e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

PLANT LEAF DISEASE DETECTION AND SIMULTANEOUS PESTICIDE DISTRIBUTION USING MOBILENET CNN ARCHITECTURE AND RASPBERRY PI

Ritayan Patra*1, Debmalya Chatterjee*2, Anchit Kumar Ghosh*3

*1,2,3School Of Electronics Engineering, Vellore Institute Of Technology, Vellore, Tamil Nadu, India.

DOI : https://www.doi.org/10.56726/IRJMETS34030

ABSTRACT

Agriculture is one most important occupation around the world. It is a prime source of income for a huge number of people around the world. It is like a treasure house of a country. The farmers are the treasurer of this treasure house. However, a huge threat persists in agriculture. Every year crops are attacked by a number of diseases leading to poor agricultural output. This creates a domino effect involving loss of livelihood of farmers, increase in price of essential food items, scarcity of food and decline in agricultural economy. The quick detection of the disease in the crops is of utmost importance. In this paper, the authors have used deep learning technique to fully automate the detection of disease in crops as well as speedy distribution of required pesticides in the fields. The authors have used Convolutional Neural Network (CNN) along with transfer learning to detect the type of disease and RPi microcontroller is used to spray the pesticide. The deep model is trained on wide range of plants with high accuracy.

Keywords: Agricultural Automation, CNN, Computer Vision, Deep Learning, Leaf Disease Detection, Raspberry Pi.

I. INTRODUCTION

The detection of plant disease at an early stage leads to an efficient yielding of crops. There are many categories of plant diseases ranging from common rust, black rot to bacterial spot and many more [1]. All these diseases inadvertently affect the growth of important crops and degrades the quality of the crop which in turn affects the economic output of the agriculture industry. The current solution of avoiding this adverse impact is the use highly expensive methods like excess spraying of toxic pesticides and fertilizers [2]. The excessive use of harmful chemicals causes extensive damage to the environment as well as it may enter the food chain creating health hazards in human beings as well as in other animals. Moreover, these methods increase the cost of producing and managing of crops for the farmers manifolds.

The early detection of plant disease is of utmost importance in order to have a good agricultural output. Farmers are expert in detecting the type of disease but it is a time-consuming process [3]. The use of modern technology greatly reduces the human effort and save a lot of time. The authors in this study have investigated about different plant diseases and have use deep learning technology to build a model which will instantly detect the type of disease and will auto recommend the treatment of such diseases [4].

II. LITERATURE SURVEY

In paper [1], the authors have used deep learning to recognize and classify various grape, apple, tomato and sugarcane diseases. Data augmentation techniques were used to diversify the images and the deep model was trained on them for 75 epochs and an accuracy of 96.5% was achieved. In paper [2], the authors have used MATLAB to continuously monitor rice leaf disease and the nutrient deficiencies. The use of Raspberry Pi and many other sensors facilitate in relaying of essential information to the users using an android application. The authors have used the concept of Convolutional Neural Network along with Learning Vector Quantization (LVQ) algorithm to classify various diseases occurring in tomato plant in paper [3] and have achieved an average accuracy of 86%. In paper [4], the authors have used the concept of e-nose to detect foul smell coming from plants in order to detect disease using principal component analysis algorithm from machine learning. The paper [5] describes the internal and complex working components of deep learning in context with convolutional neural network. The authors of paper [6], [7] and [8] have used various convolutional neural

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

networks like Alex Net, Squeeze Net, YOLO-v3 as well as machine learning algorithm to detect diseases in tomato, sugarcane and many other plants. The concept of transfer learning on diseased plant is elaborated in paper [9] and [10]. The MobileNet architecture based on convolutional neural network is elaborately described in paper [11] with its application on detecting diseases and pests is presented in paper [12]. The paper [13] also describes the working of MobileNet architecture and its internal structure and how it allows users to generate memory efficient deep models without compromising much on the efficiency of the deep models. The paper [14] elaborates on working with Raspberry Pi and how it can be accessed from anywhere in the world to facilitate the concept of Internet of Things in real world scenario.

III. CONVOLUTIONAL NEURAL NETWORK

Artificial Intelligence is growing at a rapid speed and entering into every aspect of human lives. Many researchers are working in this field to reduce the gap between machines and humans. One of the ways to do so to make machine learn and perceive its surrounding just like a human would do. The advancement of Deep Learning in the field of Computer Vision led to a development of an algorithm known as the Convolutional Neural Network (CNN or CovNet) [5][6].

The CNN algorithm takes an image as input and tries to learn various objects that are present in the image by learning the weights and biases of that object. The CNN algorithm is built in such a way that it would act like the neurons of a human brain. The visual cortex in human brain receives and processes visual data relayed by the retinas in the eye. The CNN algorithm acts just like the visual cortex and try to learn and distinguish various objects.

The CNN architecture generally involves four-layer structure, i.e., Convolutional Layer, Pooling Layer, Activation Layer, Fully Connected Layer [7][8].

A. Convolutional Layer

The input image is nothing but a bunch of numbers stored in a form of a matrix. In this layer, a mathematical operation known as the convolution operation is utilized in order to learn the most important features or objects present in the image. The element which is responsible for the convolution operation is known as filter or kernel. The filter generally has small dimensions than the image. This filter will shift over the image matrix with various strides length and will perform matrix multiplication over that portion of the matrix and the result is added with the help of bias to give a convoluted output as shown in Fig. (1) and Fig. (2). There can be many convolutional layers in an CNN architecture. Generally, the first few convolutional layers learn about the edges, colour or gradient orientation of the image while the rest of them try to learn more high-level features leading to good understanding of what the image is all about.

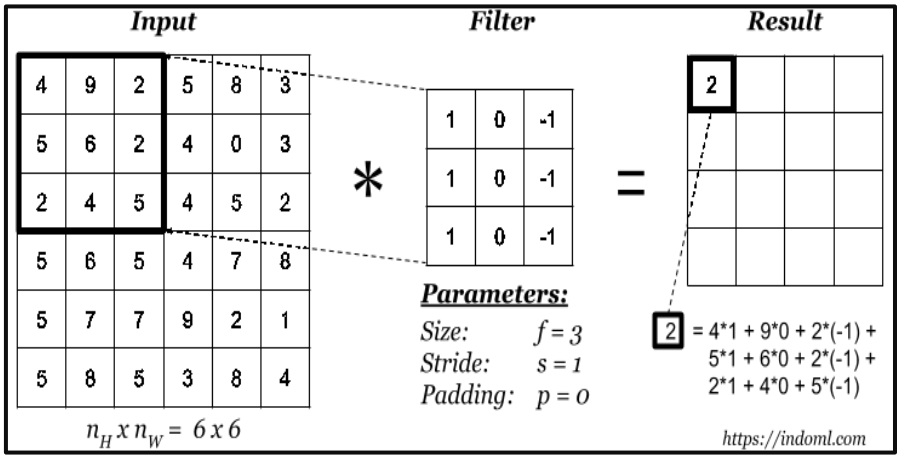

Figure. 1. First Step in Convolutional Operation on Image Matrix

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

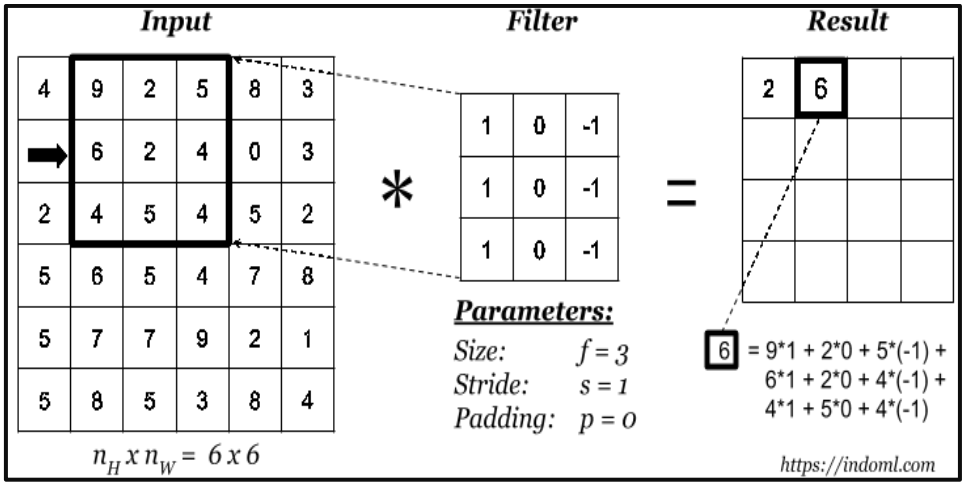

Figure. 2. Second Step in Convolution Operation on Image Matrix

B. Pooling Layer

The Pooling Layer is used to reduce the size of convoluted output. Reduction in size leads to reduce usage of computational power. Moreover, it may be used to learn more dominant features from particular portions of the image. There are generally two types of pooling: Average Pooling and Max Pooling. In both pooling operation a kernel similar to that of the convolutional layer is used. In Average Pooling, the kernel hover over different portions of the convoluted output matrix and returns the average of all the values as shown in Fig. (3) while in the case of Max Pooling, the kernel returns the max value from the hovering region of the convoluted matrix as shown in Fig. (4).

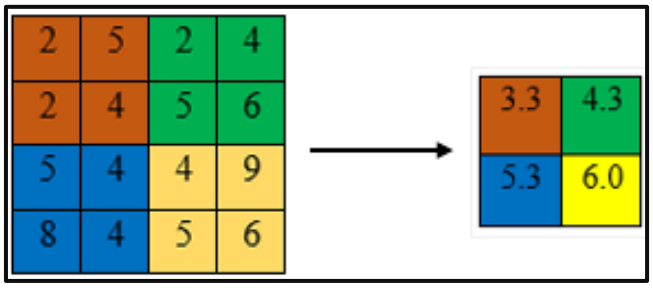

Figure. 3. Average Pooling

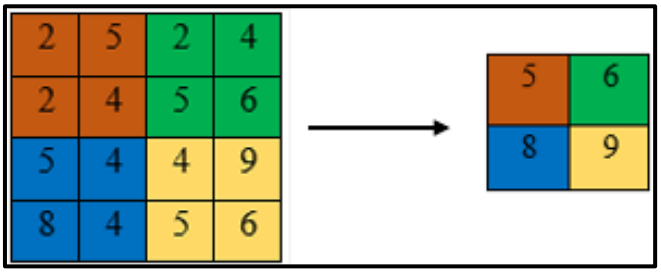

Figure. 4. Max Pooling

C. Activation Layer

The activation layer contains an activation function which is used to insert non-linearity in the model. It is done to prevent overfitting of the model. There are many activation functions like ReLU, TanH, Sigmoid, etc as shown in Fig. (5) and Fig. (6). Sometimes to prevent overfitting the application of dropout layers is also applied to the model.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

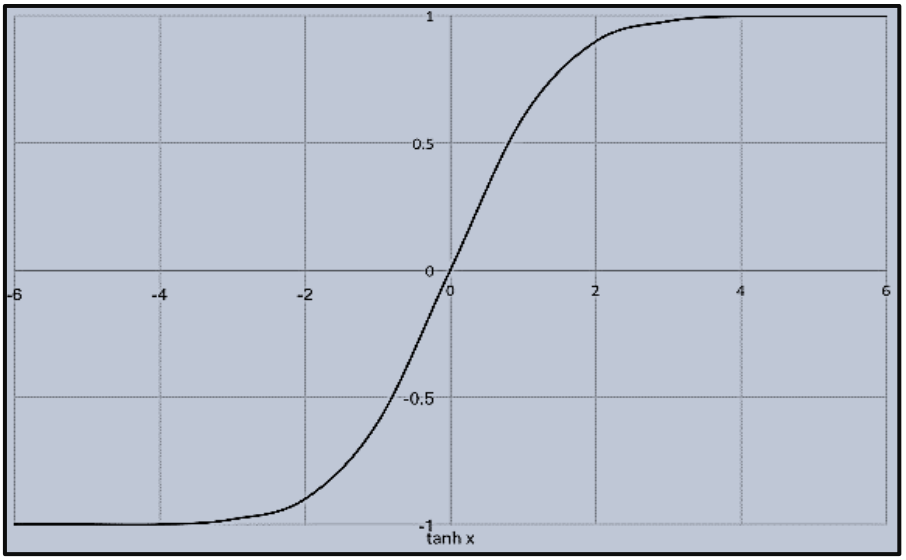

Figure. 5. TanH Activation Function

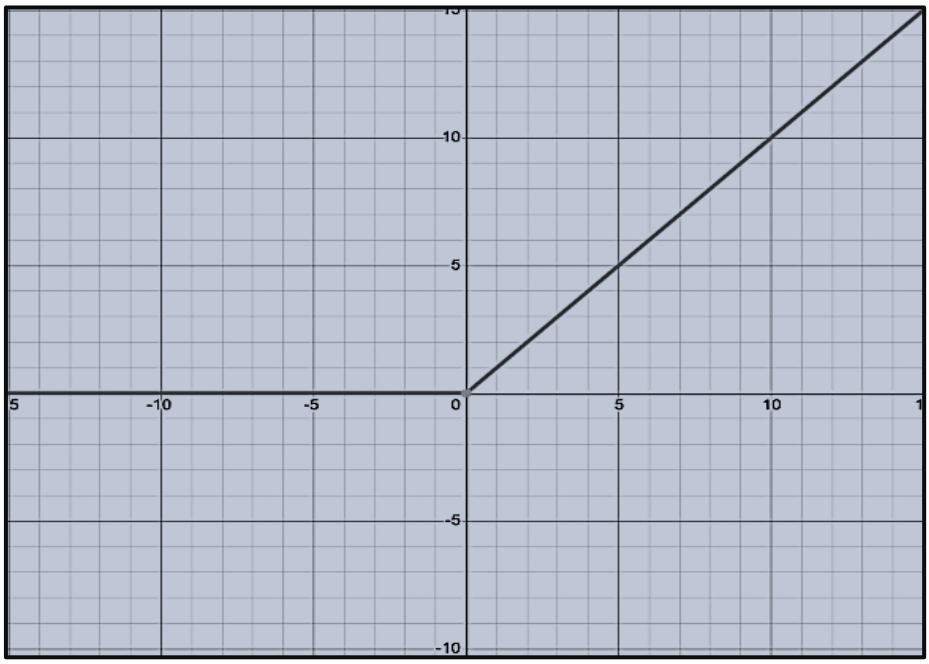

Figure. 6. Rectified Linear Unit (ReLU) Activation Function

D. Fully Connected Layer

The Fully Connected Layer or FC layer is used to analyse different class of probabilities obtained by analysing various objects or features of the image. After passing the image through various layers of convolutional and pooling layers, it is flattened to a single dimension matrix and is fed into the FC layer as shown in Fig. (7). This flatten layer output is fed into a feed-forward neural network and backpropagation technique is used to fine tune various parameters and weights. The model is trained over many epochs and it try to distinguish between low level and dominating features using well known SoftMax Classification algorithm.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

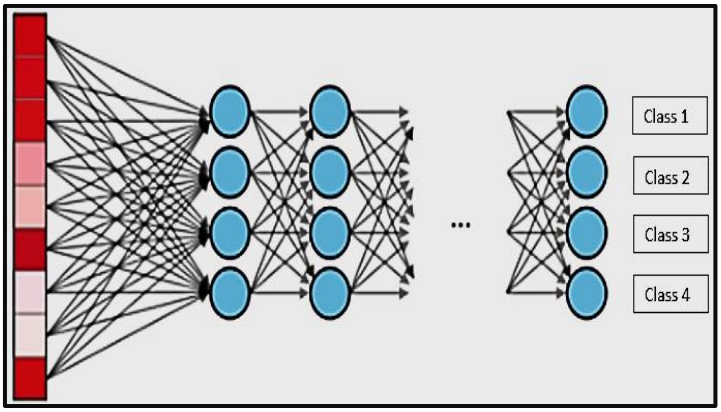

Figure. 7. Fully Connected Layer

IV. TRANSFER LEARNING

Transfer Learning is a new technique used in machine learning and deep learning where a model which trained for one task is repurposed for a different but similar related task. In this technique, researcher need not required to train a model from scratch rather use the already available and proven deep models and repurpose them for specific tasks. This technique saves a huge amount of time as well as computational power [9][10].

In this study the authors have used MobileNet CNN architecture for creating an efficient model to detect plant disease. MobileNet is designed by Google researchers in such a way that it can be used in mobile applications and it is easily portable. It uses a depth wise separable convolution technique which greatly decreases the number of parameters without compromising much on the accuracy of the model. The architecture of MobileNet is given in Fig. (8) and Fig. (9). It is very small in size and is also quite fast when used [11][12].

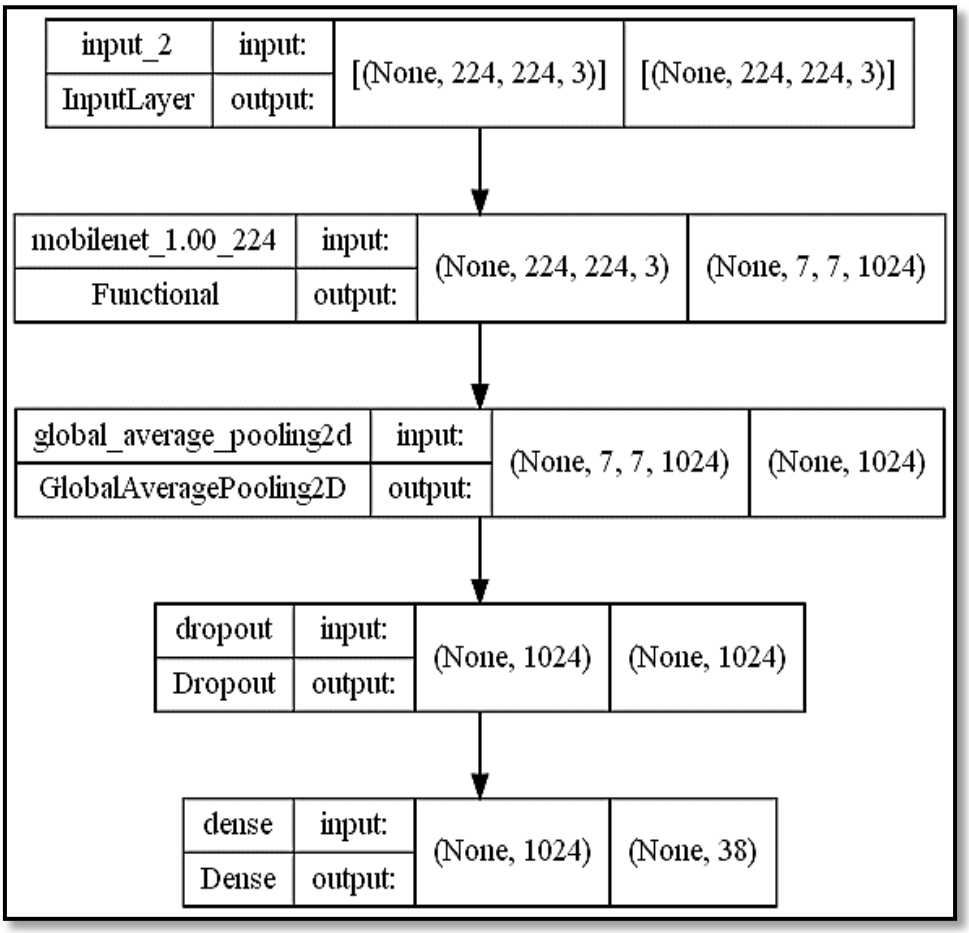

Figure. 8. The proposed Deep Learning Model

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

Table 1. MobileNet Body Architecture

Sl. No.	Type / Stride		Filter Shape	Input Size
1.	Conv / s2		3 × 3 × 3 × 32	224 × 244 × 3
2.	Conv dw / s1		3 × 3 × 32 dw	112 × 112 × 32
3.	Conv / s1		1 × 1 × 32 × 64	112 × 112 × 32
4.	Conv dw / s2		3 × 3 × 64 dw	112 × 112 × 64
5.	Conv / s1		1 × 1 × 64 × 128	56 × 56 × 64
6.	Conv dw / s1		3 × 3 × 128 dw	56 × 56 × 128
7.	Conv / s1		1 × 1 × 128 × 128	56 × 56 × 128
8.	Conv dw / s2		3 × 3 × 128 dw	56 × 56 × 128
9.	Conv / s1		1 × 1 × 128 × 256	28 × 28 × 128
10.	Conv dw / s1		3 × 3 × 256 dw	28 × 28 × 256
11.	Conv / s1		1 × 1 × 256 × 256	28 × 28 × 256
12.	Conv dw / s2		3 × 3 × 256 dw	28 × 28 × 256
13.	Conv / s1		1 × 1 × 256 × 512	14 × 14 × 256
14.	5 ×	Conv dw / s1	3 × 3 × 512 dw	14 × 14 × 512
		Conv / s1	1 × 1 × 512 × 512	14 × 14 × 512
16.	Conv dw / s2		3 × 3 × 512 dw	14 × 14 × 512
17.	Conv / s1		1 × 1 × 512 × 1024	7 × 7 × 512
18.	Conv dw / s2		3 × 3 × 1024 dw	7 × 7 × 1024
19.	Conv / s1		1 × 1 × 1024 × 1024	7 × 7 × 1024
20.	Avg Pool / s1		Pool 7 × 7	7 × 7 × 1024
21.	FC / s1		1024 × 1000	1 × 1 × 1024
22.	Softmax / s1		Classifier	1 × 1 × 1000

V. METHODOLOGY

A. Image Acquisition

A good number of images are required to build an efficient deep model. Images belonging to different plants having different diseases is present in Kaggle website. The image dataset contains approximately 38,000 images belonging to 16 different plants. The images are further subdivided into different diseases.

B. Image Pre-processing

The images in the datasets are not at all uniform in nature. Images differ in sizes, colour, contrast and various other parameters. The images are pre-processed using a python script which fixes the size of images as well as augment them to specific orientation and contrast.

C. Model Training

A deep model is build using MobileNet architecture and it takes pre-processed images as input. The model learns all important parameters present in the images and try to distinguish diseased leaves from healthy ones. The model created by the authors has around 3 million mathematical parameters which are sufficient enough to learn important features present in the dataset [13]. The model summary is shown in Fig. (9).

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

```
Model: "LeafDisease_MobileNet"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 input_2 (InputLayer)         [(None, 224, 224, 3)]     0

 mobilenet_1.00_224 (Functio  (None, 7, 7, 1024)        3228864
 nal)

 global_average_pooling2d (G  (None, 1024)              0
 lobalAveragePooling2D)

 dropout (Dropout)            (None, 1024)              0

 dense (Dense)                (None, 38)                38950

=================================================================
Total params: 3,267,814
Trainable params: 38,950
Non-trainable params: 3,228,864
_________________________________________________________________
None
```

Figure. 9. MobileNet model summary

D. Classification

The plant leaves are classified into diseased and healthy. The model will also determine the type of disease and provide the required treatment if available. Moreover, the model can now work with unknown plant leaves and give the best possible result based on that unknown input.

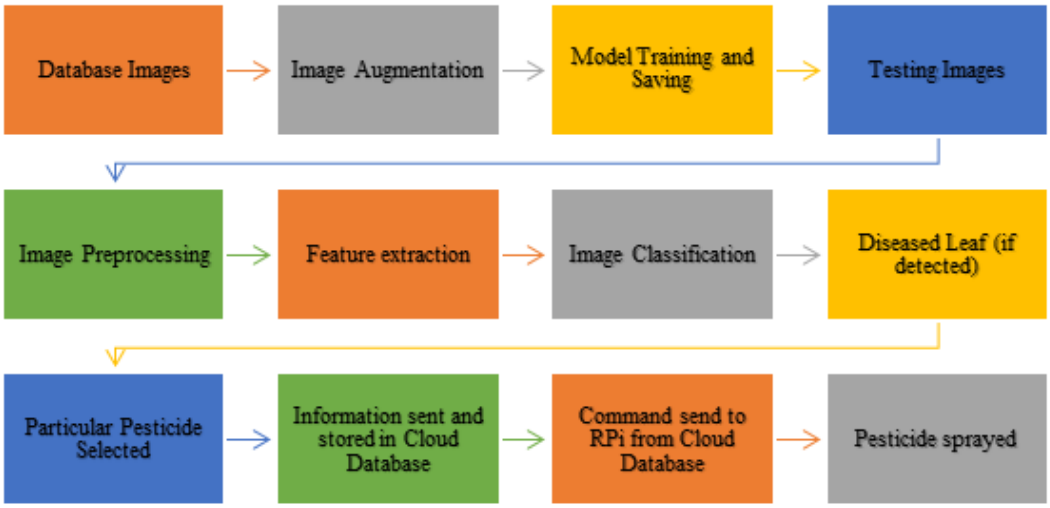

Figure. 10. Designing Steps

VI. EXPERIMENTAL SETTINGS

The size of the dataset is approximately 1.1 GB containing almost 38,000 images. Python programming language is used to build the deep model. Spyder IDE is used as development environment along with Anaconda Navigator. Important python packages like TensorFlow, NumPy, pandas and matplotlib are used. The images in the datasets are rotated, flipped or shifted in a particular direction in order to introduce non-linearity and randomness in the dataset to prevent overfitting. Adam optimizer, ReLU activation function and SoftMax classification algorithm are used in the model. The study is performed on HP Pavilion Gaming laptop having AMD Ryzen 5 processor with 8 GB RAM and 4 GB Nvidia GTX graphics card. The model is trained on GPU instead of CPU to reduce training time and to provide higher computational power. For the hardware implementation, the authors have used Raspberry Pi3, DC servo motor, LCD to display information and few jumper wires. The whole designing steps involved is shown in Fig. (10).

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

VII. RESULTS AND ANALYSIS

The deep learning model achieved 96.45% accuracy which is achieved for 10 epochs. The testing accuracy also known as validation accuracy on unknown data reached a high accuracy of 91.46%. The training and validation loss has decreased significantly. The accuracy and loss curves are plotted for 10 epochs and are shown in Fig. (11) and Fig. (12) respectively.

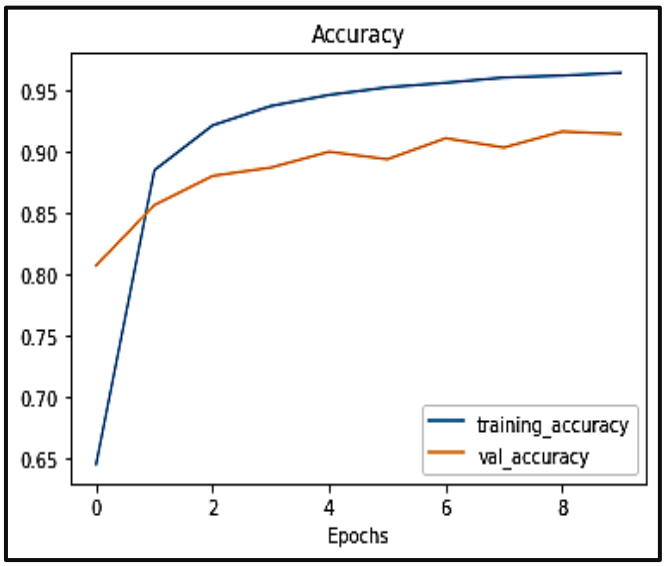

Figure. 11. Accuracy Plot of Deep Learning Model

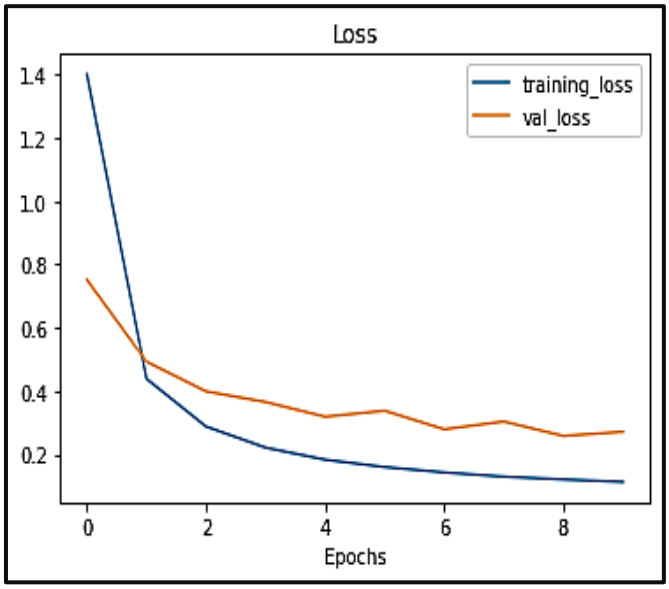

Figure. 12. Loss Plot of Deep Learning Model

The deep learning model is applied on some unknown testing data. In Fig. (13) and Fig. (14), the unknown plant disease has been predicted accurately. The model detects Apple scab disease with 98.192% confidence. Similarly, in Fig. (15) and Fig. (16), Common rust in corn plant is predicted with 99.965% confidence while in Fig. (17) and Fig. (18), Tomato Yellow Leaf Curl Virus is detected with 99.534% confidence. The required treatment for each of these diseases is also prescribed simultaneously by the model facilitating swift response. The model detects the disease with least human interaction leading to early detection of the disease.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

Figure. 13. Apple Scab Plant Disease

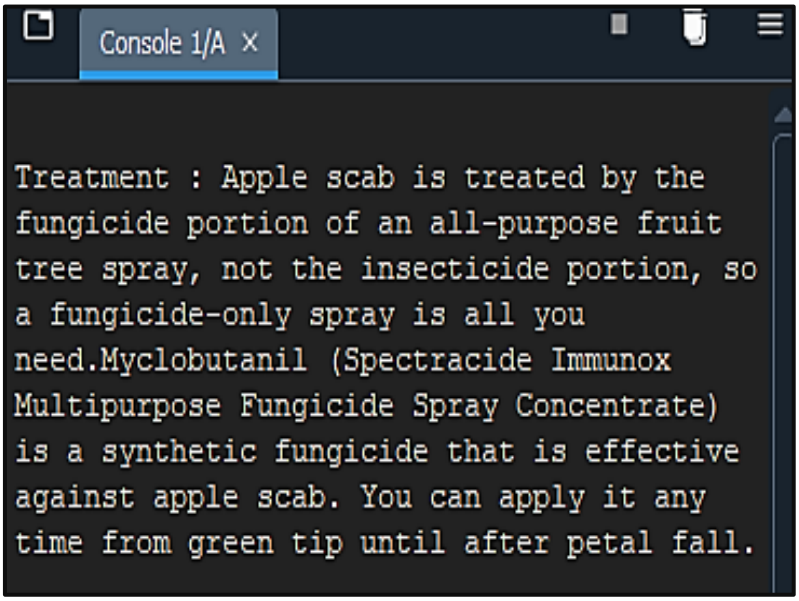

Figure. 14. Detection of Apple Scab disease and required treatment

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

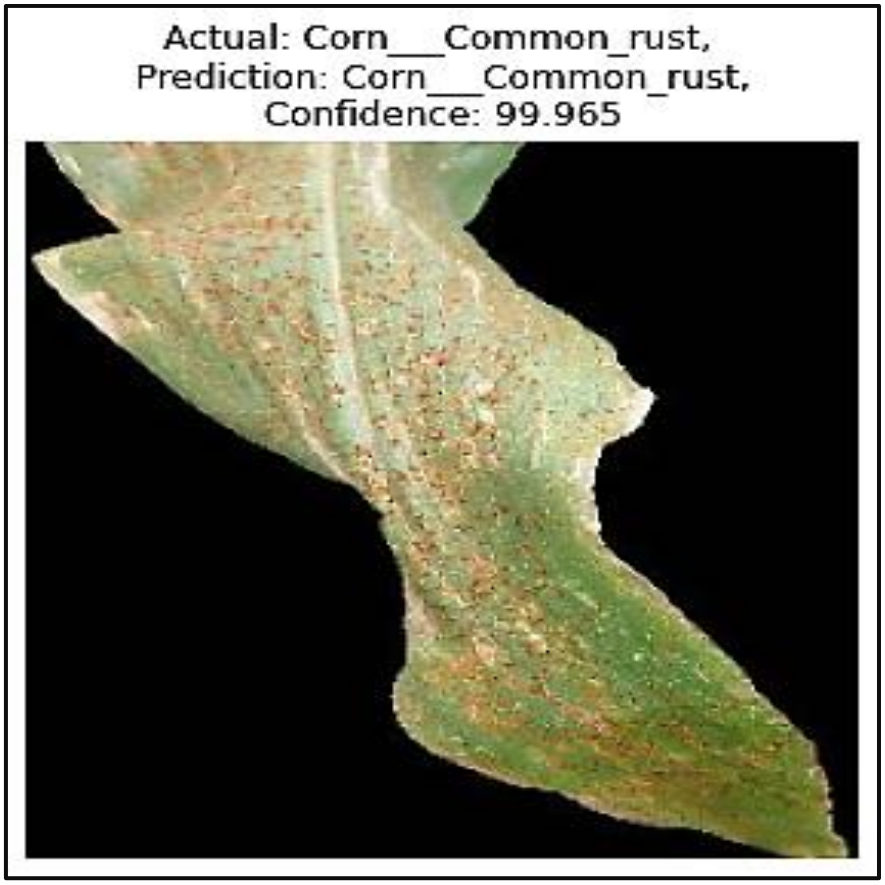

Figure. 15. Common rust in Corn

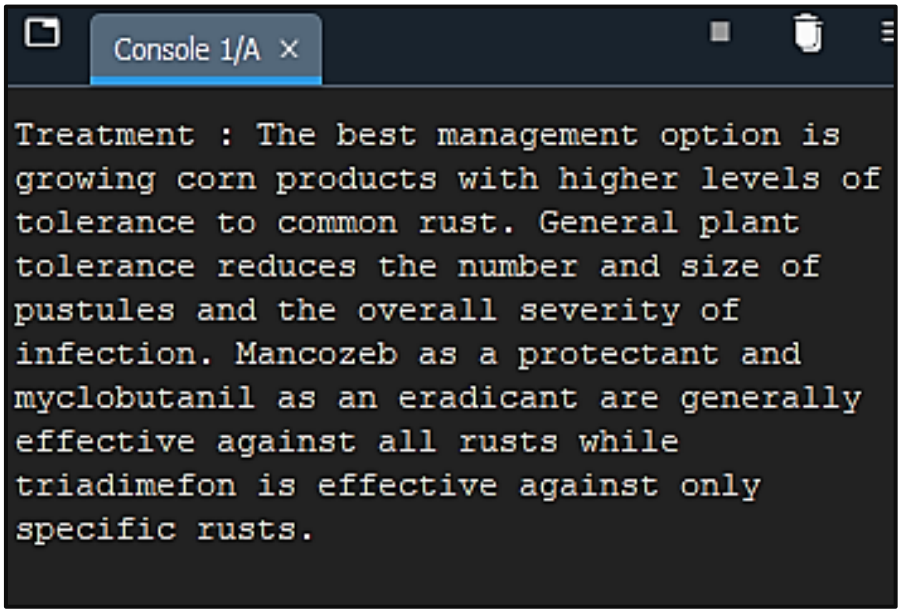

Figure. 16. Detection of Common rust in Corn and required treatment

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

Figure. 17. Tomato Yellow Leaf Curl Disease

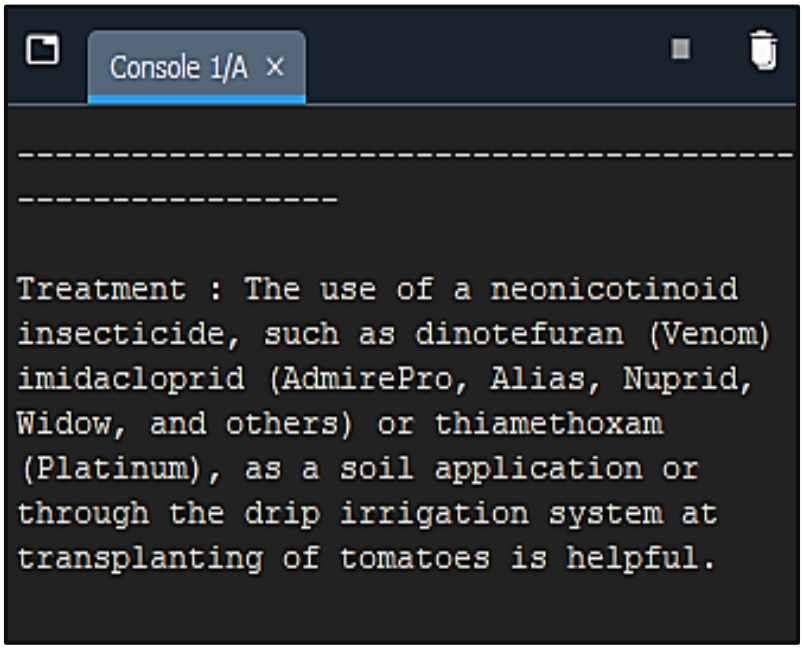

Figure. 18. Detection of Yellow Leaf curl Virus in Tomato disease and required treatment

VIII. HARDWARE IMPLEMENTATION

Model and Material which are used is presented in this section. Table and model should be in prescribed format. The authors have created a small hardware to demonstrate the spraying action of pesticide automatically. The model detects the type of disease from the input given by farmer. After detecting the type of

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

disease, it will automatically decide the which fertilizer should be given and it is stored in MongoDB cloud database as shown in Fig. (19).

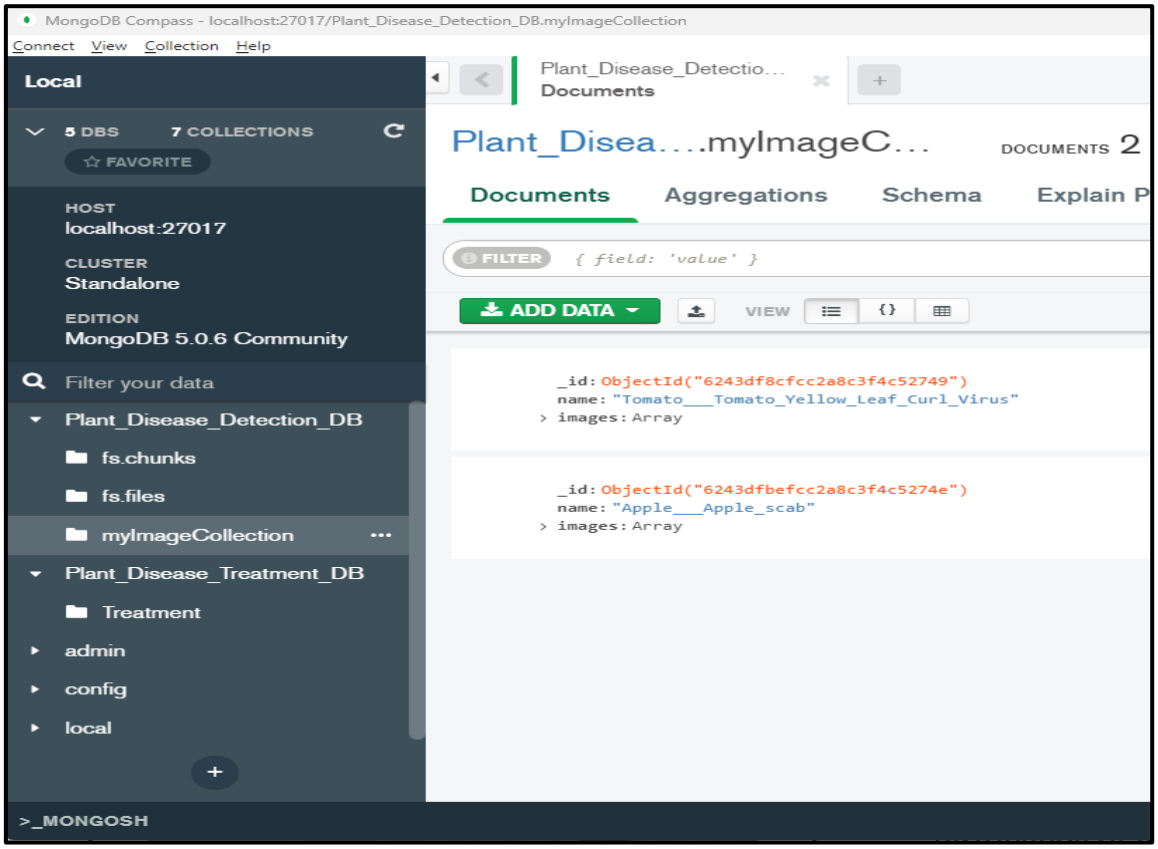

Figure. 19. MongoDB cloud database to store plant disease data

This fertilizer data is sent from cloud database to the Raspberry Pi3 (RPi) and it is sprinkled on the field. To show the sprinkling action of our model, the authors have attached a dc motor with RPi which starts working upon receiving the information from the cloud database as shown in Fig. The motor represents the distribution of fertilizer in the field [14].

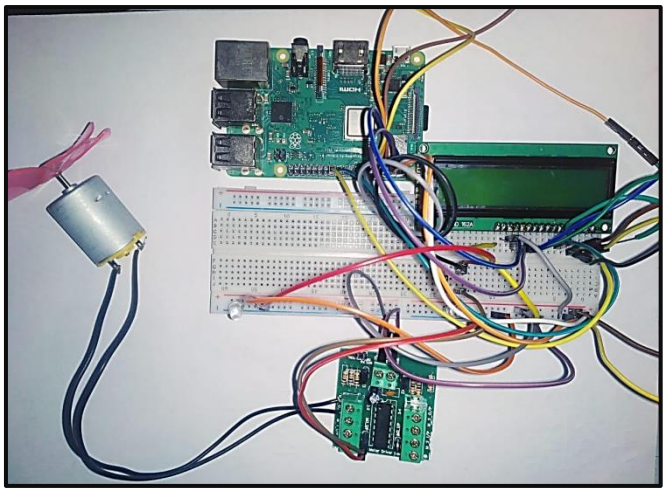

Figure. 20. Hardware model using Raspberry Pi3

Upon receiving the message about the type of the pesticide to be given the RPi will send the command to DC Servo Motor and it start spraying the pesticide. To mimic this action, we have attached a similar DC servo motor and have attached a fan along with it to display the spraying as shown in **Fig. (20)**. The information about the whether the pesticide is being sprayed or not is relayed using the LCD attached with the hardware as shown in **Fig. (21).**

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

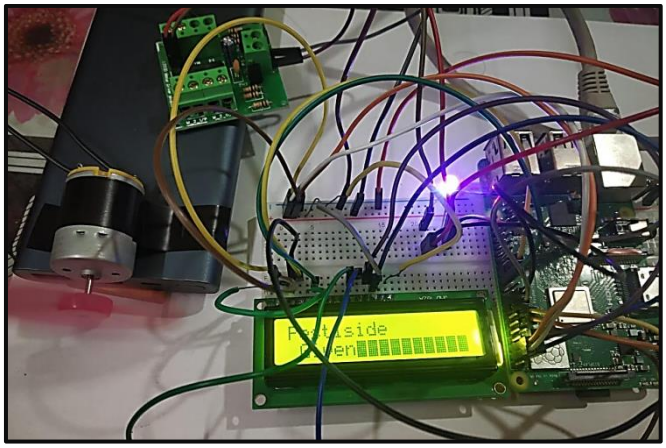

Figure. 21. Working of the hardware model

IX. CONCLUSION

The economy of a country or a state greatly depends on the agricultural output. Early detection of plant disease is of utmost importance in order to provide farmers and agricultural sector with steady flow of income. The news of suicide of farmers due to their inability to pay back loans can be heard all around the world. Moreover, the use of excessive fertilizers and pesticides degrades the soil making it unsuitable for agriculture. All these problems can be solved to great extent through the solution proposed by the authors. The solution is cost effective. It does not cost thousands of rupees like expensive pesticides. A finished product based upon the proposed idea can be setup within minutes with least amount of maintenance and can provide results with pinpoint accuracy.

X. REFERENCES

[1] S. V. Militante, B. D. Gerardo, and N. V. Dionisio, “Plant Leaf Detection and Disease Recognition using Deep Learning.” 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), 2019, doi: 10.1109/ecice47484.2019.8942686.

[2] A. J. Rau, J. Sankar, A. R. Mohan, D. Das Krishna, and J. Mathew, “IoT based smart irrigation system and nutrient detection with disease analysis.” 2017 IEEE Region 10 Symposium (TENSYMP), 2017, doi: 10.1109/tenconspring.2017.8070100.

[3] M. Sardogan, A. Tuncer, and Y. Ozen, “Plant Leaf Disease Detection and Classification Based on CNN with LVQ Algorithm.” 2018 3rd International Conference on Computer Science and Engineering (UBMK), 2018, doi: 10.1109/ubmk.2018.8566635.

[4] K. Prak Chang, “Analysis and feasibility study of plant disease using e-nose.” 2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014), 2014, doi: 10.1109/iccsce.2014.7072689.

[5] D. Arora, M. Garg and M. Gupta, "Diving deep in Deep Convolutional Neural Network," 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), 2020, pp. 749-751, doi: 10.1109/ICACCCN51052.2020.9362907.

[6] H. Durmuş, E. O. Güneş and M. Kırcı, "Disease detection on the leaves of the tomato plants by using deep learning," 2017 6th International Conference on Agro-Geoinformatics, 2017, pp. 1-5, doi: 10.1109/Agro-Geoinformatics.2017.8047016.

[7] J. Karnik and A. Suthar, “Agricultural Plant Leaf Disease Detection Using Deep Learning Techniques.” SSRN Electronic Journal, 2021, doi: 10.2139/ssrn.3917556.

[8] S. V. Militante, B. D. Gerardo, and R. P. Medina, “Sugarcane Disease Recognition using Deep Learning.” 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), 2019, doi: 10.1109/ecice47484.2019.8942690.

[9] C. Narvekar and M. Rao, "Flower classification using CNN and transfer learning in CNN- Agriculture Perspective," 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), 2020, pp. 660-664, doi: 10.1109/ICISS49785.2020.9316030.

[10] D. Kumar and V. Kukreja, "N-CNN Based Transfer Learning Method for Classification of Powdery Mildew Wheat Disease," 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), 2021, pp. 707-710, doi: 10.1109/ESCI50559.2021.9396972.

[11] D. Sinha and M. El-Sharkawy, "Thin MobileNet: An Enhanced MobileNet Architecture," 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), 2019, pp. 0280-0285, doi: 10.1109/UEMCON47517.2019.8993089.

[12] B. Dan, X. Sun and L. Liu, "Diseases and Pests Identification of Lycium Barbarum Using SE-MobileNet V2 Algorithm," 2019 12th International Symposium on Computational Intelligence and Design (ISCID), 2019, pp. 121-125, doi: 10.1109/ISCID.2019.00034.

[13] Howard, Andrew & Zhu, Menglong & Chen, Bo & Kalenichenko, Dmitry & Wang, Weijun & Weyand, Tobias & Andreetto, Marco & Adam, Hartwig. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. doi: https://doi.org/10.48550/arXiv.1704.04861.

[14] V. Sandeep, K. L. Gopal, S. Naveen, A. Amudhan and L. S. Kumar, "Globally accessible machine automation using Raspberry pi based on Internet of Things," 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2015, pp. 1144-1147, doi: 10.1109/ICACCI.2015.7275764.