MAJRADOC AN IMAGE BASED DISEASE DETECTION APP FOR AGRICULTURAL PLANTS USING DEEP LEARNING TECHNIQUES

Sara Saleh Alfozan and Mohamad Mahdi Hassan

Department of Computer Science, College of Computer, Qassim University, Buraydah, Saudi Arabia

ABSTRACT

Infection of agricultural plants is a serious threat to food safety. It can severely damage plants' yielding capacity. Farmers are the primary victims of this threat. Due to the advancement of AI, image-based intelligent apps can play a vital role in mitigating this threat by quick and early detection of plants infections. In this paper, we present a mobile app in this regard. We have developed MajraDoc to detect some common diseases in local agricultural plants. We have created a dataset of 10886 images for ten classes of plants diseases to train the deep neural network. The VGG-19 network model was modified and trained using transfer learning techniques. The model achieved high accuracy, and the application performed well in predicting all ten classes of infections.

KEYWORDS

Plant diseases, plant diseases diagnosis, deep learning, VGG19 CNN, mobile application.

1. INTRODUCTION

Agriculture is one of the vital sources of economic development [1]. Healthy plants are essential for human survival. It ensures higher productivity and quality of crops, fruits, and vegetables. However, infections due to various diseases negatively impact both productivity and quality of yields that may cause food insecurity [2].

Disease identification in a plant is very important in a successful farming system. In general, a farmer discovers disease symptoms in plants through naked-eye observations, which demands constant monitoring [3]. Moreover, with low education levels of farmers coupled with limited awareness and lack of access to plant pathologists, human-assisted disease diagnosis is not effective and cannot keep up with the exorbitant requirements [4]. It is not an easy task to diagnose plant disease through optical observation of the symptoms on plant leaves. This method incorporates a significantly high degree of complexity. Even skilled agronomists and plant pathologists frequently fail to detect specific illnesses due to this intricacy and the enormous number of grown plants and their existing phytopathological problems, leading to incorrect conclusions and treatments [5][6].

A smart mobile application for the detection and diagnosis of plant diseases is proposed in this work to assist farmers in detecting plant diseases. This work would be of great use to agronomists who are asked to do such diagnostics by optical observation of infected plant leaves. Instead of optical observations, the farmer can diagnose plant disease by feeding the application an image of leaves of a specific infected plant. The application will give the correct disease diagnosis with

David C. Wyld et al. (Eds): CSE, AI & FL, NLPTT - 2021 pp. 19-31, 2021. CS & IT - CSCP 2021

20 Computer Science & Information Technology (CS & IT)

high accuracy. The mobile application was developed using an image based convolutional neural network (CNN) which can easily detect and give a faster diagnosis of plant diseases which may help develop an early treatment technique. A pre-trained deep learning model (VGG19) was trained to recognize 6 different diseases of 4 crop species, namely pepper powdery mildew, cucumber downy mildew, zucchini powdery mildew, tomato mosaic virus, tomato bacterial spot, and tomato spotted Spider mite. The model was trained and tested using a data set collected by a professional person. The VGG19 model architecture and the dataset are discussed in section 3. Section 2 presents related work. Section 4 comprehensively discusses the simulation analysis and results to ensure good performance. Finally, in section 5, the conclusions are presented.

2. Related Work

Researchers who are located at varied places on the earth, alongside experts in artificial intelligence and botanists, have explored variant techniques in order to classify plant ailments. A study by Rumpf et al. has provided early findings regarding the diagnosis and the identification of sugar beet diseases using Support Vector Machine (SVM), based on the spectrogram of plant indexes [7]. Using the K-Means clustering method on color and texture extracted features, with Artificial Neural Network (ANN) Al-Hiary et al. has conducted segmentation of diseased areas for a set of five plant diseases [8]. Ravathi was able to identify 6 observed diseases from cotton leaves by Cross Information Gain Deep Forward Neural Network. The inputs of the NN were a set of vectorized information obtained from the images, such as texture, color, edge-based features, in addition to Particle Swarm Optimization for feature selection [9]. A further related study, by Mokhtar et al., has been implemented using SVM, on Tomato leaf ailments for the purpose of recognition of two viruses; the widespread tomato yellow leaf curl virus (TYLCV), and tomato spotted wilt virus (TSWV) [10]. And many other researches were conducted on different types of plants leaves and agronomic diseases such as vine, and wheat [11][12].

It is a fact that all the above-mentioned studies have successfully achieved good classification accuracies for well-known agronomic defects, within a range between 75% - 92%, by the use of regular machine learning algorithms. Nonetheless, they require manual pre-processing and multi-feature extraction techniques in a way that is expensive computationally and has excessive processing time [13].

Most of the recent researches show competitive exertions and achievements regarding plant diseases recognition with well-developed automated systems. Hitherto, the continuous underpinning of artificial intelligence technologies provided researchers community significant results with low-time consumption. Moreover, the main highlight of recently utilized deep neural network models is the exclusion of features extraction manually [14].

Thaiyalnayaki and Joseph have conducted classification of soybean diseases (absent and colored brown) using Probabilistic Neural Network (PNN), a custom net multilayer perceptron, with a large database, consisting of 683 instances (36 attributes). The PNN recruited structure has achieved 94.1435% classification accuracy with 19 output neurons and 84 input neurons. SoftMax and ReLu activation functions were chosen for the layers aside, with negative loglikelihood loss optimization function and SGD. However, the classic SVM has just achieved 88.7262% accuracy [15]. Moreover, another considerable performance for the recognition of plant diseases by image-based naive networks (Relu rectifier linear unit), and transfer learning (VGGNet, ResNet50, and Inception-v3). VGG16 had the most relevant result with 93.5 classification accuracy [16]

Rangarajan et al. (2018) have examined the separability of pre-trained AlexNet and VGG16 net among six health classes of tomato crops (13,262 segmented images from PlantVillage dataset).

When the models were at their best state by tuning the hyperparameters, changing minibatch size, and the number of images used with the models, 97.29% and 97.49% classification accuracy was achievable for AlexNet and VGG16 respectively [17]. A further automated diseases identification AI model was implemented to detect 5 different diseased mediums of leaves, by the use of CNN (convolutional neural network); 1) Early Blight, 2) Late Blight, mostly found in Potatoes, and 3) Esca, 4) Isariopsis, 5) Black Rot, in Grapes. The recruited model of CNN concluded that 87.47% and 91.96% of classification accuracies were possibly obtained for Potatoes and Grapes respectively [18].

Hallau et al. (2018) built a system to identify sugar beet leaf diseases based on captured images using smartphone cameras. The system can identify five categories of leaf diseases of sugar beet-Cercospora or leaf spot, beet rust, bacterial blight, ramularia leaf spot, andphoma leaf spot. The system goes through the following steps: 1) Infected region detection, 2) Feature extraction, and 3) Class prediction by using Support Vector Machine (SVM). They find out multiple diseases can occur simultaneously on the same plant which complicates the identification process [19].

Mohanty et. al (2016) used the PlantVillage data set which contains 38 class labels (i.e., infections) of 14 crops. They built a model using a deep convolutional neural network (CNN) to diagnose 26 diseases of those 14 crops or their absence. According to them, CNN is applicable to image classification problems even without any feature engineering. Their trained model correctly classifies crop disease with an accuracy of 99.35%. Training the model took a significant amount of time but during testing, it was very fast. Based on their tested model they developed an app for smartphones. They found some limitations though, like when tested on a set of images taken under conditions different from the images used for training, the model's accuracy reduced substantially[20].

Alvaro et al. (2017) presented a deep learning approach to detect diseases and pests in tomato plants using images captured by portable camera devices with various resolutions. Their dataset was collected from different farms on Korean Peninsula. It consists of about 5000 images, categorized and annotated into nine tomato diseases as Gray mold, Leaf mold, Low temperature, Plague, Leaf miner, Whitefly, Canker, Nutritional excess or deficiency, and Powdery mildew. Since they had a relatively smaller dataset, they applied extensive data augmentation to avoid overfitting. They considered three main detectors: Faster Region-based Convolutional Neural Network, Single Shot Multibox Detector, and Region-based Fully Convolutional Network. Finally, combined each of these meta-architectures with deep feature extractors. Their experimental results suggest various deep-meta-architectures with feature extractors can successfully and accurately recognize nine different categories of tomato diseases [21].

3. MATERIAL AND METHODS

3.1. VGG19 Network

VGG-19 [22] is a convolutional neural network that is 19 layers deep based on the stacked architecture of AlexNet with more numbers of convolution layers added to the model. More than a million images from the ImageNet database were used for training VGG-19 CNN. This network can classify photos into 1000 different object categories. It has 13 convolution layers, each of which is followed by a ReLU layer. Similar to AlexNet, some of the convolution layers are followed by max-pooling to minimize the dimension. The convolutional kernel is 3*3 in size, while the input is 224*224*3.

Computer Science & Information Technology (CS & IT)



Figure 1. VGG19 network architecture 19 [22]

Figure 1 illustrates VGG-19 network model structure with the size of each layer .VGG-19 CNN is used as a pre-processing model. The network depth has been improved in comparison to traditional convolutional neural networks. It is better than a single convolution because it employs an alternating structure of several convolutional layers and non-linear activation layers. The multiple layer structure can better extract image features, use Maxpooling for downsampling, and modify the linear unit (ReLU) as the activation function, that is, choose the largest value in the image area as the area's pooled value. The downsampling layer is primarily used to increase the network's anti-distortion capabilities to the image while preserving the sample's key features and minimizing the number of parameters[23].

3.2. VGG19 CNN Implementation

In this paper, VGG19 CNN was implemented using Keras library and python language. The application of pre-trained deep learning models for classifying new classes of objects is employed in this work, which is referred to as transfer learning. In VGG-19, the parameters are concentrated in three FC layers. The parameters of the network were originally designed for 1000 classification, but this article only focuses on the classification of 10 categories (Healthy Pepper, Pepper powdery mildew disease, Healthy Cucumber, Cucumber downy mildew disease, Zucchini powdery mildew disease, Healthy Zucchini, Healthy Tomato, Tomato mosaic virus, Tomato bacterial spot, Tomato spotted Spider mite). Therefore, the last layer has been replaced with the output layer, which is equal to the number of classes. In addition, VGG-19's three fully connected layers were replaced with a single Flatten layer and three fully connected layers. Because the convolution layer and the Dense fully connected layer cannot be connected directly, a Flatten layer is added. The modified architecture of VGG19 is shown in figure 2.

Computer Science & Information Technology (CS & IT)

Layer (type)	Output	Shape	Param #
input_1 (InputLayer)	(None,	224, 224, 3)	0
block1_conv1 (Conv2D)	(None,	224, 224, 64)	1792
block1_conv2 (Conv2D)	(None,	224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None,	112, 112, 64)	0
block2_conv1 (Conv2D)	(None,	112, 112, 128)	73856
block2_conv2 (Conv2D)	(None,	112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None,	56, 56, 128)	0
block3_conv1 (Conv2D)	(None,	56, 56, 256)	295168
block3_conv2 (Conv2D)	(None,	56, 56, 256)	590080
block3_conv3 (Conv2D)	(None,	56, 56, 256)	590080
block3_conv4 (Conv2D)	(None,	56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None,	28, 28, 256)	0
block4_conv1 (Conv2D)	(None,	28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None,	28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None,	28, 28, 512)	2359808
block4_conv4 (Conv2D)	(None,	28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None,	14, 14, 512)	0
block5_conv1 (Conv2D)	(None,	14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None,	14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None,	14, 14, 512)	2359808
block5_conv4 (Conv2D)	(None,	14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None,	7, 7, 512)	0
spec (Flatten)	(None,	25088)	0
dense_1 (Dense)	(None,	100)	2508900
dense_2 (Dense)	(None,	100)	10100
dense_3 (Dense)	(None,	100)	10100
dense_4 (Dense)	(None,	12)	1212
Total params: 22,554,696 Trainable params: 2,530,312 Non-trainable params: 20,024,	384		

Figure 2. The modified VGG19 architecture

3.3. Dataset

Training and validation datasets were collected with the help of a specialist inplant diseases at Qassim University. We were able to obtain 10886 images for 10 classes, namely Healthy Pepper, Pepper powdery mildew disease, Healthy Cucumber, Cucumber downy mildew disease, Zucchini powdery mildew disease, Healthy Zucchini, Healthy Tomato, Tomato mosaic virus disease, Tomato bacterial spot disease, Tomato spotted wilt disease. The images were divided into two datasets, the training dataset, and the validation dataset, by randomly splitting the 10886 images, so the data are allocated into training and testing set in the ratio of 80:20. In other words, 80% of the data are selected to conduct the training process while 20% are for testing purposes. The Pareto principle is a common rule of thumb to divide the dataset into two sub-sets; training and testing data. This is also called the 80/20 rule[24]. 80% of the images were used for training, and 20% of the images were used for validation. The number of images in each category are shown in figure 3.





Figure 3. Number of images in each class in the dataset



Figure 4. Samples of the dataset (a) Health Pepper(b) Healthy Zucchini (c) Healthy Tomato (d) Zucchini powdery mildew disease

3.4. Mobile Application

For building the iOS mobile application, we had to convert the trained model from Keras format into a suitable format that can be used in Xcode for developing the plant disease detection and diagnosis mobile app. The tool that was used for converting the model is coremitools. Xcode was used for developing the mobile application using the converted model.

3.4.1. Coremitools

Coremltools is a Python package that allows you to: (i) convert trained models from common machine learning tools to Core ML format (. mlmodel), (ii) write models to Core ML format

using a simple API, and (iii) make predictions using the Core ML framework (on certain platforms) to check conversion. Apple's Core ML framework makes it simple for developers to integrate machine learning (ML) models into their apps. iOS, iPadOS, watchOS, macOS, and tvOS all support Core ML. Deep neural networks (convolutional and recurrent), tree ensembles (boosted trees, random forest, decision trees), and generalized linear models are among the ML



Figure 5. Keras model to Core ML model with coremltools

approaches that Core ML presents as a public file format (.mlmodel). Within XCode, you can embed core ML models straight into apps. Since our trained model was in Keras format, it was necessary to convert it into Core ML in order to be used for building an iOS mobile app using XCode. In Figure 5 we show the overall conversion process.

3.4.2. XCode

Xcode is Apple's integrated development environment for macOS, used to develop software for macOS, iOS, iPadOS, watchOS, and tvOS. It is the only officially supported way to develop iOS and other Apple OS apps. We used Xcode for developing the iOS mobile app using the model that was converted from Keras format into Core ML by coremitools Python package. The developed mobile application is easy to use and can be used by farmers for detecting the plant diseases that were mentioned above.

4. RESULTS AND DISCUSSION

The modified VGG19 model was trained using the training dataset which is 80% of the whole dataset. The learning rate was set to 0.001 and the number of epochs was set to 150. During the training process, the training accuracy, validation accuracy, training loss, and validation loss were plotted for the whole 150 epochs to show the performance of the model. Figure 6 shows that the training loss and validation loss decreases over the 150 training epochs, while the training accuracy and the validation accuracy increase consistently. The test dataset was used to find the test accuracy of the model. It was found to be 99.29% at epoch 150. The trained model was saved at this point. This high accuracy indicates that our trained model will perform well on classifying the 10 classes that it was trained to recognize, namely Healthy Pepper, Pepper powdery mildew disease, Healthy Cucumber, Cucumber downy mildew disease, Zucchini powdery mildew disease, and Tomato spotted wilt disease.



Figure 6. Accuracy and loss for training and validation of the modified VGG19 CNN model

4.1. Testing using Majradoc Mobile App

To evaluate the final performance of our proposed system for recognizing plant diseases, the mobile app that was built using the trained model was installed on iOS mobile. Random images were chosen for which the trained model has never seen. These images were captured with different orientations and illuminations and at different distances from the camera to check the robustness of the trained model. The images were fed to the mobile application one after the other. The application was able to predict the classes of the images correctly. All the following images are screenshotting for MajraDoc app using iPhone 11.

i. Healthy-Plants Images

Figure 7 illustrates the performance of the trained model for three healthy plants images were chosen randomly. It shows some results of diagnosis using MajraDoc application. Where (a) Healthy Tomato leaf (b) Healthy Cucumber leaf (c) Healthy Zucchini. The three images were fed to the mobile application. It was able to predict the classes of the images correctly.



Figure 7. Testing the performance of the model using MajraDoc application for healthy plants

ii. Sick-Plants Images

Figure 8 shows prediction to three trained plants diseases, in each prediction the model was able to give the correct diagnosis with all of these sick plants images. It shows some results of diagnosis using MajraDoc application Where (a) Zucchini powdery mildew disease (b) Pepper powdery mildew disease (c) Cucumber downy mildew disease. It was able to predict the diagnoses of the images correctly.

Computer Science & Information Technology (CS & IT)





iii. Untrained-Plants Images

Figure 9 illustrates the performance of the trained model where not trained plants were captured. It shows results of diagnosis using iPhone 11 in MajraDoc application. It was able to know the image is not from the trained plants' diseases classes.



Figure 9. Testing the performance of the model using MajraDoc application for random plant

iv. Not-Plants Images

Figure 10 illustrates the performance of the trained model where no plants were captured. It shows results using iPhone 11 in MajraDoc application. It showed the MajraDoc app able to handle images, not for plants or a wrong image.



Figure 10. Testing the performance of the model using MajraDoc application for not plant

The results and discussion in this section demonstrate that our proposed system that was constructed using our modified VGG19 CCN illustrated in a previous section and the MajraDoc iOS mobile application is performing well in recognizing plant diseases for which the model was trained to recognize. In addition, performed well in distinguishing between trained and untrained plants or a random image not containing plant. This application will be beneficial for farmers in diagnosing crop disease easily.

Compared with the previous researches, our paper achieved high accuracy, as well as works on a real mobile environment .the reasons, are due to the trained dataset that was used, in addition to using a powerful and suitable Al model for the plant diseases detection. The dataset had images were captured in several directions and modify them under different lights, which gave better results.

5. CONCLUSIONS

In this work, the performance of the model was evaluated by using the test dataset that we collected. The obtained test accuracy for the model was 99.29%. An iOS mobile application was developed for the classification of plant diseases using the trained model that was converted to Core ML format (. mlmodel). The final mobile application was evaluated by feeding new images to the application. All the images were classified correctly. This study proves that deep learning architectures such as VGG-19 are suitable for the detection of plant diseases. As well as, the quality of the dataset used to train the model is very important to give correct results and achieve high accuracy. However, in this study, the model was trained to recognize only 6 plant diseases and 4 healthy plants, it can be improved to recognize more diseases by training the model on a large dataset that contains images of more plant diseases. The developed application will help farmers in detecting plant disease easily without the need of consulting plant pathologists.

REFERENCES

- U. Shruthi, V. Nagaveni, and B. K. Raghavendra, "A Review on Machine Learning Classification Techniques for Plant Disease Detection," in 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Mar. 2019, pp. 281–284, doi: 10.1109/ICACCS.2019.8728415.
- [2] K. K. Singh, "An Artificial Intelligence and Cloud Based Collaborative Platform for Plant Disease Identification, Tracking and Forecasting for Farmers," in *Proceedings - 7th IEEE International Conference on Cloud Computing in Emerging Markets, CCEM 2018*, 2019, pp. 49– 56, doi: 10.1109/CCEM.2018.00016.
- [3] V. Singh and A. K. Misra, "Detection of plant leaf diseases using image segmentation and soft computing techniques," *Inf. Process. Agric.*, vol. 4, no. 1, pp. 41–49, 2017, doi: 10.1016/j.inpa.2016.10.005.
- [4] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agric.*, vol. 145, pp. 311–318, 2018, doi: 10.1016/j.compag.2018.01.009.
- [5] M. Jeger *et al.*, "Global challenges facing plant pathology: multidisciplinary approaches to meet the food security and environmental challenges in the mid-twenty-first century," *CABI Agric. Biosci.*, vol. 2, no. 1, 2021, doi: 10.1186/s43170-021-00042-x.
- [6] S. R. Maniyath et al., "Plant disease detection using machine learning," in Proceedings 2018 International Conference on Design Innovations for 3Cs Compute Communicate Control, ICDI3C 2018, 2018, pp. 41–45, doi: 10.1109/ICDI3C.2018.00017.
- [7] T. Rumpf, A. K. Mahlein, U. Steiner, E. C. Oerke, H. W. Dehne, and L. Plümer, "Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance," *Comput. Electron. Agric.*, vol. 74, no. 1, pp. 91–99, 2010, doi: 10.1016/j.compag.2010.06.009.
- [8] H. Al Hiary, S. Bani Ahmad, M. Reyalat, M. Braik, and Z. ALRahamneh, "Fast and Accurate Detection and Classification of Plant Diseases," *Int. J. Comput. Appl.*, vol. 17, no. 1, pp. 31–38, Mar. 2011, doi: 10.5120/2183-2754.
- [9] P. Revathi and M. Hemalatha, "Cotton disease identification using proposed CIG-DFNN classifier," *Asian J. Sci. Res.*, vol. 7, no. 2, pp. 225–231, Mar. 2014, doi: 10.3923/ajsr.2014.225.231.
- [10] U. Mokhtar, M. A. S. Ali, A. E. Hassanien, and H. Hefny, "Identifying two of tomatoes leaf viruses using support vector machine," in *Advances in Intelligent Systems and Computing*, 2015, vol. 339, pp. 771–782, doi: 10.1007/978-81-322-2250-7_77.
- [11] X. E. Pantazi, D. Moshou, A. A. Tamouridou, and S. Kasderidis, "Leaf disease recognition in vine plants based on local binary patterns and one class support vector machines," in *IFIP Advances in Information and Communication Technology*, 2016, vol. 475, pp. 319–327, doi: 10.1007/978-3-319-44944-9_27.
- [12] A. Johannes *et al.*, "Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case," *Comput. Electron. Agric.*, vol. 138, pp. 200–209, Jun. 2017, doi: 10.1016/j.compag.2017.04.013.
- [13] T. M. M. S. Mohamed *et al.*, "The Identification of Significant Mechanomyography TimeDomain Features for the Classification of Knee Motion," in *Lecture Notes in Electrical Engineering*, 2022, vol. 730, pp. 313–319, doi: 10.1007/978-981-33-4597-3_29.
- [14] F. Jiang *et al.*, "Artificial intelligence in healthcare: Past, present and future," *Stroke and Vascular Neurology*, vol. 2, no. 4. pp. 230–243, Dec. 2017, doi: 10.1136/svn-2017-000101.
- [15] K. Thaiyalnayaki and C. Joseph, "Classification of plant disease using SVM and deep learning," in *Materials Today: Proceedings*, May 2021, vol. 47, pp. 468–470, doi: 10.1016/j.matpr.2021.05.029.
- [16] A. V Panchal, S. C. Patel, K. Bagyalakshmi, P. Kumar, I. R. Khan, and M. Soni, "Image-based Plant Diseases Detection using Deep Learning," *Mater. Today Proc.*, Aug. 2021, doi: 10.1016/j.matpr.2021.07.281.
- [17] A. K. Rangarajan, R. Purushothaman, and A. Ramesh, "Tomato crop disease classification using pre-trained deep learning algorithm," in *Proceedia Computer Science*, 2018, vol. 133, pp. 1040– 1047, doi: 10.1016/j.procs.2018.07.070.
- [18] A. Ghosh and P. Roy, "AI Based Automated Model for Plant Disease Detection, a Deep Learning Approach," in *Communications in Computer and Information Science*, 2021, vol. 1406 CCIS, pp. 199–213, doi: 10.1007/978-3-030-75529-4_16.

- [19] L. Hallau *et al.*, "Automated identification of sugar beet diseases using smartphones," *Plant Pathol.*, vol. 67, no. 2, pp. 399–410, Feb. 2018, doi: 10.1111/ppa.12741.
- [20] 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, Sep. 2016, doi: 10.3389/fpls.2016.01419.
- [21] A. Fuentes, S. Yoon, S. Kim, and D. Park, "A Robust Deep-Learning-Based Detector for RealTime Tomato Plant Diseases and Pests Recognition," *Sensors*, vol. 17, no. 9, p. 2022, Sep. 2017, doi: 10.3390/s17092022.
- [22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.
- [23] A. Ajit, K. Acharya, and A. Samanta, "A Review of Convolutional Neural Networks," 2020, doi: 10.1109/ic-ETITE47903.2020.049.
- [24] H. B. Harvey and S. T. Sotardi, "The Pareto Principle," J. Am. Coll. Radiol., vol. 15, no. 6, p. 931, 2018, doi: 10.1016/j.jacr.2018.02.026.

Authors

Sara Saleh Alfaozan: Master Student in Qassim Unvirsity, Computer Science Department, Qassim University, Saudi Arabia.



Mohammad Mahdi Hassan: an Assistant Professor in the Computer Science, Department at Qassim University, Saudi Arabia. He received his PhD in Computer Science (Specialize in Software Engineering) from University of Western Ontario, Canada. His research interests include Software Engineering, Testing, Mining Software Data, Intelligent System Development and Application of Blockchains.

The accuracy result of the trained Model

\varTheta 😐 💼 sa	rasaleh — -bash — 98×49			
372/372 [====================================] - 3058s 8s/step - loss: 0.0177 - acc: 0.9923 - val_loss:			
372/372 [====================================] - 3102s 8s/step - loss: 0.0273 - acc: 0.9904 - val_loss:			
372/372 [====================================] - 3130s 8s/step - loss: 0.0221 - acc: 0.9916 - val_loss:			
372/372 [====================================] - 3155s 8s/step - loss: 0.0253 - acc: 0.9904 - val_loss:			
372/372 [====================================] - 3176s 9s/step - loss: 0.0213 - acc: 0.9928 - val_loss:			
372/372 [====================================] - 3197s 9s/step - loss: 0.0308 - acc: 0.9887 - val_loss:			
372/372 [====================================] - 3226s 9s/step - loss: 0.0201 - acc: 0.9929 - val_loss:			
372/372 [] - 3262s 9s/step - loss: 0.0248 - acc: 0.9913 - val_loss:			
372/372 [] - 3289s 9s/step - loss: 0.0229 - acc: 0.9916 - val_loss:			
372/372 [====================================] - 4503s 12s/step - loss: 0.0200 - acc: 0.9938 - val_loss:			
372/372 [====================================] - 4008s 11s/step - loss: 0.0249 - acc: 0.9914 - val_loss:			
372/372 [====================================] - 3407s 9s/step - loss: 0.0208 - acc: 0.9916 - val_loss:			
372/372 [====================================] - 3671s 10s/step - loss: 0.0246 - acc: 0.9902 - val_loss:			
372/372 [====================================] - 3752s 10s/step - loss: 0.0223 - acc: 0.9928 - val_loss:			
372/372 [====================================] - 16321s 44s/step - loss: 0.0256 - acc: 0.9906 - val_loss			
71/71 [====================================				
(sa) MacBook-alkhas-b-Emtenan:training emtenansaleh\$ cd Desktop				

© 2021 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.