

Citrus Diseases Recognition by Using CNN Model

Walaa N. Jasim^{*1}, Sahera Abued Sead Almola², Mohammed H. Alabiech³, Esra'a J. Harfash⁴
 E-mail: Walaa.jasim@uobasrah.edu.iq, sahera.sead@uobasrah.edu.iq, mohplan@yahoo.com, esra.harfash@uobasrah.edu.iq

¹ Department of Pharmacognosy, College of Pharmacy, University of Basrah, Iraq.

² College of Computer Science and Information Technology, University of Basrah, Iraq.

³ General Company of Electricity Energy Production –Southern Region, Ministry of Electricity, Basrah, Iraq.

⁴ College of Computer Science and Information Technology, University of Basrah, Iraq.

Keywords: Citrus Diseases, Recognition, Detection, Classification, machine learning, Pattern Recognition.

Received: July 12, 2022

Pattern recognition is attracting the interest of researchers in the recently few years as a machine learning approaches due to its vast extending application areas. The application area includes communications, medicine, automations, data mining, military intelligence, document classification, bioinformatics, speech recognition and business. In this research convolutional neural networks (CNN) using for building system to recognize diseases that are happened in citrus. In this study presented dataset for seven classes of citrus diseases which contains 2450 images such as anthracnose, brown rot, citrus black spot, citrus canker, citrus scob, melanose and sooty mold citrus. The proposed system recognizes learned via CNN. The experimental result shows our model has ability to recognize citrus diseases with high and robustness accuracy. The study presented here gives 88% recognition of citrus diseases for the entire database.

Povzetek: Konvolucijske nevronske mreže so uporabljene za detekcijo oz. klasifikacijo boleznih citrusov.

1 Introduction

The citrus is the widest production of fruit in all the world. Because of the harm by different of diseases, so the citrus production is retrenched and the quality is becoming worst. The detection and recognition diseases of the citrus are very significant. In the present the major measures in order to control and citrus disease is taken by strewing pesticides, that is doing spoil for the soil highly therefore, that is not perfect for the environment [1].

The citrus diseases, out of optical observation regarding signs on the plant's leaves, have a great degree of complexity. Due to such complexities, as well as the large numbers of the plowed plants and their present disease issues, even the most experienced and well-trained plant agronomists and pathologists frequently fail to succeed in curing certain diseases, leading to flawed treatments and inferences [2]. In addition, the technologies of image detection and identification can detect what type of citrus diseases they have with minimum cost and maximum efficiency.

The presence of an automated intelligent system for recognition and diagnose citrus diseases, would presentation a beneficial relief to the "agronomist" who is required from him/her implement such diagnoses out of optical monitoring of leaves of infected plant and citrus fruit. Plant diseases, like citrus diseases, may be identified in various approaches. A few diseases don't have obvious signs, or they only manifest after it's too late for any interference [3].

Currently, disease image recognition research is centered on deep learning (DL), particularly the CNN model. DL

and Convolutional Neural Networks (CNN) are two disciplines of NN computing that have recently gained popularity. CNN is a type of NN with a unique structure that was created for modelling the human vision system (HVS). As a result, CNNs has become one of the most demanded areas of Information technology and it has been successfully applied to solving many issues of, computer vision and data visualization. [4].

The goal of this work is to classification citrus diseases that is help agronomists to detect seven categories for citrus diseases these are Anthracnose, brown rot, Citrus black spot, Citrus Canker, Citrus scob, melanose, Sooty Mold citrus. with a view to detect citrus disease automatically, we researched the traits and pests of citrus fruit and citrus leaves to improve accuracy and reduce recognition loss. In this work introduce approaches to complete the recognition of citrus diseases, CNN models is used here to perform the classification of these seven classes of citrus aforementioned. The accuracy of recognition for citrus can reach 88%.

The remainder of the research is organized as follows: after presenting the introduction in section 1, related work is presented in section 2. The CNN is explained in section 3, and the database of the present study is explained in section 4. Section 5 depicts the building of a CNN model for such citrus datasets, while Section 6 details the major stages and approaches utilized to finish the system's work, as well as the suggested system's outcomes and performance evolution. Lastly, the last section contains the conclusion.

2 Literature review

Intelligent recognition systems deliver considerable convenience to all aspects of life, including industrial production, healthcare, transportation, agriculture, and other fields [5]. Manavalan Radhakrishnan [6] published a survey in 2020 on several image processing approaches and ML methods used for extracting and rapid testing of different citrus fruits like grape leaves, oranges, and lemons. The difficulties encountered while using arithmetical methodologies to evaluate citrus leaves are also discussed, as well as future directions. Zongshuai Liu *et al.* [7] presented a DL-based method for recognizing an image of citrus disease in 2020. They created a database for citrus images that contained six of the most common citrus diseases. In the experimentation, they have employed the MobileNetV2 model as essential network and compared it to other models of networks in terms of accuracy, speed, and size of the model. Their findings reveal that their approach has resulted in the reduction of model size and prediction time at the same time as maintaining flawless accuracy of classification. In 2019, Hafiz Tayyab Rauf *et al.* [8] introduced a method that uses ML for the detection of the diseases in citrus early. They were obtained in Dec. from orchards in Pakistan's Sargodha zone, when the fruit has been ready to become round and the most disease have been identified on the citrus plants. A smart mobile detection method for citrus diseases depending on densely connected convolutional networks was presented by WENYAN PAN *et al.* [5] in 2019. With the help of specialists, they were able to create an image database of 6 different forms of citrus diseases and construct an intelligent system for diagnosing citrus diseases through using simplified highly connected convolutional networks (DenseNet). The We-chat applet on mobile devices was used to implement their system. The findings of the experiment show that by simplifying the DenseNet structure, the detection of citrus diseases can be done with an accuracy of roughly 88%, and the time it takes for predicting exhaustion may be reduced. Through DL approaches, Konstantinos P. Ferentinos [2] constructed models of CNNs for implementing diagnosis and detection of plant disease using straightforward images of diseased leaves and healthy plants in 2018. The models were trained through using an open dataset of 87848 images, which included 25 different plants in a group of 58 distinct [(plant, disease)] categories, which include healthy plants. She/He trained various architectures, with the best success rate in recognizing the corresponding [(plant, disease)] (or healthy plant) being around 99.53%. A hybrid technique for classifying and identifying diseases in citrus plants was suggested by Muhammad Sharif *et al.* [9] in 2018. Their suggested method consisted of two stages: first, recognizing lesion areas on citrus leaves and fruits, and second, classifying citrus diseases. A segmentation of optimum weighted technique, which is implemented on an upgraded inputted image, removes the citrus lesion areas. After that, in codebook, texture, geometric features, and color are combined. Furthermore, good

features are selected using a hybrid feature selection approach that incorporates Principal Component Analysis (PCA) entropy, score, and skewness-based covariance vectors. For the ultimate classification of citrus disease, the chosen features are given to the Multiclass Support Vector Machine (MSVM). H.Ali *et al.* [10] reported a method for the detection and classification of major citrus diseases with economic significance in 2017. Their method used the ΔE color variation algorithm for insulating the disease-affected area, as well as color histogram and textural features for disease classification. Our method was successfully executed, with a total accuracy of 99.90% and a similar sensibility of 0.99 plot beneath the curve. In comparison to individual channels, a combination of texture features and color was used for testing, and the results were the same. PCA was used to test the Dimension Reduction(DR) feature sets, and such miniature features have been tested as well with the use of state-of-the-art classifiers.

3 Convolutional neural network (CNN)

The CNN is the most common and widely utilized algorithm in the area of DL [11] [12]. CNN have recently become one of the most appealing approaches, and have been a final factor in a variety of new successes and challenging applications associated with ML applications such as natural object classification and segmentation, handwriting recognition, object detection, face recognition, image classification, and many other fields involving pattern recognition [13]. The CNN [14][15] is a well-known image processing applications. Generally, CNN was advancing scales to larger classification tasks. CNN has three-dimensional neurons, similar to a three-channel RGB image. A CNN typically uses an order 3 tensor as input, like an image with W columns, H rows, and 3 channels, for instance (R, G, B color channels). Higher order tensor inputs, however, can be handled in a comparable way by CNN. The data is subsequently passed through a set of the processing steps in order. A layer is a processing phase which includes a pooling layer, a convolution layer, a normalizing layer, a loss layer, and a fully connected (FC) layer [16]. The CNN then consists of one or many blocks of convolution and subsampling layers, followed by at least 1 FC layer and 1 output layer. Figure ()1 depicts a typical CNN building block [17].

We will present CNN's work in general for classification images and then the steps how used CNN for classifying citrus diseases in our proposed system the CNN is consisted of multi-layers of neurons, each of that is a non-linear operation on a linear transformation regarding the previous layer's outputs. The layers fundamentally contain pooling layers, convolutional layers, and FC layers. The convolutional layers include weights which necessity to be trained, whereas the pooling layers transform the activation by using a specified function [17] [19].

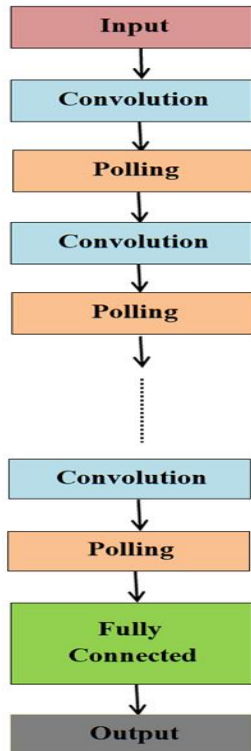


Figure 1: A typically building block of CNN

3.1 The structure of CNN

A CNN contains input, output, and hidden layers. Neural network in any feed-forward, the middle layers are called hidden because their inputs and outputs are convincing via the final convolution and the activation function. The hidden layers in CNN contain layers that perform convolutions [19]. These three essential sub-structures layer of the CNN, are called: pooling layers, convolutional layers, and FC layers. The collecting of these three layers makes up the Convolutional Neural Network [18], we will summarize the work of each layer separately.

A) Convolutional Layers: The convolutional layer (conv. layer) is made up of many feature maps, each of which is created by convolving a small region in the input data known as local receptive field. Different types of data like image, text that can be used the convolution. Such as in the image, an area of pixels is convolved as with our proposed system in this paper, and in the text, words or a set of the characters are convolved [17][19]. Images are usually fixed in its nature, i.e., the structure of one portion of the image is selfsame as any other portion. Therefore, feature is learnt in one portion could match like pattern in another portion.

A small section of a large image is captured and passed through all of the points in the image (Inputting). They are convolved into a single position while passing through any point (Outputting). Kernel (filter) [17] refers to any little component of an image that passes over the larger image. Later, back propagation methods were used to build the filters. A new feature map

is produced through the sliding of a local receptive field over data, as seen in Figure 3. Matrix multiplication is the mathematical operation used to achieve this task. The outcome is determined by [18]:

$$W = \frac{(W-F+2P)}{S+1} \dots \dots (1)$$

F: Filter Length
 P: Padding
 S: Stride

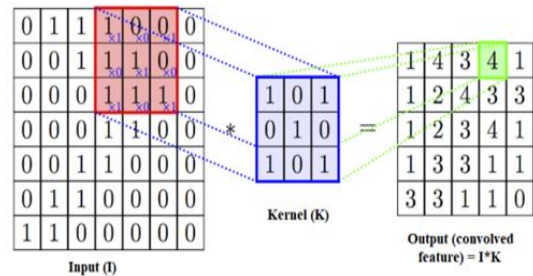


Figure 2: Convolutional Layer

B) Pooling: Feature motifs, which appear due to the convolution technique, can appear in the image at various locations. Once features have been retrieved, their precise location becomes less crucial as long as their relative position to others is kept. Sub-samplings are another name for such layers [17] [20]. After each convolution layer is completed, the pooling layers are removed. It gathers comparable data in the receptive field's neighborhood and produces the dominant response in this region [20]:

$$Z_l^k = g_p(F_l^k) \dots \dots \dots (2)$$

This equation depicts the pooling operation, where Z_l^k denotes the pooled feature-map regarding the l^{th} layer for k^{th} input feature-map $k F_l^k$, and $(.)_p$ denotes the pooling operation type. Pooling can be divided into two types: average pooling and max pooling. The latter takes the greatest values of a region's pixels, while the former takes the average value. Since it takes the maximum value from the picked kernel, the pooling layers decrease the outputting layer from the inputting layer. As a result, using more pooling layers reduces the image size and prevents it from shrinking [17][18]. Padding is utilized to pad the image according to the required pad size surrounding the image. Figure 3 depicts the average and max pooling operations.

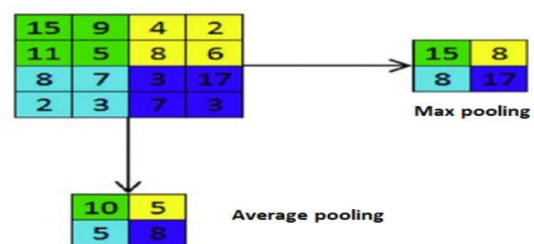


Figure 3: Pooling Layer (Max Pooling & Average Pooling)

C) Fully connected layers: Pooling layers are followed by FC layers. The third and final part of CNN, as depicted in Figure 4, are essentially FC layers. This layer takes the input from all of the neurons in preceding layer and performs operations on every one of the neurons in the current layer to generate output [17]. FC layer also assists to compose a numeral value of the image, that is means specifying the image with a probability value.

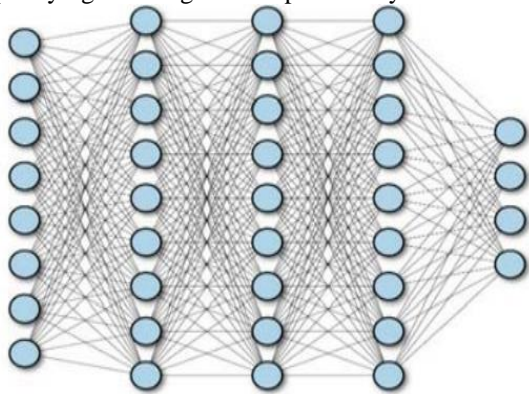


Figure 4.: Fully connected layer

4 Database of system

The dataset has 2450 images. It includes seven types of citrus disease. Each class contains over 150 images, 1470 for training and 980 for validation(tasting). The names of the dataset classes are: Anthracnose, brown rot, Citrus black spot, Citrus Canker, citrus scob, melanose, Sooty Mold citrus. All classes' images are collected with variable, realistic background. The Figure (5) is shown Some samples images of citrus diseases

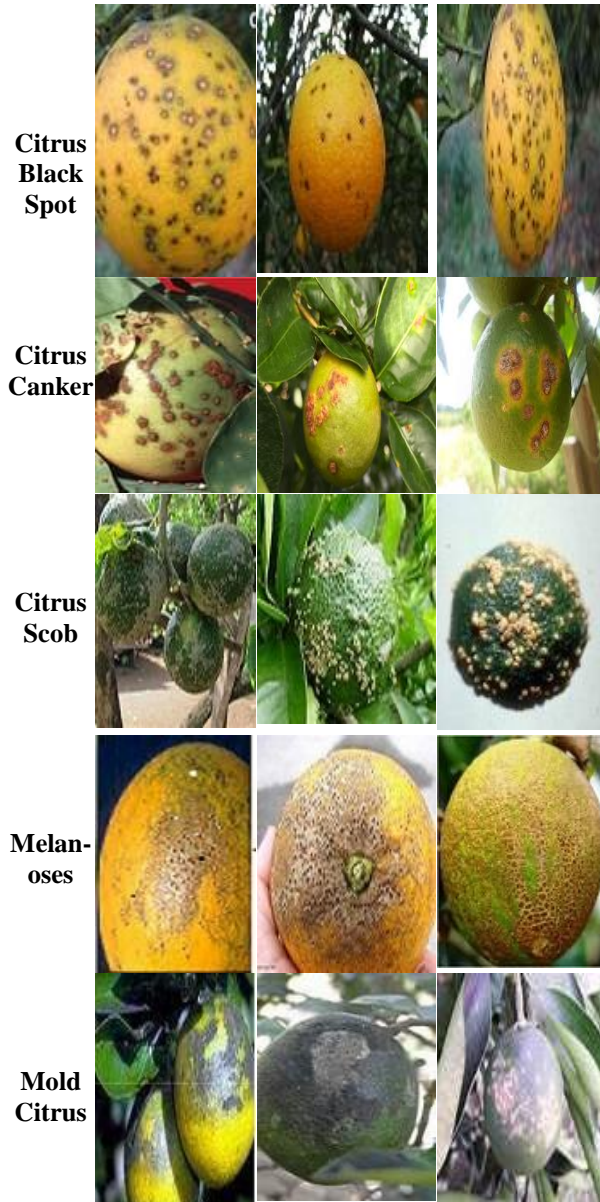
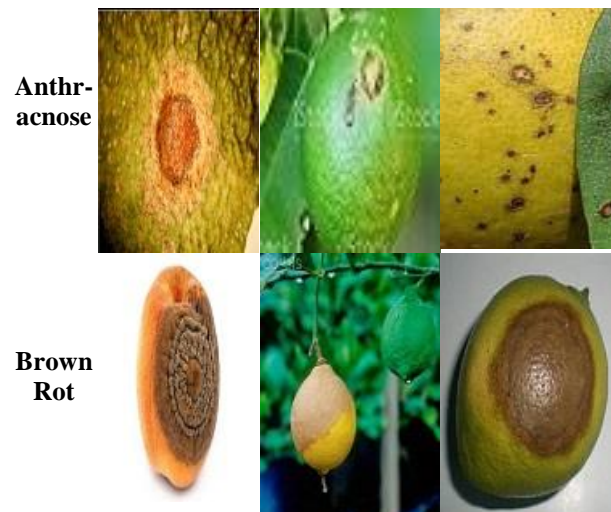


Figure 5: Examples of Database Models

5 Building CNN Model Citrus Dataset

The CNN is applied for image detection and image classification. Such as Neural Networks (NN), CNN also draws impulse from brain. We utilized object recognition model that proposed by Hubel and Wiesel. We layout CNN to recognize citrus diseases straight from pixel images. In our proposed system we applied roughly the same general architecture as CNN. Figure 6 is shown the general structure of CNN in this proposed system.

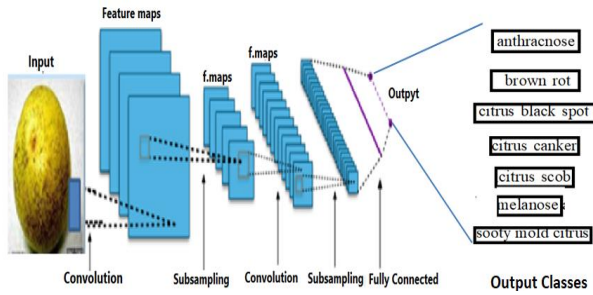


Figure 6: General structure of CNN model

6 Description of the system implementation

Deep Learning is an important tool for solving self-perception problems such as understanding images and classes them. In this paper we objective for Applying the idea of the CNN for the detection citrus diseases for data that contains 2450 images which divided images for testing and training images. In this section we explained the structure of this system by using CNN and the all results.

Throughout training, ML and supervised DL use inputs and outputs to generate data patterns or rules. Understanding the model's data patterns or rules helps us understand how the outputs were generated from the input data, as seen in Figure 7.

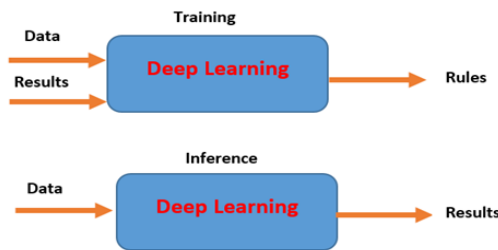


Figure 7: Results derived from the input data

6.1 Training Stage

As an input to the training, CNN gathers an image and its class from a database dedicated to training. We acquired trained weights, which are the rules or data patterns extracted from the images, as a result of the training. The image will be the only input to the trained model in the prediction, and the trained model will return the image's class. The image's class will be determined based on the data patterns learned throughout training. The basic stages for creating a simple CNN image training model are as follows:

1- Before the CNN model training, the Fully Connected Neural Network basics must determine of this dataset. Then in this work the must Specify the CNN architecture. In this work, the description of a convolution layer model as the following architecture:

- INPUT :150×100×3
- CONV3 :3x3 size,8 filters, 1 stride
- ReLU: $\max(0, h\theta(x))$

- POOL :2x2 size,2 stride
- CONV3: 3x3 size,16 filters,1 stride
- ReLU: $\max(0, h\theta(x))$
- POOL :2x2 size,2 stride
- CONV3: 3x3 size,32 filters,1 stride
- FC :7 Output Classes

2- Determine some of Options of the training network include:

- Solver Name: (sgdm or rmsprop or adam), As explained below in 3 steps.
- Batch Size for each iteration: 128
- Initial Learn Rate:0.001

3- Train the network: The network is trading according to the specified parameter in steps 1 and 2. During the training the ConvNet goes through several epochs, to adjusting its parameter to becomes better at classifying the images of training, then the output here is the trained network to be classification problem ConvNet, Update the network learnable parameters is don using the following function:

- the stochastic gradient descent with momentum (SGDM) algorithm: in the (SGDM) is variants were the dominating optimization techniques, SGD with large batch training keeps attracting more and more attention [21]:

$$v_t = \gamma v_t + \eta \nabla_{\theta} J(\theta) \dots \dots (2)$$

$$\theta_t = \theta_{t-1} - v_t$$

- using the root mean squared propagation (RMSProp) algorithm: is a Neural Network (ANN) training, for each of the parameters the learning rate is adapted. The concept is to divide a weight's learning rate by a running mean of recent gradient magnitudes for that weight [21]. To begin, running average is estimated based on the means square:

$$v(w, t) := \gamma v(w, t - 1) + (1 - \gamma)(\nabla Q_i(w))^2 \dots \dots (3)$$

In which, γ represent the forgetting factor, and parameters are updated based on,

$$w = w - \frac{\eta}{\sqrt{v(w, t)}} \nabla Q_i(w) \dots \dots (4)$$

- using the adaptive moment estimation (Adam) algorithm.: is a gradient descent-based algorithm of learning has been based upon 1st- and 2nd- order statistical moments, in other words, mean and variance. For the parameters $w^{(t)}$ and a loss function $L^{(t)}$, where t denotes current training iteration (which is indexed at 1), Adam's parameter update has been represented as:

$$m_w^{(t+1)} \leftarrow \beta_1 m_w^{(t)} + (1 - \beta_1) \nabla_w L^{(t)} \dots \dots (5)$$

β_1 is the forgetting factors for gradients and second moments of gradients [22] - [26].

6.2 Testing Stage

After training the CNN, now a test dataset to verify its accuracy. The test dataset is a set of images and labeled images. Each image is testes through the trained ConvNet, and the output is compared with the label of this

image. The following paragraphs explain the most important results obtained from implementing CNN on our Dataset with a change the learning function. Table1 shows the final results of all experimentations for 10 epochs

Table1: Accuracy Results for 10 epochs

Epochs	Accuracy (sgdm)	Accuracy (rmsprop)	Accuracy (adam)	Maximum iteration
1	0.4693	0.5086	0.6301	17
2	0.6535	0.7109	0.7572	34
3	0.7341	0.7572	0.8786	51
4	0.7687	0.7687	0.8381	68
5	0.8265	0.8092	0.8034	85
6	0.8323	0.8150	0.8959	102
7	0.8323	0.8323	0.8034	119
8	0.8612	0.7745	0.8670	136
9	0.8768	0.8092	0.8497	153
10	0.8762	0.8381	0.8382	170

6.3 Discussion and experimental results

We will review the most important results that obtained during the experiments, and the must clarify here that in all figure bellow:

- In A: The blue line: represent training accuracy
The black line: for the validation accuracy,
- In B: The red line refers to the training loss
And black line: indicates validation loss

- CNN model training and with SGDM: The Figure 8 Show the Accuracy result with Stochastic gradient descent with momentum (Sigm function). In Figure 8 A, the training set and validation accuracy It gets better with the progress the number of epoch, but the testing accuracy low compare with training, because the training loss in figure 8 is approached zero over time, and the validation loss still quite not low. The Figure 9 show the accuracy rate of each class separately.

After trying to Apply transformations (using randomly picked values) and build augmented data store. Specify data augmentation options and values/ranges. Figure 10 Show the Accuracy result for ten with Sigm algorithm after apply Applying transformations. We note that the results here for the training and validation set are less compare without augmentation, as we note that the loss function does not approach zero in both cases.

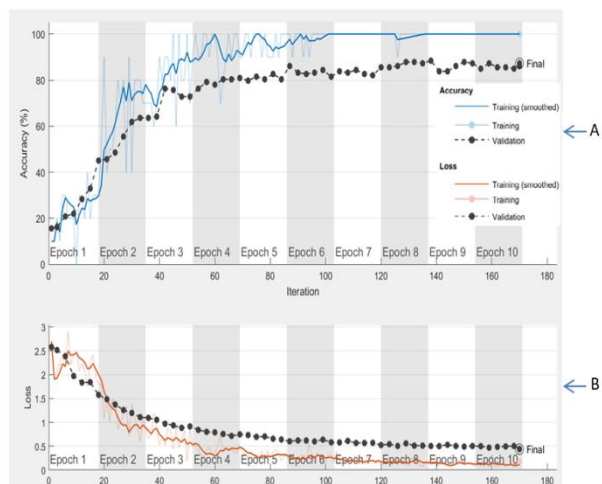


Figure 8: Accuracy result with momentum (Sigm) training

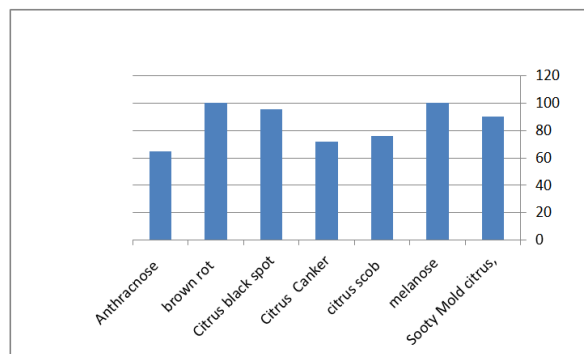


Figure 9: The success rate of each class based on applying sigma function

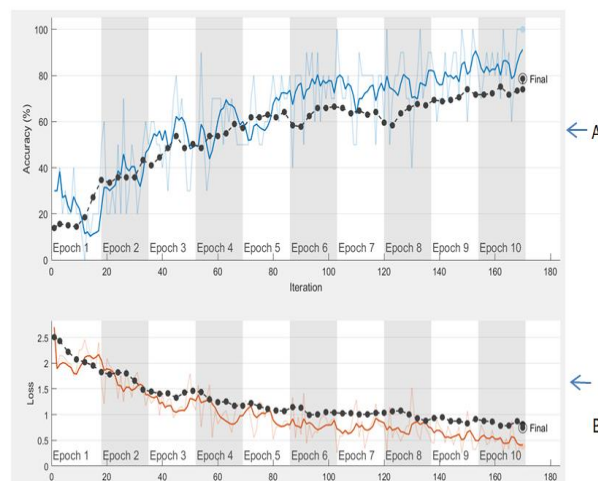


Figure 10: The Accuracy result with Sigm and Applying transformations

Training with Root mean square propagation (RMSProp): The Figure (11) Show the Accuracy result by depending Stochastic gradient descent with momentum. We see that the result of validation accuracy is here fall down slightly

compare with SGDM training, and also the validation accuracy is low slightly compare with training accuracy with time where the results of loss function indicate and explain these results. Figure (12) show the accuracy of each class.

Also, After Applying the Randomly Transformations the results are shown in Figure (13), where the accuracy is fall down compare with result in Figure (11) as well as very clear of the accuracy and loss function

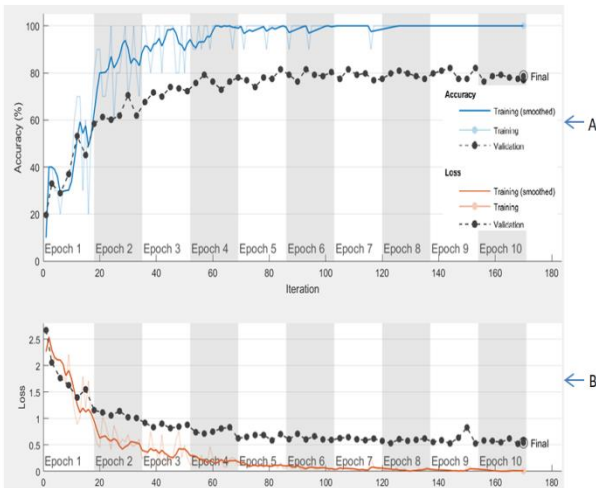


Figure 11: The Accuracy result with Root mean square propagation training

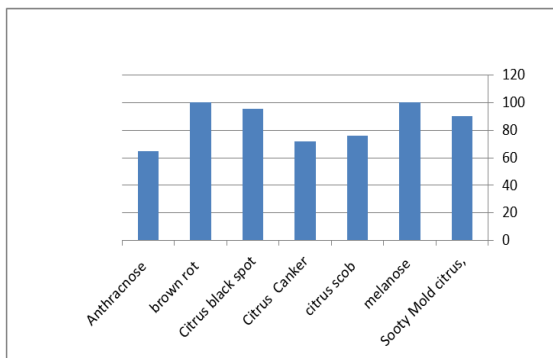


Figure 12: The success rate of each class based on applying (RMSProp)

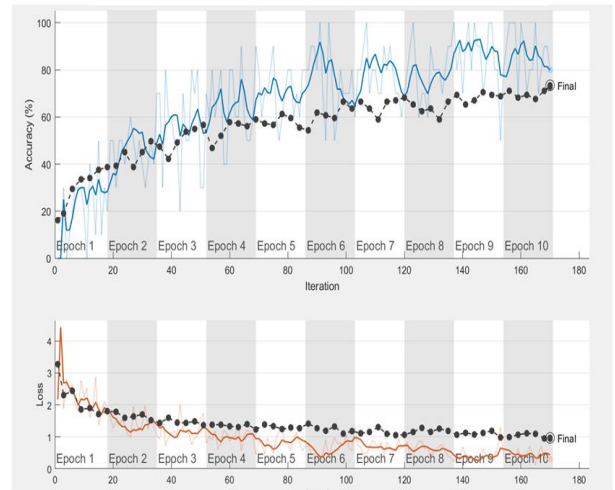


Fig 13: The Accuracy result and Root mean function after Applying transformations

Training with Adaptive moment estimation (ADAM): Figure (14) and Figure (15) Show the overall results, and it is noticeable that the accuracy result are converges the results accuracy that obtain of (RMSProp). Figure (16) show the result after the apply the transformation. We noticed that the accuracy results (for training and validation set) are approximately identical over the progresses of run time.

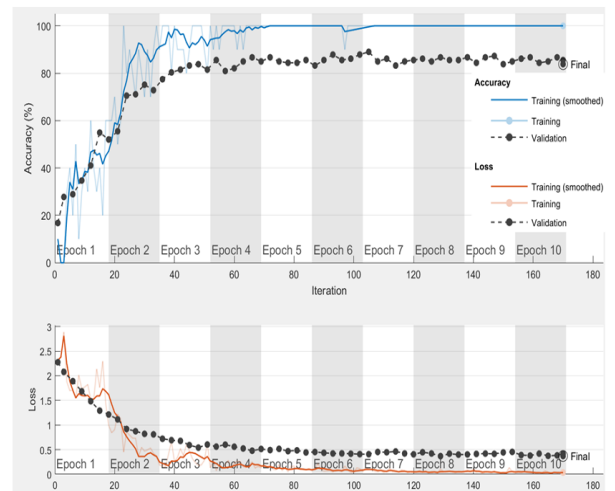


Figure 14: The result accuracy with (ADAM)

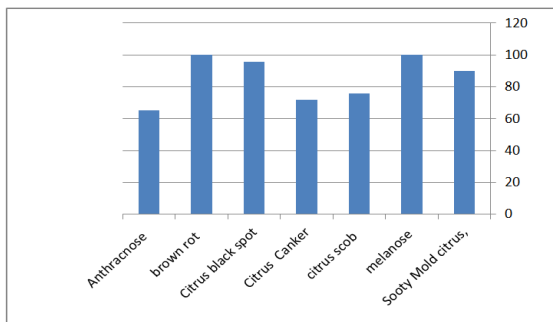


Figure 15: The success rate of each class based on applying (ADAM)

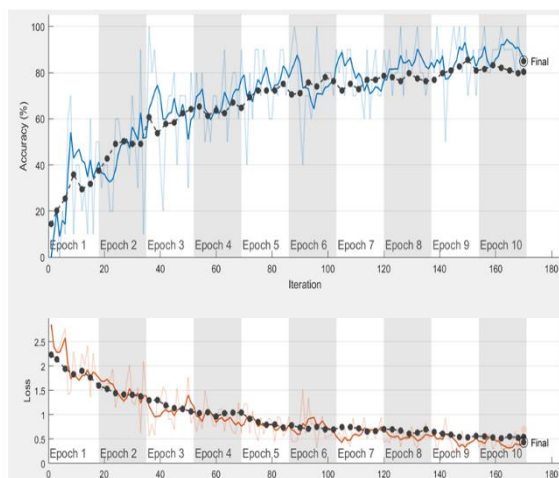


Fig 16: The Accuracy result with (ADAM) after Applying transformations

7 Conclusion and future work

In this paper, a model is presented that can identify and classify citrus diseases automatically based on the CNN model. What distinguishes this work is that the training of the model was applied using a database of 2450 images that were taken in different real conditions in the fields of agriculture, where the database consists of seven categories of citrus diseases, which contains 150 images each, such as anthracnose, and mold Brown, citrus black spot, citrus canker, citrus scoop, melanosis, sooty mold. According to the experiments conducted in this work, it was clear that working with the CNN model gives great results, as explained in the above paragraphs. Also, this work is distinguished by experimenting with many measurements with the CNN model in order to make the results of classification of diseases / outputs more accurate, for example by increasing the number of hidden neurons and convolution layers.

The average success rate is 88%, as the classification efficiency was observed even with the change of the learning algorithm. The results given by CNN with all algorithms are good and semi-closed, but the training with the root mean squared propagation (RMSProp) showed the accuracy slightly better compared to (RMSProp) and (ADAM) and with the same parameters.

Also, the overall score was reduced by a certain percentage with transformations applied using randomly picked values to build multiplying data.

In the future, we plan to develop a real-time citrus disease recognition system by taking live images of citrus and directly identifying the type of disease affecting citrus with choosing different CNN architectures.

References

- [1] Li, K., Chen, M., Lin, J., & Li, S. (2019, November), Citrus Disease and Pest Recognition Algorithm Based on Migration Learning, In International Symposium on Intelligence Computation and Applications (pp. 3-20). Springer, Singapore. <https://doi.org/10.1016/j.ijbiomac.2018.11.166>.
- [2] Ferentinos, K. P. (2018), Deep learning models for plant disease detection and diagnosis, Computers and Electronics in Agriculture, 145, 311-318. <https://doi.org/10.1016/j.compag.2018.01.009>.
- [3] Tian, L. G., C. Liu, Y. Liu, M. Li, J. Y. Zhang, and H. L. Duan. (2020), "Research on plant diseases and insect pests identification based on CNN." In IOP Conference Series: Earth and Environmental Science, vol. 594, no. 1, p. 012009. IOP Publishing. <https://doi.org/10.1088/1755-1315/594/1/012009>
- [4] Bottleson, J., Kim, S., Andrews, J., Bindu, P., Murthy, D. N., & Jin, J. (2016, May), clcaffe: Opencl accelerated caffe for convolutional neural networks, In 2016 IEEE international parallel and distributed processing symposium workshops (IPDPSW) (pp. 50-57). IEEE., <https://doi.org/10.1109/DSMP.2018.8478621>.
- [5] Pan W., Qin J., Xiang X., Wu Y., Tan Y., & Xiang, L. (2019), "A smart mobile diagnosis system for citrus diseases based on densely connected convolutional networks.", IEEE Access 7 (2019): 87534-87542, <https://doi.org/10.1109/ACCESS.2019.2924973>.
- [6] Manavalan, R. "Automatic identification of diseases in grains crops through computational approaches: A review." Computers and Electronics in Agriculture 178 (2020): 105802. <https://doi.org/10.1016/j.compag.2020.105802>.
- [7] Liu, Z., Xiang, X., Qin, J., Ma, Y., Zhang, Q. and Xiong, N.N., "Image Recognition of Citrus Diseases Based on Deep Learning." CMC-COMPUTERS MATERIALS & CONTINUA 66, no. 1 (2021): 457-466, <https://doi.org/10.32604/cmc.2020.012165>.

- [8] Rauf, H.T., Saleem, B.A., Lali, M.I.U., Khan, M.A., Sharif, M. and Bukhari., "A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning." *Data in brief* 26 (2019):104340. <https://doi.org/10.1016/j.dib.2019.104340>.
- [9] Sharif, M., Khan, M.A., Iqbal, Z., Azam, M.F., Lali, M.I.U. and Javed, M.Y., "Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection." *Computers and electronics in agriculture* 150 (2018): 220-234, <https://doi.org/10.1016/j.compag.2018.04.023>.
- [10] Ali, H., Lali, M. I., Nawaz, M. Z., Sharif, M., & Saleem, B. A.. Saleem. "Symptom based automated detection of citrus diseases using color histogram and textural descriptors." *Computers and Electronics in agriculture* 138 (2017): 92-104, <https://doi.org/10.1016/j.compag.2017.04.008>.
- [11] Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M. and Farhan, L., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions." *Journal of big Data* 8, no. 1 (2021): 1-74, <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00444-8>
- [12] Krizhevsky, A., Sutskever, I., & Hinton, G. E., "ImageNet classification with deep convolutional neural networks." *Communications of the ACM* 60, no. 6 (2017): 84-90, <https://doi.org/10.1145/3065386>.
- [13] Hossain, Md Anwar, and Md Shahriar Alam Sajib. "Classification of image using convolutional neural network (CNN)." *Global Journal of Computer Science and Technology* (2019), <https://doi.org/10.34257/GJCSTDVOL19IS2PG13>.
- [14] A. Qayyum, C. K. Ang, S. Sridevi, M. K. A. A. Khan, L. W. Hong, M. Mazher, T. D. Chung, "Hybrid 3D-ResNet Deep Learning Model for Automatic Segmentation of Thoracic Organs at Risk in CT Images," 2020 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), pp. 1-5, 2020, <https://doi.org/10.1109/ICIEAM48468.2020.9111950>.
- [15] Phan, T.H., Tran, D.C. and Hassan, M.F., "Vietnamese character recognition based on CNN model with reduced character classes." *Bulletin of Electrical Engineering and Informatics* 10, no. 2 (2021): 962-969, <https://doi.org/10.11591/EEI.V10I2.2810>.
- [16] Khan, Salman, et al. "A guide to convolutional neural networks for computer vision." *Synthesis lectures on computer vision* 8.1 (2018): 1-207. <https://doi.org/10.2200/s00822ed1v01y201712cov015>
- [17] Sultana, F., Sufian, A., & Dutta, P., "Advancements in image classification using convolutional neural network." 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN). IEEE, 2018, <https://doi.org/10.1109/ICRCICN.2018.8718718>.
- [18] Zhou, Yu, Haipeng Wang, Feng Xu, and Ya-Qiu Jin. "Polarimetric SAR image classification using deep convolutional neural networks." *IEEE Geoscience and Remote Sensing Letters* 13, no. 12 (2016): 1935-1939, <http://dx.doi.org/10.1109/LGRS.2016.2618840>.
- [19] Mou, L. and Jin, Z., "Tree-Based Convolutional Neural Networks": Principles and Applications. Springer, 2018. <https://vdoc.pub/documents/tree-based-convolutional-neural-networks-principles-and-applications-4cfkkikos1c0>
- [20] Khan, A., Sohail, A., Zahoor, U., & Qureshi, A. S., "A survey of the recent architectures of deep convolutional neural networks." *Artificial Intelligence Review* 53, no. 8 (2020): 5455-5516, <https://doi.org/10.1007/s10462-020-09825-6>.
- [21] Zhao, S. Y., Xie, Y. P., & Li, W. J., "Stochastic Normalized Gradient Descent with Momentum for Large Batch Training." *arXiv preprint arXiv:2007.13985* (2020), <https://doi.org/10.48550/arXiv.2007.13985>
- [22] Kingma, D. P., & Ba, J., "Adam: A method for stochastic optimization". *arXiv:1412.6980*, (2014), <https://doi.org/10.48550/arXiv.1412.6980>
- [23] Tian, L. G., Liu, C., Liu, Y., Li, M., Zhang, J. Y., & Duan, H. L., "Research on plant diseases and insect pests identification based on CNN." In *IOP Conference Series: Earth and Environmental Science*, vol. 594, no. 1, p. 012009. IOP Publishing, 2020. <https://doi.org/10.1088/17551315/594/1/012009>
- [24] Wala'a, N. J., & Rana J. M., (2021). A Survey on Segmentation Techniques for Image Processing, *Iraqi Journal for Electrical and Electronic*

- Engineering, vol. 17 , pp. 73-9,
<https://doi.org/10.37917/ijeee.17.2.10>.
- [25] Saddam, Saba Abdual Wahid., (2022). "Wind Sounds Classification Using Different Audio Feature Extraction Techniques." Informatica 45, no. 7
<https://doi.org/10.31449/inf.v45i7.3739>
- [26] Nemer, Z.N., 2022. Hand Gestures Detecting Using Radon And Fan Beam Projection Features. Informatica, 46(5).
<https://doi.org/10.31449/inf.v46i5.3744>