DOI: https://doi.org/10.33103/uot.ijccce.21.4.2

Detection and Classification of Leaf Disease Using Deep Learning for a Greenhouses' Robot

Shahad Ahmed Sabeeh[1], Saman Hameed Ameen[2]
[1,2]*Computer Engineering Department, University of Technology, Baghdad, Iraq*
[1]*ce.19.10@grad.uotechnology.edu.iq*, [2]*120108@uotechnology.edu.iq*

***Abstract*—** *Plant diseases are a severe threat to the environment, economy, and health. Early disease identification remains a challenging task in Iraq due to the scarce of the necessary resources and infrastructure. This paper uses various deep learning algorithms to detect different diseases on plant leaves and detect healthy ones, using an RGB camera as a crucial part of our real-time autonomous greenhouses' robot along with using two datasets, plant-village and cotton dataset, to investigate the best convolutional neural network architecture. The first dataset contained 10,190 images from the plant-village open datasets; it includes four crops with ten distinct classes of diseased and healthy leaves. Moreover, the cotton dataset contained 2,204 images for training and 106 images for testing; it has four classes of diseased and healthy plants and leaves. Different network architectures were tested in this paper for the best suitable lightweight architecture for our mobile robot. Results show that the best performance is 99.908% which achieved by the VGG16 network. The highest accuracy of VGG16 obtained in our research makes it the best tool for our autonomous plant disease detection robot.*

Index Terms *— Convolutional neural networks architectures, Computer vision, Deep learning, Leaf disease detection and classification, Precision agriculture.*

I. INTRODUCTION

Robotics have been widely adopted in the field of agriculture in greenhouses. A greenhouse can revolutionize long-term crop production and food security in areas where food scarcity is a significant issue by artificially providing suitable conditions for crop growth. It enhances crop production compared to farms [1]. Moreover, in recent years, food security has been considered a significant issue. According to the FAO (Food and Agricultural Organization of the United Nations), food growth should be raised by 70% by 2050 to meet the growing global population's food requirements [2]. Global famines are a distinct possibility if effective and accurate methods do not improve plant production; moreover, crop diseases continue to pose the most significant threat to food security. Although identifying and diagnosing plant diseases may be time-consuming, it significantly reduces product losses [3]. As a consequence, expenditure on agricultural robotics research has increased at an exponential rate [4].

Precision agriculture is farming more healthy quantity growing crops utilizing technology; some information technology procedures get used for precision agriculture applications, like accurate pesticide administration. Despite all the technological advances in precision agriculture, it is still not as good as a human expert, but humans are prone to danger

Received 4/5/2021; Accepted 28l8l2021

DOI: https://doi.org/10.33103/uot.ijccce.21.4.2

and illnesses due to continuous exposure to farms and pesticides. Therefore, our proposed farming tools must process and make inferences from the acquired data professionally. Unlike conventional image processing techniques, Deep learning (DL) is a subset of *machine learning* algorithms that progressively uses multiple layers to extract higher-level features; it prevents labor-intensive feature engineering and threshold segmentation used in previous plant disease detection[5]-[6]. It is the most convenient in leaf disease detection by preprocessing the image of the infected plant leaf and fitting it into the disease detection network model.

Computer vision and Artificial Intelligence research have facilitated raw image automatic plant disease detection [7]. They guarantee real-time execution, robustness, adaptability, and scalability [8]. Traditionally, classification approaches depended on semantic features [9]. However, the characteristic of DL is the ability to obtain attributes from image patterns automatically facilitated the use of convolutional neural networks in the field of computer vision and pattern recognition systems, e.g., detecting handwritten eastern Arabic numbers [10].

As shown in *Fig.* 1[11], the convolutional neural network (CNN) consists of four layers: an input layer, a convolutional and pooling layer, fully connected layers, and an output layer. Convolutional layers hold the results of filter or kernel convolution with the previous. These filters, or kernels, are composed of learnable weights and biases. Pooling layers reduces the number of pixels of an image (downsampling) while not losing important information. Max pooling is one method that retains the most representative pixel while ignoring the least representative ones. Output from successive convolution and pooling layers is flattened to a single vector to input the next layer. The first fully connected layer inputs from feature analysis and applies weights to predict the correct label, and the fully connected output layer gives the final probability for each label [12].

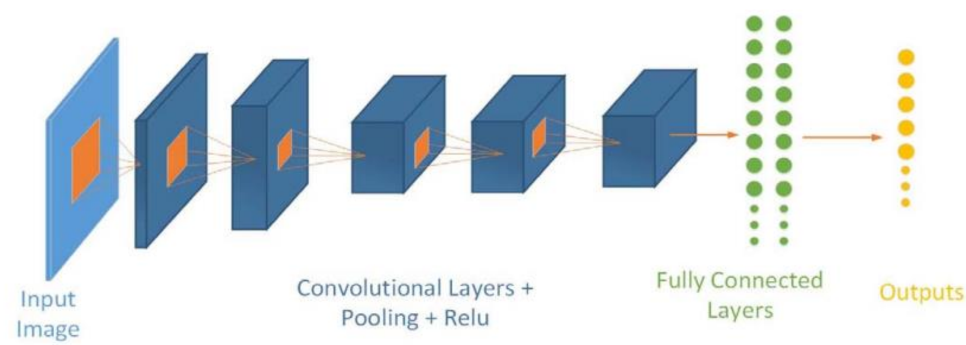

FIG. 1. CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE [11]

Transfer Learning (TL) represents the idea of overcoming the isolated learning paradigm and utilizing knowledge acquired for one task to solve related ones. It achieves higher accuracy with less training time compared to classical methods neural networks [13]. In this study, previous studies are presented, additionally using two datasets, plant-Village dataset [14] and

Received 4/5/2021; Accepted 28l8l2021

DOI: https://doi.org/10.33103/uot.ijccce.21.4.2

cotton dataset [15], four pre-trained state-of-the-art CNN architectures are trained and fine-tuned using TL in two methods (shallow TL and deep TL) to find a high accuracy network model with less RAM size due to our usage of the Jetson Nano in our robot.

The remainder of the paper is organized as follows: Section II provides related works, in Section III materials and methods are described, Section IV presents the study results and discussion. Section V contains the conclusion.

II. RELATED WORK

Plant disease detection and identification are a tedious task that requires human intellect and experience to complete correctly. Nowadays, various self-automated plant disease classification techniques are used, replacing parts of human's agricultural jobs, including machine learning and basic engineering, which befit fewer classes and are more based on specific environments. As a consequence, small environmental changes can result in significant accuracy drops. Recently, CNN architectures are used to improve plant disease classification and have made substantial progress toward resolving the existing challenges confronting researchers in this field.

In [16], E. Fujita and Y. Kawasaki suggested a classifier for cucumber diseases based on a network comprised of four convolutional layers with max-pooling layers in-between. Local reaction normalization was used for image pre-processing. For preparation and validation, they used two datasets. These databases provide seven distinct categories of diseases as well as one healthy class. The first dataset comprises 7320 images of leaves taken in ideal conditions. The second dataset includes 7520 images obtained under optimal and insufficient lighting environments to maximize the likelihood of recognizing diseases accurately. Finally, the proposed method achieved an accuracy of 82.3%.

Furthermore, A. Dhakal and P. S. Shakya [17] suggested developing a classifier for tomato leaves. The dataset was assembled using data from the plant-village dataset and a variety of online sources. The images in the dataset depict leaf samples infected with bacterial spot, yellow leaf curl virus, late blight, and healthy leaf. Resize, zoom, shear, and rotate augmentation methods used images to maximize the dataset's scale. The suggested CNN, which consists of four convolutional and pooling layers, was constructed entirely from scratch and obtained an overall accuracy of 98.59% in identifying plant disease.

S. P. Mohanty [18] collected a dataset of 966 images to identify nine classes of rice diseases. They used image augmentation strategies such as grey transformation, image grey transformation, and image filtering and relied on transfer learning architectures VGGNet, ImageNet, and Inception. The total accuracy obtained is 92%. On the other hand, other studies used transfer learning, such as [19] and [20], in [19] I. Z. Mukti and developed a TL network using the ResNet50 model. The use of shallow Fine-Tuning has improved the identification accuracy. This work used a dataset that included images of 38 different diseases as well as

Received 4/5/2021; Accepted 28l8l2021

DOI: https://doi.org/10.33103/uot.ijccce.21.4.2

healthy plants. The dataset increased because of the rotation, width shift, height shift, shear range, and horizontal flip data augmentation. The model achieved an accuracy of 99.80% using ResNet50, 94.96% using VGG16, 91.75% using VGG19, and 83.66% using AlexNet. While in [20], V. Kumar recommended using the ResNet34 model using TL and utilizing images from the new plant diseases dataset, which contains 15200 images labelled with 38 distinct groups and covering 14 different crops. Rotation and normalization image pre-processing techniques are implemented to improve data quality to support feature extraction easily. Two metrics were used to evaluate accuracy and precision using SVM, decision trees, logistic regression, and K-NN algorithms along with ResNet34. Compared with the mentioned models, ResNet34 achieved 99.40% accuracy and 96.51% precision. Other studies reviewed existing strategies for deep learning architectures used for the identification of plant diseases.

Other studies reviewed existing strategies for deep learning architectures used for the identification of plant diseases. J. Wäldchen and P. Mäder [21] recently released a study on identifying plant diseases using computer vision techniques. Their analysis included nearly 120 works and a comprehensive overview of datasets.

The majority of the research presented in this section focused on improving disease classification accuracy using a small number of classes, as stated in [16]-[17], which used customized CNN architectures, and [18]-[20], which applied TL to existing deep CNN architectures using datasets of single leaves without accounting for background changes or using just shallow TL in [19]. Additionally, they do not indicate the hardware or framework on which the training and testing of the CNN are performed. This study aims to detect leaves with complicated backgrounds in greenhouses accurately and identify diseases that are seen. Two datasets were used in this work to classify four crops with ten classifications of diseased and healthy leaves for the first dataset; moreover, four classes of healthy and diseased cotton plant and leaves are detected using two deep learning techniques, shallow TL and deep TL, which achieves a better level of accuracy. However, further study on the local datasets is necessary owing to their scarcity.

III. MATERIALS and METHODS

This work is part of designing and implementing an autonomous robot system to monitor plants and identify diseases. This robot does not only detect diseases but also sprays pesticides locally to treat the infected leaves. High accuracy in navigation and detection is obtained through the use of deep learning algorithms. Moreover, this is an environment-friendly approach and will reduce exposure to pesticides for farm workers and crops. The outcome of using such a robot will reduce production costs and will increase sustainability. As shown in *Fig*. 2, the flowchart of our suggested approach for detecting and classifying plant diseases, a dataset is used for training and testing. The first dataset including four crops and ten classifications of sick and healthy leaf images was obtained from the plant-village [14] collection; this is a reliable public dataset that has been utilized in many prior studies. Otherwise, gathering and labeling a new dataset is a costly job that needs the expertise of an academic; the dataset used is detailed in Table I. Moreover, a background class [22] was added

Received 4/5/2021; Accepted 28l8l2021

DOI: https://doi.org/10.33103/uot.ijccce.21.4.2

to distinguish leaves in the greenhouse complex environment. or utilizing a cotton dataset [15] including four classes of cotton plant and leaves, both diseased and healthy, plus a background class [22]; this dataset is detailed in Table II.

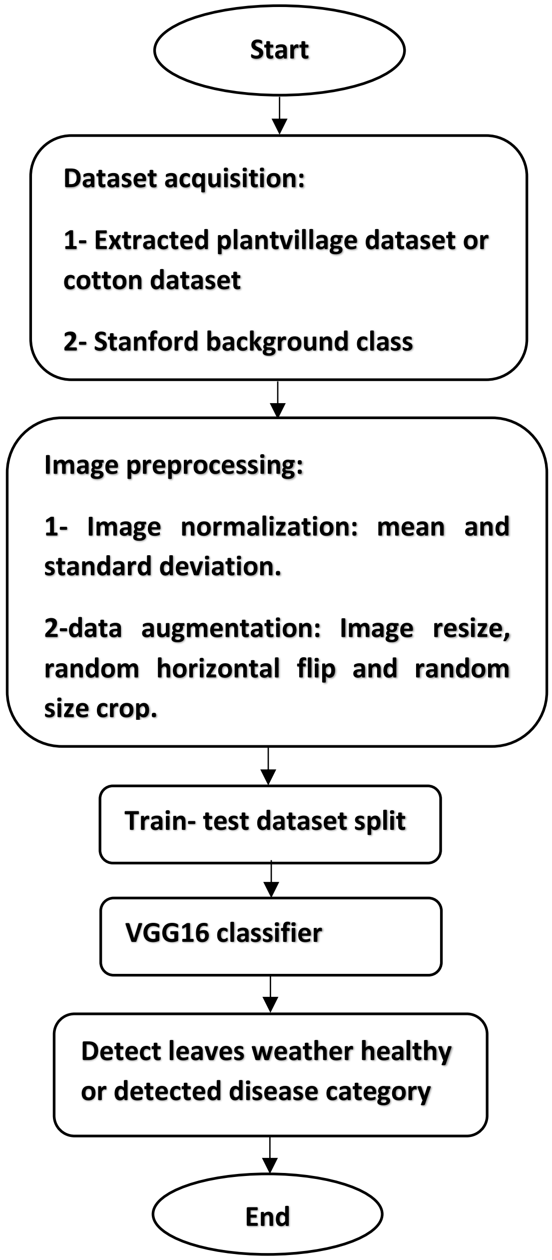

FIG. 2. THE FLOWCHART OF PLANT DISEASE DETECTION PROCESS.

Received 4/5/2021; Accepted 28l8l2021

DOI: https://doi.org/10.33103/uot.ijccce.21.4.2

TABLE. I. DERIVED PLANTVILLAGE DATASET DETAILS.

No	Name	No. of Images
1	Cherry Healthy	854
2	Cherry Powdery Mildew	1052
3	Grape Healthy	423
4	Grape Black Rot	1180
5	Grape Black Measles (Esca)	1383
6	Grape Leaf Blight	1076
7	Peach Healthy	360
8	Peach Bacterial Spot	2297
9	Strawberry Healthy	456
10	Strawberry Leaf Scorch	1109

TABLE. 2. COTTON DATASET DETAILS.

No	Name	No. of Images
1	Diseased Cotton Plant	921
2	Fresh Cotton Plant	514
3	Diseased Cotton Leaf	356
4	Fresh Cotton Leaf	519

Subsequently, image preprocessing is performed in the following steps: image resize, image normalization, and data augmentation. Images were resized to 224*224 as it is the same input size the chosen pre-trained models were trained. For image normalization to mean and standard deviation, values of RGB image pixels were computed using the corresponding values mean= [0.485, 0.456, 0.406], standard deviation= [0.229, 0.224, 0.225] to speed up training and reduce cost function. Moreover, the normalization ensures that each input parameter, pixel, has similar data distribution, making convergence faster while training a network [23]. Using the ImageNet pre-trained models with their own mean and standard deviation eases the optimization process while using ImageNet-like images similar to our dataset. The datasets were distributed in an 80/20 ratio rule for training and evaluation consisting of the following details (Cherry Healthy, Cherry Powdery Mildew, Grape Healthy, Grape Black Rot, Grape Black Measles, Grape Leaf Blight, Peach Healthy, Peach Bacterial Spot, Strawberry Healthy, Strawberry Leaf Scorch), shown in Table I, for the first dataset and the following details (Diseased Cotton Plant, Fresh Cotton Plant, Diseased Cotton Leaf) for the second dataset are shown in Table II. We chose the ratio above since validating models with a small number of hyperparameters does not need much data because doing so is computationally expensive.

It was selected four pre-trained models based on their high accuracy through training on ImageNet shown in Table III [24], ImageNet is a dataset comprised of 14 million images created primarily to classify 1000 classes created by academics. Implementing transfer learning on pre-trained models facilitates complicated classification tasks requiring large datasets and reduced training time since these pre-trained models become efficiently tuned to similar behavior. The following CNN models (Inceptionnet-v3 [25], ResNet50 [26], Squeezenet1-1 [27], and VGG16 [28]) were retrained using a backpropagation algorithm on GPU using CUDA to reduce training time by Pytorch framework, characteristics of the used machine

Received 4/5/2021; Accepted 28l8l2021

Iraqi Journal of Computers, Communications, Control & Systems Engineering (IJCCCE), Vol. 21, No. 4, December 2021

DOI: https://doi.org/10.33103/uot.ijccce.21.4.2

shown in Table IV. Pytorch is a programming framework based on python, used to build and code deep learning models. It also achieves on average high-performance using GPU and CUDA compared to other frameworks, which results in less training time. TL is often represented in computer vision via the use of pre-trained models. A pre-trained model has been trained on an extensive benchmark dataset to solve a similar issue to the one we are trying to solve. Inceptionnet-v3, ResNet50, Squeezenet1-1, and VGG16 were trained to utilize TL in two methods. The first method is to fine-tune the network's final layer (shallow training) while the previous layers work as feature extractors; the second method is to fine-tune each layer (deep training) in the network and apply backpropagation from the pertained layers. Training a neural network involves solving an optimization problem; this entails decreasing the loss function iteratively by changing the trainable parameters. When a model is trained for the first time, it is assigned an arbitrary set of weights to each connection between neurons, and these weights are continuously changed before they achieve their optimum values through training. Optimizing weights varies depending on the optimization algorithm used; in our work, the stochastic gradient descent (SGD) with the momentum 0.9 was used since it achieves high performance with long a training time [29].

As shown in *Fig*. 3, VGG16 architecture, CNN improved architecture of the AlexNet network, contains 16 weight layers in which 13 are convolution layers and 3 fully connected layers stacked together designed for image classification. The model achieves 92.7% top-5 accuracy in ImageNet. The images, of size (224*224*3) RGB collected after the normalization step, are used to input layers. VGG is the idea of deeper networks, and with smaller filters, it uses a 3*3 filter which is the smallest convolutional filter with an increased number of layers from eight layers in AlexNet. The first two layers are convolutional layers with 3*3 filters, whilst the first two layers utilize 64 filters, resulting in a volume of 224*224*64 due to the use of the same convolutions. The filters are often three by three with a stride of one. Following that, a pooling layer with a max pool of 2*2 size and stride two was used to reduce the volume's height and width from 224*224*64 to 112*112*64. Following that, two additional convolution layers with 128 filters are added. As a consequence, a new dimension of 112*112*128 is formed. The amount is decreased to 56*56*128 after the pooling layer is used. Two additional convolution layers of 256 filters are inserted, accompanied by a down sampling layer that shrinks the image to 28*28*256. A max-pool layer separates two additional stacks, each with three convolution layers. Following the final pooling layer, the 7*7*512 volume is flattened into a Fully Connected (FC) layer with 4096 channels and 1000 groups of SoftMax output [28]. Cross entropy loss was used as it is a measure of the difference between two probability distributions for a given random variable or set of events; the goal is to minimize the loss, i.e., the smaller the loss, the better the model. The calculation formula is:

$$Loss(x, class) = -\log\left(\frac{\exp(x[class])}{\sum_i \exp(x[i])}\right) = -x[class] + \log(\textstyle\sum_i \exp(x[i])). \quad (1)$$

From the above equation, x is the input, and the output of the last layer of the network and class is the category index. Also, accuracy is calculated through the formula:

$$\text{Accuracy} = \text{(correctly predicted class / total testing class)} \times 100\%. \quad (2)$$

Received 4/5/2021; Accepted 28l8l2021

DOI: https://doi.org/10.33103/uot.ijccce.21.4.2

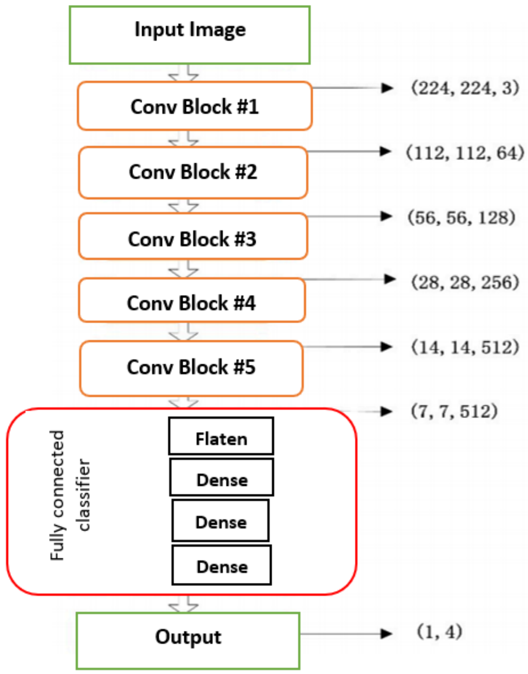

FIG. 3. VGG16 ARCHITECTUR [30].

For the implementation of the disease detection process in greenhouses, the VGG16 model was saved after training. Images acquired from greenhouses with 720*720 are resized to 224*224 because images should be uniform before being fed to a CNN model. Preprocessing images after deploying a model is essential to perform steps that simplify and improve the accuracy of the applied method [31]. The images were captured using a Logitechc270 RGB camera, which is shown in *Fig*. 4 the camera is mounted on a robotic arm as an integral component of the robot shown in *Fig*. 5. Four, the robot's primary goal is to identify plant diseases and spray pesticides exclusively on diseased leaves. This camera records frames for analysis by the Jetson Nano embedded system-on-module shown in *Fig*. 6, it is a small, powerful computer with 4 GB of RAM that allows the concurrent execution of several neural networks for image classification, object identification, segmentation, and voice processing while using as little as 5 watts. It also includes the Linux operating system, NVIDIA CUDA®, cuDNN, and TensorRT^{TM} software libraries for deep learning and GPU computing, which assist computer vision applications.

TABLE 3. PRETRAINED MODELS AND THEIR PERFORMANCE.[24]

No	Model	Year	No of parameters	Top-1 Accuracy
1	Inceptionnet-v3	2015	24 million	78.8%
2	ResNet50	2015	25 million	77.15%
3	squeezenet1-1	2016	1.2 million	57.5%
4	VGG16	2014	138 million	74.5%

Received 4/5/2021; Accepted 28l8l2021

DOI: https://doi.org/10.33103/uot.ijccce.21.4.2

TABLE. 4. COMPUTER ATTRIBUTES.

No	Computer System	Attributes
1	Operating System	Windows 10 64 bits
2	Processor (CPU)	Intel(R) Core (TM) i7-10750H CPU @ 2.59 GHz
3	Graphics (GPU)	Nvidia GTX 1660Ti 6GB
4	Memory	16.0 GB

FIG. 4. LOGITECH C270 CAMERA USED IN RESEARCH.

FIG. 5. THE ROBOT USED IN THE RESEARCH.

Received 4/5/2021; Accepted 28l8l2021

DOI: https://doi.org/10.33103/uot.ijccce.21.4.2

FIG. 6. JETSON NANO DEVELOPER KIT BOARD.

IV. RESEARCH RESULTS AND DISCUSSION

Our findings on training TL CNN architectures hint that fine-tuning all layers in a model (Deep training) achieves higher accuracy than the shallow training counterpart. As Table V elaborates, VGG16 has achieved the highest accuracy among models with 99.9% accuracy with a comparable result of 99.6% to Squeezenet1.1. Whereas in shallow training, Squeezenet1.1 accomplished an accuracy of 98.5%, with a similar accuracy of 97.8% for Resnet50. These results tie well with previous studies wherein TL training consumes less time than training networks from scratch [23]. In terms of training time, shallow training consumes less time as in Squeezenet1.1 that consumed around 10 minutes on our dataset due to its low number of trainable parameters, than Resnet50 that consumed a bit longer of around 21 minutes. We also observe that loss in deep is lower than shallow leaning as Resnet50 has a loss of 0.006; likewise, Squeezenet1.1 loss is 0.046.

From the results shown in Table VI, it is clear that VGG16 has achieved the highest accuracy among models with 99.7% accuracy with a comparable result of 98.8% to Resnet50. While ResNet50 achieved an accuracy of 95.2% during shallow training, Inception-v3 achieved a near accuracy of 92.8%. In terms of training time, shallow training consumes less time as in Squeezenet1.1consumed around 4 minutes on our dataset due to its low number of trainable parameters, than Resnet50 consumes a bit longer of around 6 minutes. Consuming a short amount of time is related to the use of a Gpu. We also observe that loss in deep is lower than shallow leaning as VGG16 has a loss of 0.068; likewise, ResNet50loss is 0.108.

The results now provide evidence that the deep VGG16 model has the best performance to be used to diagnose plant diseases on our cost-effective and autonomous robot. Although other models have high accuracies, VGG16 was chosen because of its highest accuracy of all models. Also, diseases need to be accurately diagnosed. Moreover, there is no need for high response time needed for navigation in greenhouses. Plant diseases are detected using a standard RGB camera on our mobile robot computer. Images are analyzed by Jetson Nano Developer Kit, which is strong enough for these computational tasks. The detection code must be the lightest and most minor complex to occupy the most miniature RAM and be run using the minimum computational power because mobile computing needs to compute more than just the detection task. Tasks like autonomous navigation and driving the motors are among the

Received 4/5/2021; Accepted 28l8l2021

Iraqi Journal of Computers, Communications, Control & Systems Engineering (IJCCCE), Vol. 21, No. 4, December 2021

DOI: https://doi.org/10.33103/uot.ijccce.21.4.2

things needed to be co-computed by the robot. In this work, the trained model for VGG16 is saved and then used for classification. The classification method determines if a plant leaf is contaminated or not, the type of plant disease, and the variety of plants. The detection and recognition of correct plant disease classifications of various images of artificial trees using our robot are shown in *Fig*. 7, online grape images in *Fig*. 8 and online cotton leaf images in *Fig*. 9.

TABLE.5. ACCURACY, TRAINING TIME, AND LOSS OF CNN MODELS USING THE FIRST DATASET.

Model and Training Type	Training Loss	Training Time (in min)	Test Accuracy%
Inception-v3 (Shallow training)	1.251	30.688	96.282
Inception-v3 (Deep training)	0.048	151.215	97.108
ResNet50 (Shallow training)	0.321	21.723	97.888
ResNet50 (Deep training)	0.006	108.752	99.667
Squeezenet1.1 (Shallow training)	0.076	10.150	98.577
Squeezenet1.1 (Deep training)	0.046	29.901	99.678
VGG16 (Shallow training)	0.271	30.409	97.062
VGG16 (Deep training)	0.063	185.900	99.908

TABLE.6. ACCURACY, TRAINING TIME, AND LOSS OF CNN MODELS USING THE SECOND DATASET.

Model and Training Type	Training Loss	Training Time (in min)	Test Accuracy%
Inception-v3 (Shallow training)	0.869	8.510	92.885
Inception-v3 (Deep training)	0.277	15.537	96.44٢
ResNet50 (Shallow training)	0.341	6.571	95.256
ResNet50 (Deep training)	0.108	11.85	98.8١٤
Squeezenet1.1 (Shallow training)	0.169	4.433	90.909
Squeezenet1.1 (Deep training)	0.337	5.343	92.09٤
VGG16 (Shallow training)	0.900	7.760	86.561
VGG16 (Deep training)	0.068	17.527	99.754

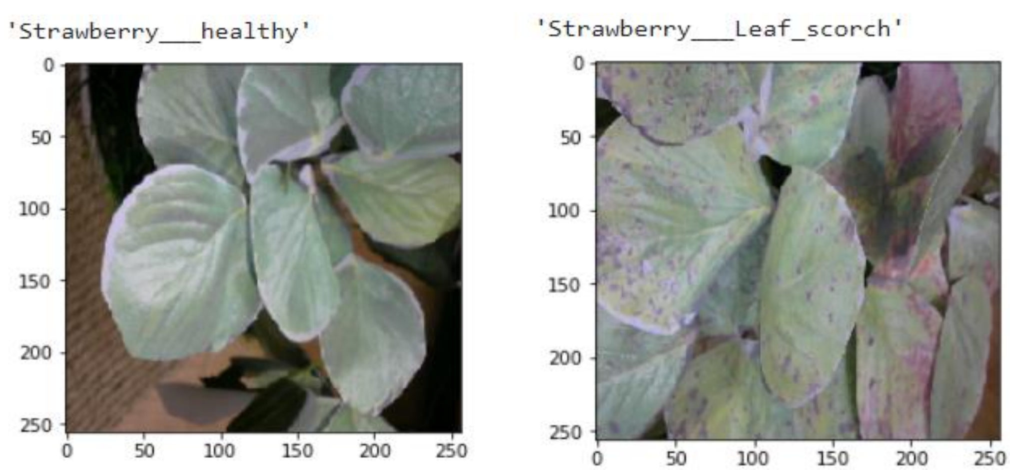

FIG. 7. STRAWBERRY PLANT.

Received 4/5/2021; Accepted 28l8l2021

Iraqi Journal of Computers, Communications, Control & Systems Engineering (IJCCCE), Vol. 21, No. 4, December 2021

DOI: https://doi.org/10.33103/uot.ijccce.21.4.2

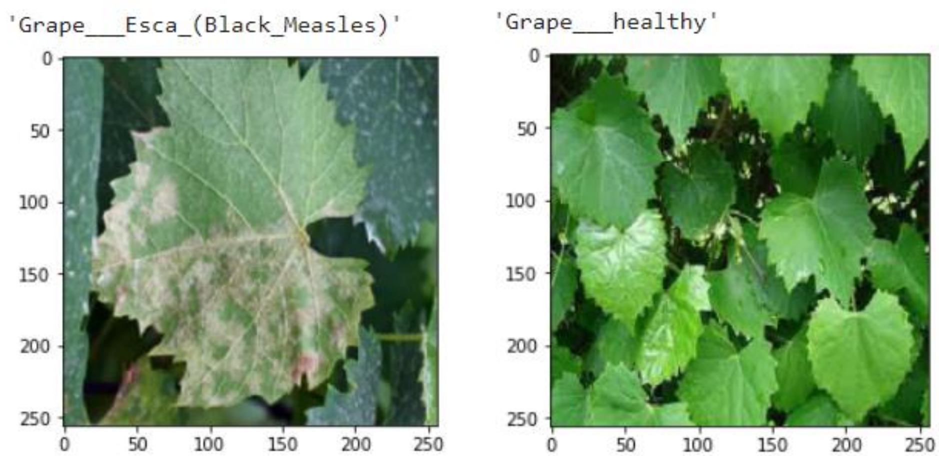

FIG. 8. GRAPE PLANT.

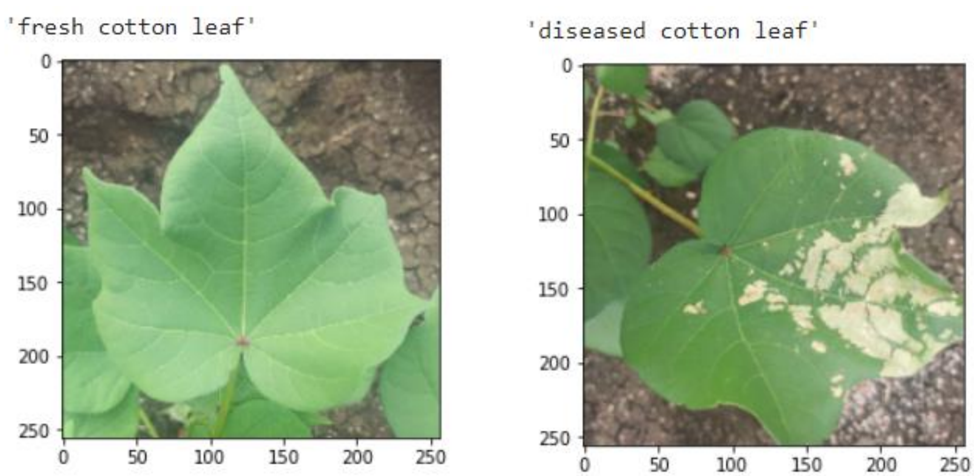

FIG. 9. COTTON PLANT.

V. CONCLUSION

The findings of the current work confirm that using transfer learning on pre-trained CNN architectures achieves high accuracy and consumes a short time. This work used two transfer learning methods: shallow transfer learning and deep transfer learning for retraining Inceptionnet-v3, ResNet50, Squeezenet1-1, and VGG16. Both transfer learning methods achieve considerable accuracies on test images. However, detection of plants diseases needs to be most accurate to stop the spread of disease. Even though in-depth training takes longer than the shallow counterpart, it achieves higher accuracy with less training loss. The highest accuracy achieved from VGG16 using deep transfer learning makes it the best model to be part of our plant disease detection system within the robot. In addition, local datasets might prove an important area for future research.

Received 4/5/2021; Accepted 28l8l2021

DOI: https://doi.org/10.33103/uot.ijccce.21.4.2

Abbreviations:

FAO: Food and Agricultural Organization of the United Nations.

DL: deep learning.

CNN: convolutional neural network.

TL: transfer learning.

REFERENCES

[1] H. T. Jadhav and K. A. Rosentrater, “Economic and environmental analysis of greenhouse crop production with special reference to low cost greenhouses: A review,” *2017 ASABE Annu. Int. Meet.*, 2017, doi: 10.13031/aim.201701178.

[2] D. JH, “An Overview of Plant Immunity,” *J. Plant Pathol. Microbiol.*, vol. 6, no. 11, 2015, doi: 10.4172/2157-7471.1000322.

[3] A. M. Mutka and R. S. Bart, “Image-based phenotyping of plant disease symptoms,” *Front. Plant Sci.*, vol. 5, no. JAN, pp. 1–9, 2015, doi: 10.3389/fpls.2014.00734.

[4] S. S. H. Hajjaj and K. S. M. Sahari, “Review of agriculture robotics: Practicality and feasibility,” *IRIS 2016 - 2016 IEEE 4th Int. Symp. Robot. Intell. Sensors Empower. Robot. with Smart Sensors*, no. December 2016, pp. 194–198, 2017, doi: 10.1109/IRIS.2016.8066090.

[5] F. Qin, D. Liu, B. Sun, L. Ruan, Z. Ma, and H. Wang, “Identification of alfalfa leaf diseases using image recognition technology,” *PLoS One*, vol. 11, no. 12, pp. 1–26, 2016, doi: 10.1371/journal.pone.0168274.

[6] S. D. Khirade and A. B. Patil, “Plant disease detection using image processing,” *Proc. - 1st Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2015*, pp. 768–771, 2015, doi: 10.1109/ICCUBEA.2015.153.

[7] K. Elangovan, “Plant Disease Classification Using Image Segmentation and ppp Techniques,” *Int. J. Comput. Intell. Res.*, vol. 13, no. 7, pp. 1821–1828, 2017, [Online]. Available: http://cs231n.stanford.edu/reports/2017/pdfs/325.pdf.

[8] A. S. Al-araji and N. Q. Yousif, “A Cognitive Hybrid Tuning Control Algorithm Design for Nonlinear Path-Tracking Controller for Wheeled Mobile Robot,” *Al-Khwarizmi Eng. J.*, vol. 13, no. 3, pp. 64–73, 2017.

[9] Y. Lecun, “Scaling Learning Algorithms Towards AI Authors : Yoshua Bengio , Yann LeCun Presenter : Marilyn Vazquez Curse of Dimensionality Shallow Learning,” 2017.

[10] A. K. Taqi and A. E. Korial, “Comparison between Feature Based and Deep Learning Recognition Systems for Handwriting Arabic Numbers,” vol. 9, no. 4, pp. 51–66, 2018.

[11] H. Durmus, E. O. Gunes, and M. Kirci, “Disease detection on the leaves of the tomato plants by using deep learning,” *2017 6th Int. Conf. Agro-Geoinformatics, Agro-Geoinformatics 2017*, 2017, doi: 10.1109/Agro-Geoinformatics.2017.8047016.

[12] S. K. Panigrahi, K. Das, D. Mishra, and B. K. Veedhi, *A survey on hybridized gene selection strategies*, vol. 153. 2021.

[13] A. Lofti, H. Bourchachia, A. Gegov, C. Langensiepen, and M. McGinnity, *Advances in Computational Intelligence Systems: Contributions Presented at the 18th UK Workshop on Computational Intelligence, September 5-7, 2018, Nottingham, UK*, vol. 840. 2019.

[14] D. P. Hughes and M. Salathe, “An open access repository of images on plant health to enable the development of mobile disease diagnostics,” 2015, [Online]. Available: http://arxiv.org/abs/1511.08060.

[15] Janmejaybhoi, “Cotton Dataset,” 2020. https://www.kaggle.com/janmejaybhoi/cotton-disease-dataset (accessed Jul. 22, 2021).

[16] E. Fujita, Y. Kawasaki, H. Uga, S. Kagiwada, and H. Iyatomi, “Basic investigation on a robust and practical plant diagnostic system,” *Proc. - 2016 15th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2016*, pp. 989–992, 2017, doi: 10.1109/ICMLA.2016.56.

[17] A. Dhakal and P. S. Shakya, “Image-Based Plant Disease Detection with Deep Learning,” vol. 61, no. 1, pp. 26–29, 2018.

Received 4/5/2021; Accepted 28l8l2021

DOI: https://doi.org/10.33103/uot.ijccce.21.4.2

[18] S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using deep learning for image-based plant disease detection,” *Front. Plant Sci.*, vol. 7, no. September, pp. 1–10, 2016, doi: 10.3389/fpls.2016.01419.

[19] I. Z. Mukti and D. Biswas, “Transfer Learning Based Plant Diseases Detection Using ResNet50,” *2019 4th Int. Conf. Electr. Inf. Commun. Technol. EICT 2019*, no. December, pp. 1–6, 2019, doi: 10.1109/EICT48899.2019.9068805.

[20] V. Kumar, H. Arora, Harsh, and J. Sisodia, “ResNet-based approach for Detection and Classification of Plant Leaf Diseases,” *Proc. Int. Conf. Electron. Sustain. Commun. Syst. ICESC 2020*, no. Icesc, pp. 495–502, 2020, doi: 10.1109/ICESC48915.2020.9155585.

[21] J. Wäldchen and P. Mäder, *Plant Species Identification Using Computer Vision Techniques: A Systematic Literature Review*, vol. 25, no. 2. Springer Netherlands, 2018.

[22] S. Gould, R. Fulton, and D. Koller, “Decomposing a scene into geometric and semantically consistent regions,” *Proc. IEEE Int. Conf. Comput. Vis.*, no. Iccv, pp. 1–8, 2009, doi: 10.1109/ICCV.2009.5459211.

[23] S. Imambi, K. B. Prakash, and G. R. Kanagachidambaresan, *PyTorch*. 2021.

[24] L. Wang, C.-Y. Lee, Z. Tu, and S. Lazebnik, “Training Deeper Convolutional Networks with Deep Supervision,” 2015, [Online]. Available: http://arxiv.org/abs/1505.02496.

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.

[26] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[27] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size,” pp. 1–13, 2016, [Online]. Available: http://arxiv.org/abs/1602.07360.

[28] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[29] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, “On the importance of initialization and momentum in deep learning,” *30th Int. Conf. Mach. Learn. ICML 2013*, no. PART 3, pp. 2176–2184, 2013.

[30] Q. Yan, B. Yang, W. Wang, B. Wang, P. Chen, and J. Zhang, “Apple leaf diseases recognition based on an improved convolutional neural network,” *Sensors (Switzerland)*, vol. 20, no. 12, pp. 1–14, 2020, doi: 10.3390/s20123535.

[31] T. Sziranyi, I. Kopilovic, and B. P. Toth, “Anisotropic Diffusion as a Preprocessing Step for Efficient Image Compression * Analogical and Neural Computing Laboratory , Comp . & Automation Inst ., Hungarian Academy of,” pp. 11–13.

Received 4/5/2021; Accepted 28l8l2021