test will be sufficient. Once the proposed model identifies the lack of iron in leaves, we can proceed with soil analysis to identify which conditions are the reasons for iron deficiency. The aim will be to work on datasets consisting of digital photos of plant leaves. The first iteration involves a training model to identify leaves. The following iterations involve training the model to distinguish between healthy and diseased leaves. Once this is done, the next iterations will involve training it to identify iron-deficient plants.

II. LITERATURE REVIEW

For years, scientists have been researching the standardization of laboratory techniques for soil analysis [7]. While they provided the most concrete insights on identifying nutrient deficiencies, Aleksandrov [1] proposed a new way to determine plant illness through the analysis of the rapid fluorescence of chlorophyll α. In their research, bean plants were grown on a nutrient medium and compared with another medium. The second medium leaks either nutrients like N, P, K, which are required in large quantities (more than a thousand mg/kg), or micronutrients like Fe, Zn which are needed in small quantities (less than a hundred mg/kg). The deficiency was evaluated by the stress response of plants estimated by leaves' "Photosynthetic Activity" which was estimated by analyzing chlorophyll fluorescence. For the purposes of this study, plants' immediate fluorescence signals were examined, and the signals served as input data for a neural network. However, in the training period, a Hoagland nutrient solution was needed to preprocess the sample plants. Also, the temperature was given along with a pH level of 5 and a photoperiod circle of 16/8 (Day/Night) in the training period. The plants were grown in dark glass pots. Solutions were supplied with oxygen by electrical pumps which had to be replaced every two days. If we take remote farming into consideration where some or all parameters are different, the training period can be over a week, therefore making the process difficult for the farmers.

Because the study of chlorophyll fluorescence discussed previously is nearly impractical for remote farming, we need the help of other branches of science to make it possible to diagnose the disease from a distance. Several research was already being conducted by scientists where the approach was to use artificial neural networks to sample images and outline plant diseases. Back in 2017, a reliable deep-learning-based method for the identification of illnesses and pests in tomato plants was proposed by Fuentes et al. [8]. They integrated each of the deep feature extractors— Visual Geometry Group (VGG) net and Residual Network (ResNet)—with region-based Convolutional Neural Networks (R-CNN), Region-based Fully Convolutional Networks (R-FCN) and Single Shot Detector (SSD) as deep learning meta-architectures. Their method was able to distinguish between nine distinct pests and illnesses. To identify and categorize grape leaf disease, Kothawale et al. [9] employed image processing techniques and the Support Vector Machine (SVM) classifier approach. On 90 photos of both healthy and damaged grape leaves, they trained their algorithm. They got an output result with an accuracy of 89.90%. Using photos of tomato leaves and two pre-trained deep learning algorithms (AlexNet and VGG16 net), Rangarajan et al. [10] classified tomato crop illnesses (6 diseases and a healthy class). For the VGG16 net and AlexNet, they obtained classification accuracy of 97.29 and 97.49 percent, respectively. In the same year, Horea and Mihai [11] provided a fresh, high-quality dataset with 55244 photos of fruits divided into 81 distinct groups. They also tested with deep neural network fruit recognition. Fruit frames were then retrieved from the video after the fruit's dataset had been rotated for filming. For various network configurations on the dataset, they got diverse results; they got a maximum training accuracy of 99.42 percent and a maximum test accuracy of 96.52 percent. In 2019, Francis and Deisy [12] published a paper demonstrating usage of convolutional neural network to detect and classify plant diseases. Four convolution layers each followed by a pooling layer were proposed where two fully connected dense layers and a sigmoid function were used. As a result, a previous overfitting problem was successfully removed, setting the dropout value to 0.2. The researchers managed to achieve an accuracy of 87%. However, they confessed that bringing change in the architecture might result in higher accuracy. Their model supported parallel processing, so a Graphics Processing Unit (GPU) can be used to increase the speed and accuracy. In 2020, a new family of object detectors called EfficientDet was introduced. Tan et al. [13], in his paper, proposed a weighted bi-directional feature pyramid network (BiFPN), which allows easy and fast multi-scale feature fusion. In the same year, Younis et al. [14] used Single Shot Detector (SSD) architecture combined with Mobilenet to detect objects and got an average precision of 99.7 percent. These two studies conducted in the same year require an in-depth comparison.

In general, there are two ways to train a deep neural network: either from scratch on the desired dataset, which is referred to as "training from scratch," or by adopting an existing network that has been pre-trained and retraining it on the desired dataset while making some modifications, which is referred to as "transfer learning." To appropriately restrict optimization, deep neural networks must be trained with millions of parameters and millions of samples [15]. This has led researchers to adapt pre-trained networks to the desired task domain with domain-specific data by means of transfer learning, which applies the knowledge learned from one problem to another, different but related problems [15]. Training these networks from scratch requires large amounts of training data and high computational resources. The transfer learning approach was selected to fine-tune the model

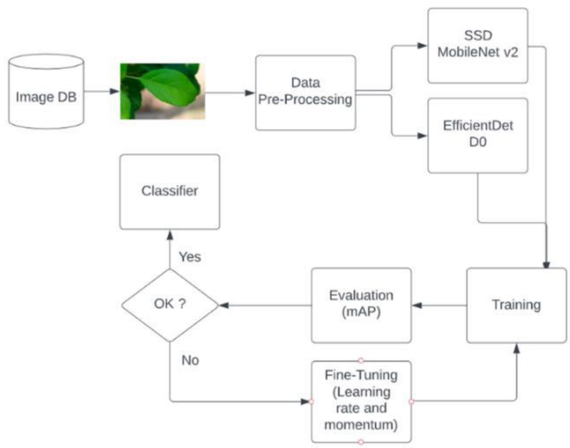

FIGURE 1. Proposed methodology.

parameters of diagnosing iron deficiency in plants since the dataset utilized for this job in this article is not a very huge dataset to train a deep neural network from scratch. So, two pre-trained COCO-Tensorflow object identification models [16] for deep learning, labeled SSD MobileNet v1 and EfficientDet is used. For the identification of plant nutrient deficiency considering these two pre-trained models, we will use the concept of transfer learning and retrain those models using datasets of leaves and compare speed with average precision. The goal of this research will be to identify nutrient deficiency and freeze the models for future use. Also, the focus will be to contrast the architecture and performance of the models at each stage.

III. METHODOLOGY

The suggested approach for identifying and classifying plant deficiencies accepts input in the form of pictures represented by a 3D matrix. Images are scaled, labeled, and then run through the model during the data pre-processing step. The model is trained using photos of leaves in various field conditions and two object identification models as base models: SSD Mobilenet v2 and EfficientDet D0. All the photos of healthy and deficient leaves are collected from the Google image database. The dataset is evaluated after training. Data validation occurs during the evaluation process. Both models' parameters are adjusted, training is repeated until it improves, and the evaluation is then sent to the classifier. For the leaves that are used for detection and classification, the model categorizes the labels. The proposed system methodology is shown in Figure 1.

IV. ARCHITECTURE

A. SSD MOBILENET V2

In 2016, Liu et al. [17] introduced this object-detecting method called Single Shot Detector (SSD). This is based on feed-forward CNN [18]. SSD combines the principles of various networks, including RPN in faster R-CNN, YOLO, and multi-scale CONV features, to produce quick detection speeds while preserving excellent detection quality [17], [19]. It produces a fixed-size collection of bounding boxes that scores for the presence of object class instances in the boxes. SSD combines single-stage regression prediction ideas and anchoring mechanisms by using VGG as the base feature extraction network. Figure 2 illustrates the network structure.

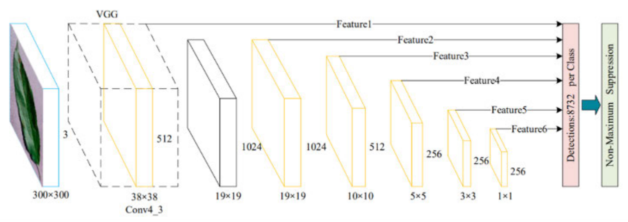

FIGURE 2. SSD network structure [20].

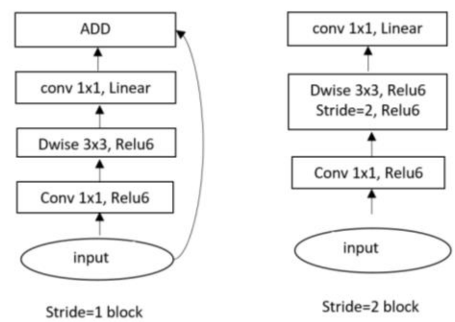

FIGURE 3. MobileNet v2 architecture [23].

The base network contains a truncated version of VGG16 which will be used for image classification. The advantage is that the operation speed is improved while maintaining the accuracy of detection.

A convolutional neural network design called MobileNetV2 [21] aims to function well on mobile devices. It is built on an inverted residual structure where the bottleneck layers are connected by residual connections. Lightweight depth-wise convolutions are used in the intermediate expansion layer as a source of non-linearity to filter features. The design of MobileNetV2 includes a 32-filter initial fully convolution layer as well as 19 additional bottleneck layers. Figure 3 provides an illustration of the architecture. It performs significantly better than many other well-known models. Following 1×1 convolutions, which aggregate these filters into a collection of output features, depth-wise separate convolutions first apply a single filter to each input to filter the input data. These depth-wise separable layers essentially do the same task as conventional convolution layers, but significantly more quickly and with a small variation. For this study, on the COCO (Common Objects in Context) dataset [22], a Single Shot Multibox (SSD) model with the MobileNet detector and classifier have been coupled. To perform direct regression prediction, the detector will adopt full convolution, ultimately improving the speed. However, in some cases, it might conduct a false detection. For this reason, we will

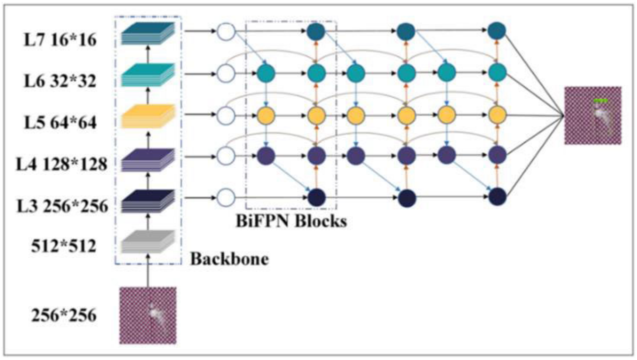

FIGURE 4. EfficientDet D0 architecture [26].

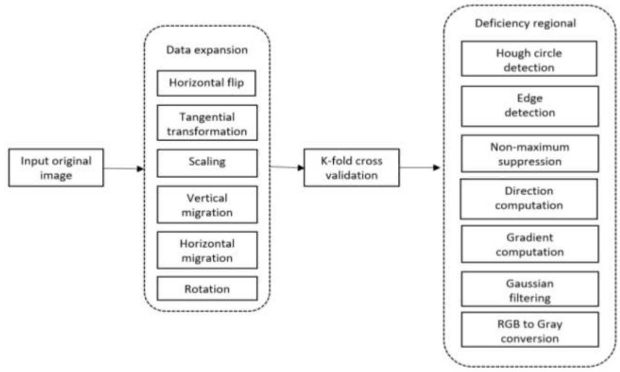

FIGURE 5. Image pre-processing flow [27].

adopt the second and comparatively latter network which is Efficientdet D0.

B. EFFICIENTDET D0

The second model used is a lightweight, scalable detection network known as EfficientDet [24]. The accuracy and temporal complexity of the model grows with model size from D0 to D7. The eight models may accommodate a variety of resource limitations. EfficientNet [25], which uses a lot of deep separable convolution to make the model lighter, is the foundation of EfficientDet. Additionally, the network's neck section makes use of a bidirectional weighted feature pyramid network (BiFPN). After the BiFPN, a class and box prediction network processes all the fused data to identify and indicate flaws. The network's structure is seen in Figure 4. Depending on the size of each model, the BiFPN, box, and class prediction networks will be iterated several times. The eight EfficientDet models are composed of different-sized backbones, BiFPNs, and prediction networks and can adapt to various computing resources. One of the base models that we'll adopt is the D0 architecture.

V. DATASET PREPARATION

A. IMAGE PRE-PROCESSING

According to Li et al. [27] image pre-processing serves two purposes. The first is to improve image quality to ensure sharp contrast and remove noise. Examples of this include using techniques like histogram equalization and grey scale transformation to enhance contrast and median filtering and adaptive filtering to remove noise. The second step is to segment the picture to make feature extraction easier, such as threshold, edge, and area segmentation. The flow of this procedure is shown in Figure 5.

To build the experiment dataset, more than 100 sample pictures were collected from the internet. Pictures were picked to train the model after several blurred, dusty, and hazy examples were removed. Images that contained obscured light were avoided because those would be reduced into smaller resolutions in the data augmentation section. By doing so, the system meets computational requirements. Some sample picture data that were used to train the model are shown in

FIGURE 6. Sample images of leaves (Healthy leaf on the left, Iron deficient leaves on the right).

TABLE 1. List of labels.

Model name	Number of classes/tables	Label Names
Model 1 for healthy leaves	1	healthy
N, D	1	deficient

Figure 6. An open-source graphical image annotation application named Labelimg was used to label and annotate the test pictures [28]. As XML files, the annotations were stored. After that, CSV files were created from the XML files. The tfrecord format was created from the CSV files. To train the CNN models, the tfrecord files were utilized as input data.

The tree leaves utilized in this experiment were labeled or categorized in Table 1, along with the leaf parts that were stained. A single class object was used to detect iron chlorosis, while a different class was used to identify the discolored areas.

B. DATA AUGMENTATION

As the inputs for the training, the photographs were reduced to 300×300. With little information loss, this size can lower the computational requirements for training. The dataset was then expanded to include more photos and make it ready for K-fold cross-validation. The following operations were targeted during data expansion: rotation, horizontal flip, vertical flip, scaling, and tangential transformation. By doing so, CNN would be able to pick up more invariant visual

FIGURE 7. A sample image of leaf after data augmentation.

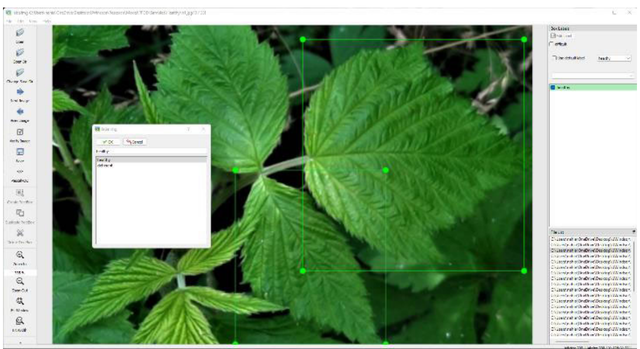

FIGURE 8. Capturing objects for 'healthy' class.

characteristics without over-fitting. Figure 7 shows nine versions of an image after applying data augmentation.

C. LABELING IMAGES

After importing and installing the dependencies, two classes were made. A package called LabelImg [28] is used to perform the labeling. On each training data, several object boxes were captured and included in either the "healthy" class or in the "deficient" class. Figure 8 exhibits two object boxes for the class "healthy".

It is possible to capture several boxes in one image. The package has the following dependencies: pyqt5 and lxml. After labeling each object box, a matching annotation file will be generated for each image. The models will look into specific folders for both the images and the annotations.

VI. EXPERIMENTAL RESULT AND DISCUSSION

To leverage a custom OD model fine-tuning or training a new computer vision model is needed. The most important dependency is TensorFlow model garden [29] which ensures

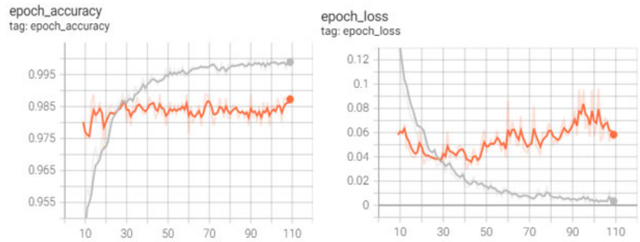

FIGURE 9. Accuracy and Loss graph of SSD Mobilenetv2.

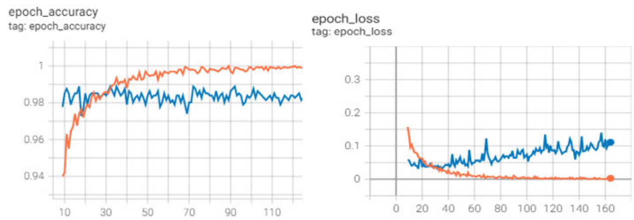

FIGURE 10. Accuracy and loss graph of EfficientDet D0.

TABLE 2. Speed and mean average precision score of several pretrained models.

Model name	Speed (ms)	mAP
SSD MobileNet V2 FPNLite 320x320	22	22.2
EfficientDet D0 512x512	39	33.6
CenterNet Resnet50 V1 FPN 512x512	27	31.2
CenterNet MobileNetV2 FPN 512x512	6	23.4
CenterNet Resnet50 V2 512x512	27	29.5
CenterNet MobileNetV2 FPN 512x512	6	23.4
Faster R-CNN ResNet50 V1 640x640	53	29.3

State-Of-The-Art (SOTA) models are invoked. GPU was used instead of CPU in the training of the model. The first model, SSD Mobilenetv2 was taking approximately 22 milliseconds to process each object box. Before fine-tuning, 10 epochs were run after adding a classifier on top of the convolution. The first 10 epochs were conducted on the top-level classifier. Doing so, a 93.6% accuracy was obtained. The rest of the iterations were conducted while weights of the pre-trained network were updated. The model was training faster but it was taking more steps to increase a little bit of accuracy. There are 154 layers in the base model and the top 54 layers were fine-tuned. The final accuracy was approximately 98%. Figure 9 illustrates epoch accuracy and loss after 9821 iterations where the gray line indicates the train, and the orange line indicates validation.

The second model, efficientdet D0 took relatively more time to train than the previous model. Despite completing the first 10 epochs with an accuracy of 87.2% which is less than the previous model, it obtained slightly higher accuracy in the

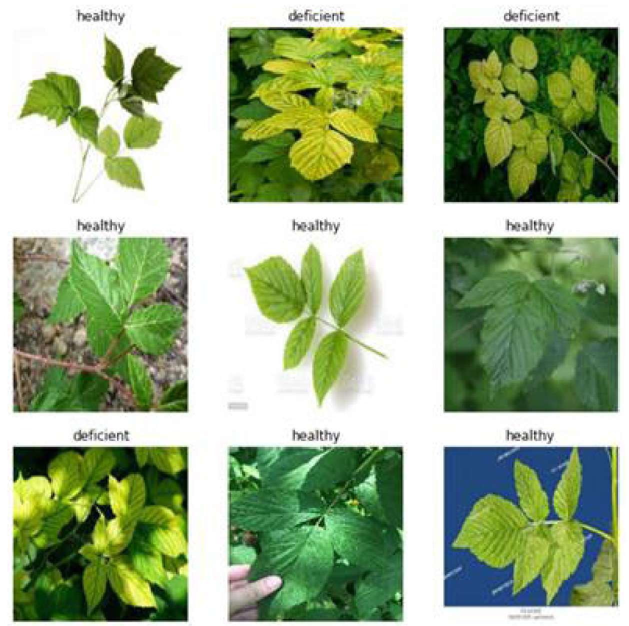

FIGURE 11. **Result of detection (Model: SSD Mobilenetv2).**

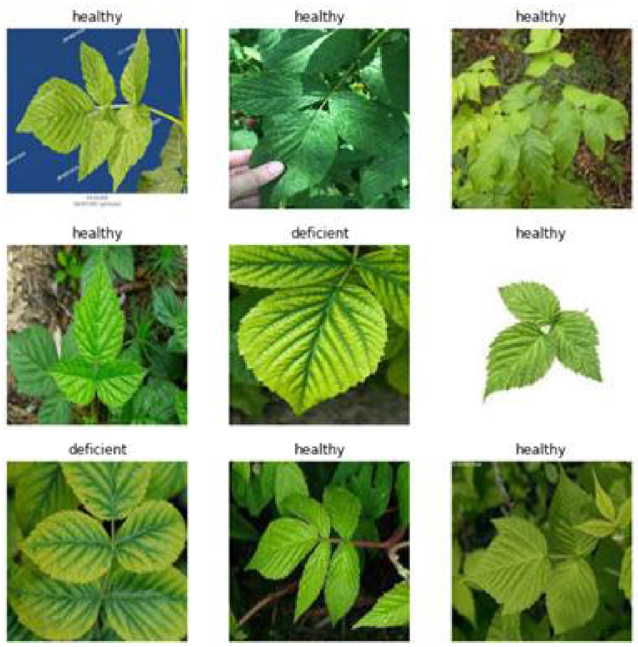

FIGURE 12. **Result of detection (Model: EfficientDet D0).**

validation phase (98.4%) and reduced loss ratio. The number of layers in the base convolution was 237 and the top 37 layers were fine-tuned. Figure 10 shows the accuracy and loss graph where the orange line indicates the training phase and the blue line is the validation phase.

After the training is done, the mean average precision is calculated for both models. The number of true positives is divided by the summation of true positives and false positives to calculate the precision. True positive means the model is successfully detecting the leaves whereas the false positive refers to the identification of wrong objects. It shows the ratio of correct and wrong detection. SSD mobilenetv2 obtained mAP score of 22 and a speed being 22.2ms. Figure 11 shows SSD mobilenetv2 detecting nine sample images.

When the same amount of image was provided to the second model, Efficientdet D0, the processing time was slightly longer, 39ms. The mAP score, on the other hand, was also better, being 33.6. This proves if a model detects an object faster, it will have lower accuracy and vice versa. Figure 12 illustrates nine more images successfully detected by the model. In Table 2, the properties of some pre-trained models along with discussed models are illustrated. Efficientdet D0 has better precision than several popular models.

VII. CONCLUSION AND FUTURE WORKS

Real-time iron deficiency detection is performed with images that were collected from the internet. The work in this paper was conducted on two pre-trained models- SSD MobileNetv2 and EfficientDet D0 using Tensorflow object detection API. After training the models on the same datasets, they had speeds of 22.2ms and 39ms respectively. Despite the SSD Mobilenetv2 being a slightly faster convolution, it has a lower precision (COCO mAP: 22). On the other hand, despite taking more time on training, EfficientDet D0 was more precise in detecting the class of image. It is concluded that both the models are usable in the real-world scenario but EfficientDet D0 will have leverage over SSD mobilenetv2 because of having a higher COCO mAP score.

The pre-trained models used were box-based detection systems. But a leaf is commonly round or oval. So, when the object boxes were prepared, 30-35% of the space of the box was occupied by the background. So, the chances of having background noise were high. There are some other pre-trained models which are based on masks and key points. Using those convolutions might result in higher accuracy in the training phase. Also, the images of the leaves were generalized, meaning there are 2 classes implemented in total in this research. Future work may include categorizing all plant species. Then the number of total classes become two times higher, one each for healthy and deficient leaf. Also, we have used GPU for all the tasks executed. A TPU (Tensor Processing Unit) may be used, and performance may be compared with that of a GPU in the future for this specific dataset.

REFERENCES

[1] V. Aleksandrov, "Identification of nutrient deficiency in plants by artificial intelligence," *Acta Physiologiae Plantarum*, vol. 44, no. 3, p. 29, 2022.
[2] M. Becker and F. Asch, "Iron toxicity in rice—Conditions and management concepts," *J. Plant Nutrition Soil Sci.*, vol. 168, no. 4, pp. 558–573, Aug. 2005.
[3] Y. Adhitya, S. W. Prakosa, M. Köppen, and J.-S. Leu, "Feature extraction for cocoa bean digital image classification prediction for smart farming application," *Agronomy*, vol. 10, no. 11, p. 1642, Oct. 2020, Accessed: Dec. 21, 2022. [Online]. Available: https://www.mdpi.com/2073-4395/10/11/1642
[4] H. Pang, Z. Zheng, T. Zhen, and A. Sharma, "Smart farming," *Int. J. Agricult. Environ. Inf. Syst.*, vol. 12, no. 1, pp. 55–67, 2021.
[5] A. K. D. Yogesh, R. Arora, and A. Rocha, "Deep learning based automated fruit nutrients deficiency recognition system," *J. Inf. Sci. Eng.*, vol. 37, no. 5, pp. 1153–1164, 2021.

[6] H. Madokoro, S. Yamamoto, Y. Nishimura, S. Nix, H. Woo, and K. Sato, "Prototype development of small mobile robots for mallard navigation in paddy fields: Toward realizing remote farming," *Robotics*, vol. 10, no. 2, p. 63, Apr. 2021.
[7] J. H. Dane. (May 27, 2020). *Methods of Soil Analysis, Part 4: Physical Methods*. Google Books. Accessed: Dec. 21, 2022. [Online]. Available: https://books.google.com/books/about/Methods_of_Soil_Analysis_Part_4.html?id=OYLWDwAAQBAJ
[8] F. Alvaro, Y. Sook, K. Sang, and P. Dong, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors*, vol. 17, no. 9, p. 2022, 2017.
[9] S. S. Kothawale, S. R. Barbade, and P. P. Mirajkar, "Grape leaf disease detection using SVM classifier," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 6, no. 4, pp. 3288-3294, 2018.
[10] A. K. Rangarajan, R. Purushothaman, and A. Ramesh, "Tomato crop disease classification using pre-trained deep learning algorithm," in *Proc. Conf. Int. Conf. Robot. Smart Manuf. (RoSMa)*, 2018, pp. 47–1040.
[11] M. Horea and O. Mihai, "Fruit recognition from images using deep learning," *Acta Universitatis Sapientiae, Inform.*, vol. 10, no. 1, pp. 26–42, 2018.
[12] M. Francis and C. Deisy, "Disease detection and classification in agricultural plants using convolutional neural networks—A visual understanding," in *Proc. 6th Int. Conf. Signal Process. Integr. Netw. (SPIN)*, Mar. 2019, pp. 1063–1068.
[13] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1–10.
[14] A. Younis, L. Shixin, S. Jn, and Z. Hai, "Real-time object detection using pre-trained deep learning models MobileNet-SSD," in *Proc. 6th Int. Conf. Comput. Data Eng.*, Jan. 2020, pp. 44–48.
[15] M. Mehdipour-Ghazia, B. A. Yanikoglu, and E. Aptoula, "Plant identification using deep neural networks via optimization of transfer learning parameters," *Neurocomputing*, vol. 235, pp. 228–235, Apr. 2017.
[16] Tensorflow. (May 7, 2021). Models/tf2_detection_zoo.MD at Master? Tensorflow/Models. GitHub. Accessed: Dec. 21, 2022. [Online]. Available: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md
[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Comput. Vis. (ECCV)*, 2016, pp. 21–37, 2016.
[18] G. Bebis and M. Georgiopoulos, "Feed-forward neural networks," *IEEE Potentials*, vol. 13, no. 4, pp. 27–31, Dec. 1994.
[19] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *Int. J. Comp. Vis.*, vol. 128, no. 2, pp. 261–318, 2019.
[20] J. Wang, L. Yu, J. Yang, and H. Dong, "DBA_SSD: A novel end-to-end object detection algorithm applied to plant disease detection," *Information*, vol. 12, no. 11, p. 474, Nov. 2021.
[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1–11.
[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Comput. Vis. (ECCV)*, 2014, pp. 740–755, 2014.
[23] P. D. B. QoChuk. Mobilenetv2 Architecture. OpenGenus IQ: Computing Expertise & Legacy. Accessed: Dec. 21, 2022. [Online]. Available: https://iq.opengenus.org/mobilenetv2-architecture/
[24] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10781–10790.
[25] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," 2019, *arXiv:1905.11946*.
[26] S. Song, J. Jing, Y. Huang, and M. Shi, "EfficientDet for fabric defect detection based on edge computing," *J. Engineered Fibers Fabrics*, vol. 16, Jan. 2021, Art. no. 155892502110083.
[27] Y. Li, H. Huang, Q. Xie, L. Yao, and Q. Chen, "Research on a surface defect detection algorithm based on MobileNet-SSD," *Appl. Sci.*, vol. 8, no. 9, p. 1678, Sep. 2018.
[28] D. Tzutalin. *LabelImg*. (2015). [Online]. Available: https://github.com/heartexlabs/labelImg
[29] H. Yu. (2020). *TensorFlow Model Garden*. [Online]. Available: https://github.com/tensorflow/models

MUNIR MAJDALAWIEH received the MSc. degree in computer science and applied math from Northeastern University, Boston, MA, USA, and the Ph.D. degree in IT and the E.M.B.A. degree from George Mason University, Fairfax, VA, USA.

Since August 2014, he has been an Associate Professor with Zayed University (ZU), United Arab Emirates, where he has been the Head of the Information Systems and Technology Management Department, College of Technological Innovation, since 2017. He was appointed to hold the following administrative positions at ZU: an Advisor of the Provost on Community Engagement and Outreach, from January 2017 to May 2018; a ZU/SAP Dual Study Coordinator, Since January 2017; an Assistant Dean for students, from August 2015 to December 2015; and an Enterprise Computing/Systems Coordinator, from August 2012 to June 2015. Prior to joining ZU in 2012, he was with the American University of Sharjah (AUS), United Arab Emirates, for six years. He was with Booz Allen Hamilton, Hewlett Packard, Compaq Computer Corporation, and Digital Equipment Corporation, USA, for more than 22 years. He has published many peer-reviewed papers in international journals and conference proceedings covering topics related to digital transformation, business process modeling, data science, AI, IT audit and control, supply chain, enterprise information management/cybersecurity, and other related topics.

SHAFAQ KHAN received the M.Sc. degree in computing and information systems from the University of Hull, U.K., and the Ph.D. degree in computer science from Salford University, U.K.

She is currently an Assistant Professor with the School of Computer Science, University of Windsor, Canada. Prior to this, she was with the College of Technological Innovations, Zayed University, United Arab Emirates, and the College of Information Technology, University of Dubai, United Arab Emirates. Since 2014, she has published more than 20 research papers in peer-reviewed journals and conference publications. Her research interests include artificial intelligence, data analytics, digital transformations, cloud computing, blockchain technology, and IT governance. She is trained and certified in innovation and entrepreneurship education from Stanford University, USA. She is a fellow of the Higher Education Academy, U.K., an IBM-Certified Business and Artificial Intelligence Analyst, a Certified Ethical Hacker (CEH), a SAP-Certified Business Associate, a Certified Project Management Professional (PMP), and a Cloud Infrastructure and Services (CIS) Associate. She has received several awards, including excellence in teaching and best faculty awards at the university level.

MD. T. ISLAM received the B.Sc. degree in computer science and engineering from the Bangladesh Army International University of Science and Technology, in April 2019, and the master's degree in applied computing from the University of Windsor, in August 2022.

From June 2019 to December 2020, he was a Web Developer with NurTech and a Research Assistant with Zayed University and the University of Windsor. His research interests include the development of machine learning models, microservices, data analytics, and requirement engineering. His awards and honors include the Dean's List Award, in 2015, as the Finalist, and the Best Mentor of BAIUST Computer Club, in 2018.