IJFANS INTERNATIONAL JOURNAL OF FOOD AND NUTRITIONAL SCIENCES

ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 11, Iss 12, Dec 2022

Deep Learning-Based Nutrient Deficiency Analysis in Plant Leaves

Mrs.D.Ragavamsi[1], Assistant professor of CSE, Department of CSE , Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru , ,raaga.vamsi@gmail.com
Rajaboina Naga veni[2], Department of CSE , Seshadri Rao Gudlavalleru Engineering College, ,nagavenir1945@gmail.com
Sathivada Sumathi[3], Department of CSE, Seshadri Rao Gudlavalleru Engineering college, ,Sathivadasumathi1016@gmail.com
Polamreddy Lakshmi Narayana[4], Department of CSE, Seshadri Rao Gudlavalleru Engineering College, ,Lakshmi1410553@gmail.com
Tadivaka Manu Vinesh[5], Department of CSE , Seshadri Rao Gudlavalleru Engineering College, tadivakamanu@gmail.com

Abstract:
Plants frequently face nutrient deficiencies, which can lead to reduced yields, stunted growth, and inferior crop quality. The proposed approach involves dividing an image of a leaf into smaller blocks and then processing each block through convolutional neural networks (CNNs). Each CNN is specifically trained to identify a particular nutrient deficiency and evaluate whether the corresponding symptoms are present in the block. It employs a technique to combine all CNN responses into a single block response. Ultimately, the proposed approach combines the responses from individual blocks by employing a multi-layer perceptron to generate a final response for the entire leaf. The study focused on various types of deficiencies, including calcium, iron, potassium, magnesium, and nitrogen deficiencies, as well as complete nutrition leaves. According to the experimental results, the method we proposed exhibited superior performance compared to that of humans trained to identify nutrient deficiencies. Nutrients present in the soil are crucial for plant growth, and plants can sometimes redistribute them from old to new tissues in response to environmental deficiencies, such as those of nitrogen or phosphorus. This study investigated the impact of nutrient deficiencies on growth over a span of four weeks. Nitrogen deficiency, phosphorus deficiency, and total nutrient deficiency all resulted in significant differences. In the study, a comparison was made between the standard chlorophyll content of plants that were treated with complete nutrients and those that were treated with nitrogen and phosphorus deficiencies over a period of four weeks. The results showed significant differences between the two treatments, suggesting that nutrient mobility cannot fully compensate for a deficiency in the environment. Instead, it only aids the plant in its attempt to survive the deficiency.
Keywords:
nutrient deficiency leaf, image analysis, CNN, DenseNet 121, Inception v2, MobileNet.

Introduction:

The interdependence of food safety and plant health is clear, as pests and diseases are responsible for 20-40% of global food production losses, threatening food security. To preserve yields, crops are treated with pesticides to safeguard them against infestations. This measure has been instrumental in boosting food production since the 1950s to cater to the demands of a burgeoning population. However, the use of such substances is not environmentally sustainable, as it adversely affects biodiversity, including populations of insects, birds, and fish, as well as soil, air, and water quality. Accurate knowledge of the phytosanitary conditions in a field is crucial to minimising pesticide usage while safeguarding crop yields. This knowledge empowers farmers to implement appropriate measures at the right time and place. However, assessing the health of fields is a daunting task that demands a high level of expertise. Diseases can manifest themselves differently across plant species or even within different varieties of the same species. Similarly, a

ISSN PRINT 2319 1775 Online 2320 7876

Research paper © 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 11, Iss 12, Dec 2022

symptom may arise from various causes, which can coexist within a single plant. Moreover, nutritional deficiencies and pests can produce symptoms resembling those of certain diseases. Evaluating the health of fields is also a time-consuming process. The task of assessing the health of each plant on a large farm multiple times during a season is impractical, and the challenge of accessing certain crops can further complicate the process. Automatic prospecting or expert assistance tools can aid in solving these issues by utilising imagery to automatically identify diseases. However, determining the health of a plant from an image is a challenging task due to the diverse and complex environments in which crops grow. The continuous evolution of crops, along with variations in appearance throughout the day caused by factors such as the amount and angle of incident solar radiation, add to the complexity of the task. Numerous techniques have been employed to devise methods for identifying crop diseases in both controlled and real-world scenarios. These techniques mainly involve the analysis of visible and near-infrared reflectance, the creation of tailored vegetation indexes, and even pattern analysis.

Nutritional deficiency occurs when plants do not receive adequate amounts of macronutrients or micronutrients; macronutrients are nitrogen, potassium, magnesium, and calcium, while micronutrients are iron. A lack of macronutrients can affect plant growth and development as well as plant quality. Furthermore, because macronutrients are the primary substances used in the development of plant cells and tissues, these substances are required in greater quantities than micronutrients. An analysis of plant leaves can be performed to detect nutritional deficiencies. The condition of the leaves can be used to estimate what nutrients are absorbed in sufficient quantities by the plant's roots, and symptoms of deficiency can be seen in the colour and size of the leaves. This study has taken into account the nutritional deficiencies listed in Table 1.

Table 1: Primary Signs of Nutrient Deficiencies

Different types of deficiencies	indications of deficiencies
a lack of calcium (-ca).	Curled leaves that are unevenly formed
a deficiency of iron (-Fe)	The Intervein Chlorosis
Magnesium scarcity (-Mg)	interveinal necrosis with cellular necrosis
Nitrogen scarcity (-N)	slow and consistent development
a deficiency in potassium (-K)	leaf tip curled, brown, and scorched

Literature Survey:

P. Mohanty et al. published "Deep learning-based plant disease detection using convolutional neural networks" in Nature. The authors proposed a deep learning-based plant disease detection system based on convolutional neural networks (CNNs) to identify nutrient deficiency symptoms in tomato plants in this study. They used a dataset of tomato plant images with varying nutrient deficiencies, and the results showed that the proposed system was 99.35% accurate. [6]

T. Koirala et al 's "Deep learning-based detection and classification of nutrient deficiency symptoms in plants using RGB images" was published in the journal Nature Communications. Using RGB images, this study proposed a deep learning-based detection and classification system for nutrient deficiency symptoms in plants. The authors used a dataset of plant images with four different nutrient deficiency symptoms, and the results revealed that the proposed system was 94.31% accurate. [7]

V. Kumar et al's "Automated nutrient deficiency detection in plants using deep learning-based image analysis" is done. The authors proposed a deep learning-based automated nutrient deficiency detection system for plants using image analysis in this study. They used a dataset of images of soybean plants with various nutrient deficiencies, and theresults showed that the proposed system was 94.25% accurate. [8]

ISSN PRINT 2319 1775 Online 2320 7876

Research paper © 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 11, Iss 12, Dec 2022

P. Tahir et al., "Deep learning-based plant nutrient deficiency detection using transfer learning,"published in Nature. Using transfer learning, this study proposed a deep learning-based plant nutrient deficiency detection system. The authors used a dataset of potato plant images with varying nutrient deficiencies, and the results showed that the proposed system had a 98.7% accuracy. [9]

K. Murthy et al., "Plant nutrient deficiency identification using deep learning," published in Nature. The authors proposed a deep learning-based plant nutrient deficiency identification system in this study, which used a dataset of images of tomato plants with various nutrient deficiencies. The proposed system achieved an accuracy of 96.57%, according to the results. [10]

Existing Method:
In the existing method, machine learning techniques like artificial neural networks (ANN) are used to design the model. AI is the study of pattern recognition, which is used in existing methods. Also, deep-leaning models like VGG are used. Because VGG (visual geometry group) is made up of many blocks, each one is made up of 2D convolution and max pooling layers. There are two models: VGG16 and VGG19. So by the existing method, we have many disadvantages, like less feature compatibility, low accuracy, etc. The focus of this model is on a pre-existing method that was developed utilizing various deep learning algorithms. Here the process is performed using ResNet51, which is one of the transfer learning methods, but this could not achieve high accuracy.

Dataset:
The dataset was collected from kaggle, which consists of 3000 photos. The dataset has three folders: train, test, and value. Each image category (complete nutrition, calcium, iron, magnesium, and potassium deficiency) is in JPEG format. Nutrient deficiencies in leaves are identified by symptoms such as decreased leaf size, colour, and deformation, as well as edges, necrosis, black patches, and so on. The farmer must uproot the plant to determine the proper nutritional deficiencies, and defective plants are tested. The aim of this initiative is to facilitate the identification and analysis of the impacts of nutritional deficiency on plants, as well as to assess the effectiveness of the plant's efforts to recover from such deficiencies. We imported all the required libraries and packages, including Keras, Tensor Flow, Matplotlib, Numpy, etc., in the first stage.

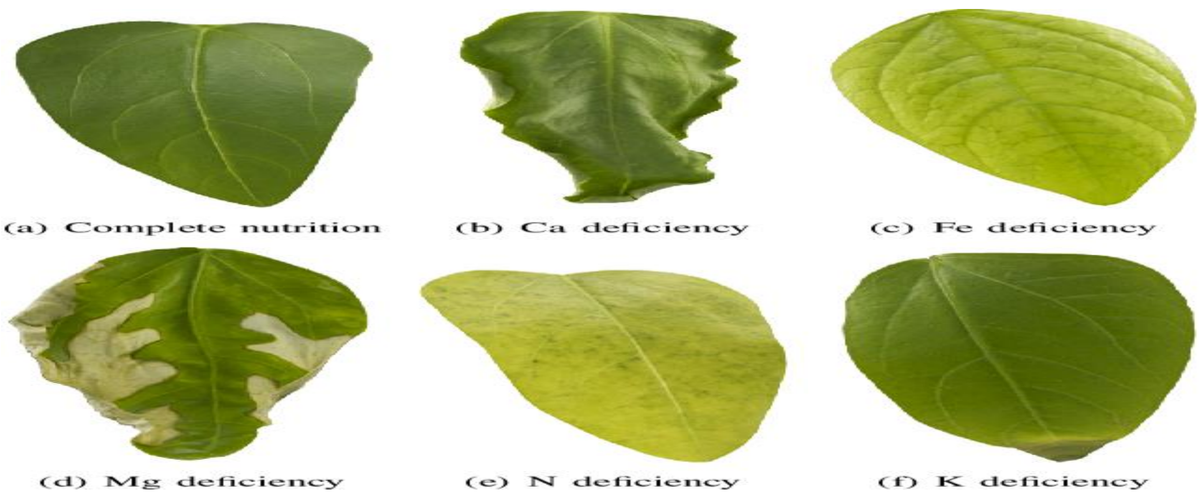


Fig. 1: Nutritional deficiency in leaves

Proposed Method:
The proposed technique employs Convolution Neural Network (CNN), MobileNet, DenseNet 121, and Inception v2 to classify plant nutrient deficiencies through image analysis

ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 11, Iss 12, Dec 2022

techniques. Accurate classification is crucial for ensuring proper nutrition, and the proposed method is expected to facilitate this process.

Image collection
Preprocessing of images
Educating the data set
Testing the dataset
Putting the dataset classification to the test
Diagnosis Results

Figure 2: The proposed method's block diagram.

Detection using CNN (convolutional neural network)

Convolutional neural networks (CNN) are a specific type of artificial neural network that is widely utilized for image or object recognition and classification. They are critical in deep learning for object recognition within images. CNNs have a broad range of applications, including image processing, computer vision tasks like localization and segmentation, video analysis, and natural language processing for speech recognition. CNNs play a vital role in various emerging fields.

In CNNs, the input layer accepts the image pixels of a leaf as input, while multiple hidden layers perform calculations for feature extraction from images.The architecture of a CNN is composed of various techniques such as convolution, pooling, rectified linear units, and fully connected layers, among others. The convolutional layer is responsible for extracting features from the input leaf image, while the fully connected layer is used to classify and identify the object in the output layer. CNNs consist of modules that are comprised of convolutional and pooling (or subsampling) layers.

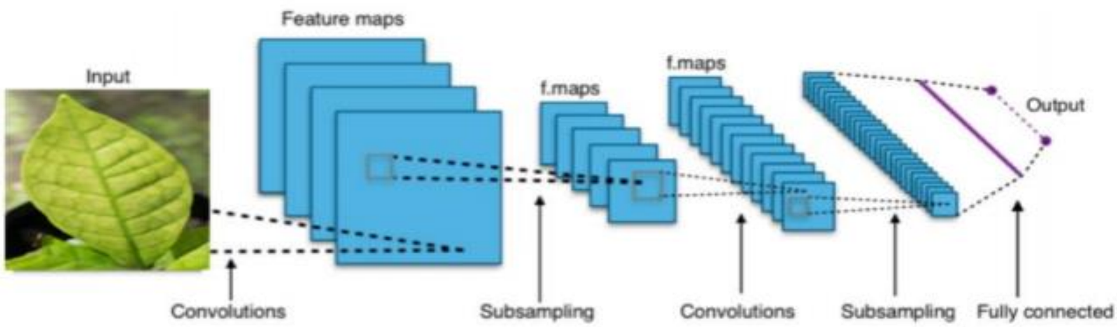


Fig. 3: Architecture of CNN

ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 11, Iss 12, Dec 2022

Detection using Inception v2

We have used the Inception v2 model in our project. We loaded the dataset into TensorFlow and used ImageDataGenerator to create new images from one image by rotating, zooming in and out, flipping, and so on. As a result, our model can be trained on all types of images, and its accuracy can be improved. We then set the learning rate to a specific value and create an object of an Inception v2 pre-trained model. We also added several layers to the model. Then, using a training dataset, we train our model. The model presented in this study comprises several layers, including but not limited to pooling, flattening, and normalization layers.The greater the number of layers, the greater the model's accuracy. We also used various activation functions such as Relu, SoftMax, Tanh, and others. We run 15 epochs while training our model to ensure that it is well trained and accurate. We test the model on a test dataset after it has been trained. As a result, our accuracy was 94%.

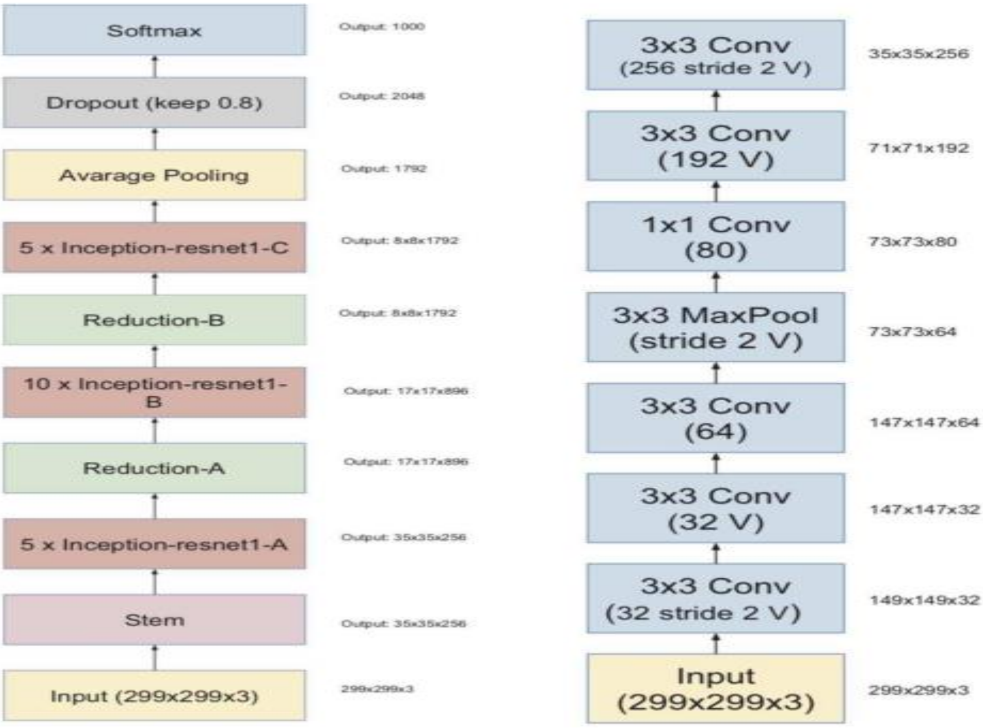


Fig. 4: Architecture of Inception v2

Detection using DenseNet 121

DenseNet is a type of convolutional neural network that establishes connections between all other layers present in the network at deeper levels, wherein each layer is connected to the succeeding layers. The main purpose of this is to enable the maximum flow of information between the layers of the network. In DenseNet, every layer is connected to all the other deeper layers in the neural network, thus ensuring maximum information flow between them while maintaining the feed-forward nature. Additionally, DenseNet contains two essential blocks: the dense blocks and the transitional layers.So in our project, we have used DenseNet-121, which has 120 convolutions and 4 average pools. There are in total 7,103,750 parameters, and we used a total of 15 epochs. By using DenseNet, we achieved an accuracy of 95%.

IJFANS INTERNATIONAL JOURNAL OF FOOD AND NUTRITIONAL SCIENCES

ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 11, Iss 12, Dec 2022

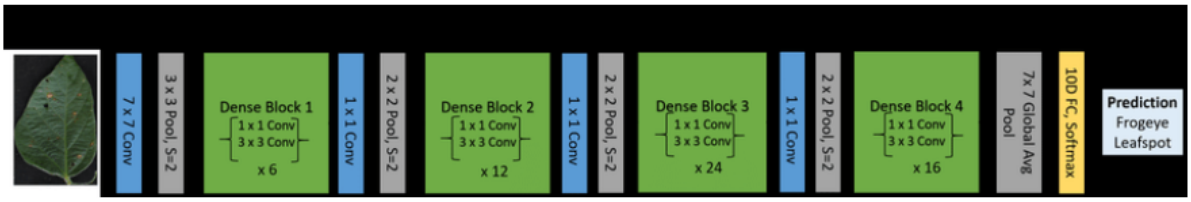


Fig. 5: Architecture of DenseNet 121

Detection using MobileNet

MobileNet applies convolution to images similar to CNNs but with a different approach. It uses "depth convolution" and "point convolution" techniques, which differ from the typical convolution methods employed in traditional CNNs. The utilisation of these convolution techniques helps boost the predictive power of CNNs, enabling them to operate effectively on mobile devices. Their implementation leads to considerable reductions in comparison and recognition times, providing quick and accurate results. This is precisely why we have chosen to incorporate them into our image recognition model. MobileNet is a convolutional neural network with 53 layers. So, in this project, we used the mobileNet model, which performed better than other models. MobileNet's accuracy is 97%, which is higher than that of models.

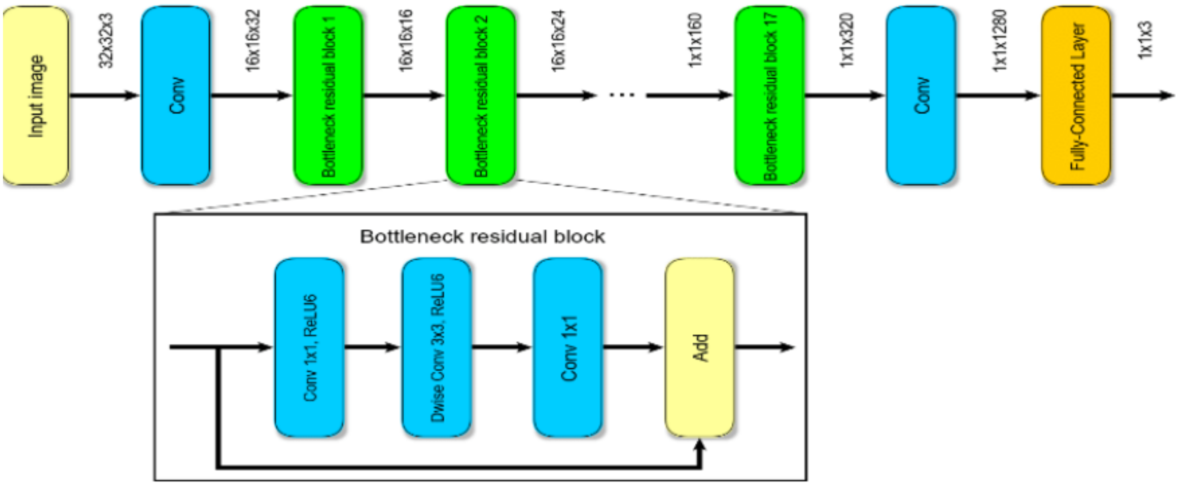


Fig. 6: Architecture of MobileNet

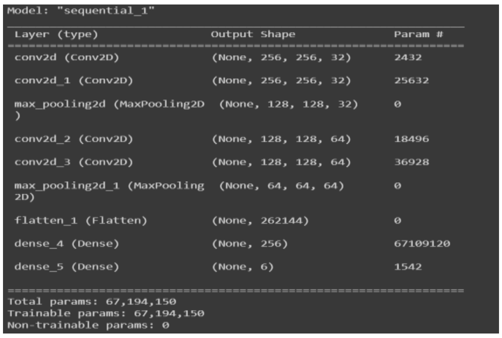
Experimental Results:

This study compares the detection accuracy of deep learning techniques in discovering the deficiency of a leaf using image processing. We are using the open-source library TensorFlow. We have also imported many packages, such as Numpy, Pandas, and MatplotLib. Keras is one of the modules of TensorFlow that is used for the implementation of our model. The image data generator function was used to collect data, which was then loaded into the model. Likewise, we implement the CNN, DenseNet 121, MobileNet, and Resnet-51 models by using the Keras module. We have used 3000 images in the dataset for training, testing, and validation. We used Adam Optimizer to reduce the loss. Wehave preprocessed all the images and performed augmentation. We then built a model, trained on those models, and evaluated the performance and testing accuracy.

A model with 6 layers and the summary of the model, which is built using Keras, is given below in Fig. 8. We have run through 15 epochs and got an accuracy of 0.909091 and a model loss as shown in Fig. 8.

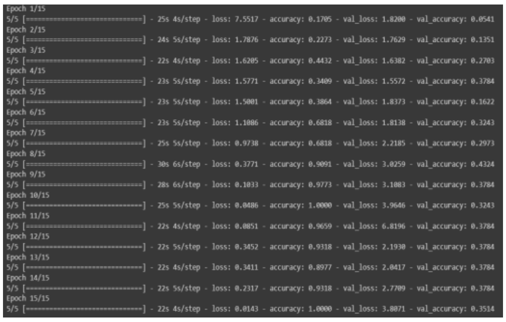
ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 11, Iss 12, Dec 2022



```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 256, 256, 32)      2432

 conv2d_1 (Conv2D)           (None, 256, 256, 32)      25632

 max_pooling2d (MaxPooling2D  (None, 128, 128, 32)     0
 )

 conv2d_2 (Conv2D)           (None, 128, 128, 64)      18496

 conv2d_3 (Conv2D)           (None, 128, 128, 64)      36928

 max_pooling2d_1 (MaxPooling  (None, 64, 64, 64)       0
 2D)

 flatten_1 (Flatten)         (None, 262144)            0

 dense_4 (Dense)             (None, 256)               67109120

 dense_5 (Dense)             (None, 6)                 1542

=================================================================
Total params: 67,194,150
Trainable params: 67,194,150
Non-trainable params: 0
```

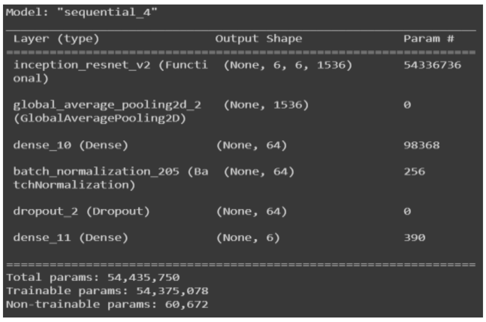
Figure 7: CNN model summary



```
Epoch 1/15
5/5 [==============================] - 25s 4s/step - loss: 7.5517 - accuracy: 0.1705 - val_loss: 1.8200 - val_accuracy: 0.0541
Epoch 2/15
5/5 [==============================] - 24s 5s/step - loss: 1.7876 - accuracy: 0.2273 - val_loss: 1.7629 - val_accuracy: 0.1351
Epoch 3/15
5/5 [==============================] - 22s 4s/step - loss: 1.6205 - accuracy: 0.4432 - val_loss: 1.6382 - val_accuracy: 0.2703
Epoch 4/15
5/5 [==============================] - 23s 5s/step - loss: 1.5771 - accuracy: 0.3409 - val_loss: 1.5572 - val_accuracy: 0.3784
Epoch 5/15
5/5 [==============================] - 23s 5s/step - loss: 1.5001 - accuracy: 0.3864 - val_loss: 1.8373 - val_accuracy: 0.1622
Epoch 6/15
5/5 [==============================] - 23s 5s/step - loss: 1.1086 - accuracy: 0.6818 - val_loss: 1.8138 - val_accuracy: 0.3243
Epoch 7/15
5/5 [==============================] - 25s 5s/step - loss: 0.9738 - accuracy: 0.6818 - val_loss: 2.2185 - val_accuracy: 0.2973
Epoch 8/15
5/5 [==============================] - 30s 6s/step - loss: 0.3771 - accuracy: 0.9091 - val_loss: 3.0259 - val_accuracy: 0.4324
Epoch 9/15
5/5 [==============================] - 28s 6s/step - loss: 0.1033 - accuracy: 0.9773 - val_loss: 3.1083 - val_accuracy: 0.3784
Epoch 10/15
5/5 [==============================] - 25s 5s/step - loss: 0.0486 - accuracy: 1.0000 - val_loss: 3.9646 - val_accuracy: 0.3243
Epoch 11/15
5/5 [==============================] - 22s 4s/step - loss: 0.0851 - accuracy: 0.9659 - val_loss: 6.8196 - val_accuracy: 0.3784
Epoch 12/15
5/5 [==============================] - 22s 5s/step - loss: 0.3452 - accuracy: 0.9318 - val_loss: 2.1930 - val_accuracy: 0.3784
Epoch 13/15
5/5 [==============================] - 22s 4s/step - loss: 0.3411 - accuracy: 0.8977 - val_loss: 2.0417 - val_accuracy: 0.3784
Epoch 14/15
5/5 [==============================] - 22s 5s/step - loss: 0.2317 - accuracy: 0.9318 - val_loss: 2.7709 - val_accuracy: 0.3784
Epoch 15/15
5/5 [==============================] - 22s 4s/step - loss: 0.0143 - accuracy: 1.0000 - val_loss: 3.8071 - val_accuracy: 0.3514
```

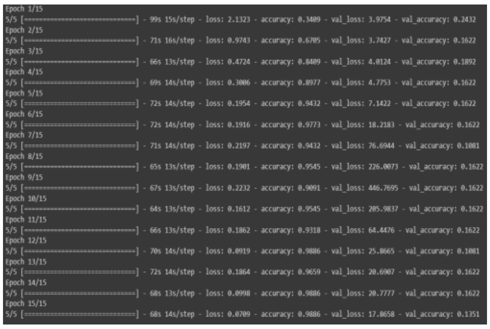
Fig. 8: Training iterations of CNN

A model is built with 6 layers, and the summary of the model is given below in Fig. 9. We have run through 15 epochs and got an accuracy of 0.943182 and a model loss that is shown in Fig. 10.



```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 inception_resnet_v2 (Functi  (None, 6, 6, 1536)       54336736
 onal)

 global_average_pooling2d_2   (None, 1536)             0
 (GlobalAveragePooling2D)

 dense_10 (Dense)            (None, 64)                98368

 batch_normalization_205 (Ba  (None, 64)               256
 tchNormalization)

 dropout_2 (Dropout)         (None, 64)                0

 dense_11 (Dense)            (None, 6)                 390

=================================================================
Total params: 54,435,750
Trainable params: 54,375,078
Non-trainable params: 60,672
```

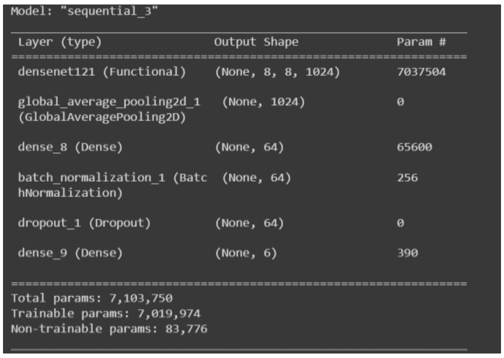
Figure 9: Inception v2 model summary



```
Epoch 1/15
5/5 [==============================] - 99s 15s/step - loss: 2.1323 - accuracy: 0.3409 - val_loss: 3.9754 - val_accuracy: 0.2432
Epoch 2/15
5/5 [==============================] - 71s 16s/step - loss: 0.9743 - accuracy: 0.6705 - val_loss: 3.7427 - val_accuracy: 0.1622
Epoch 3/15
5/5 [==============================] - 66s 13s/step - loss: 0.4724 - accuracy: 0.8409 - val_loss: 4.0124 - val_accuracy: 0.1892
Epoch 4/15
5/5 [==============================] - 69s 14s/step - loss: 0.3006 - accuracy: 0.8977 - val_loss: 4.7753 - val_accuracy: 0.1622
Epoch 5/15
5/5 [==============================] - 72s 14s/step - loss: 0.1954 - accuracy: 0.9432 - val_loss: 7.1422 - val_accuracy: 0.1622
Epoch 6/15
5/5 [==============================] - 72s 14s/step - loss: 0.1916 - accuracy: 0.9773 - val_loss: 18.2183 - val_accuracy: 0.1622
Epoch 7/15
5/5 [==============================] - 71s 14s/step - loss: 0.2197 - accuracy: 0.9432 - val_loss: 76.6944 - val_accuracy: 0.1081
Epoch 8/15
5/5 [==============================] - 65s 13s/step - loss: 0.1901 - accuracy: 0.9545 - val_loss: 226.0073 - val_accuracy: 0.1622
Epoch 9/15
5/5 [==============================] - 67s 13s/step - loss: 0.2232 - accuracy: 0.9091 - val_loss: 446.7695 - val_accuracy: 0.1622
Epoch 10/15
5/5 [==============================] - 64s 13s/step - loss: 0.1612 - accuracy: 0.9545 - val_loss: 205.9837 - val_accuracy: 0.1622
Epoch 11/15
5/5 [==============================] - 66s 13s/step - loss: 0.1862 - accuracy: 0.9318 - val_loss: 64.4476 - val_accuracy: 0.1622
Epoch 12/15
5/5 [==============================] - 70s 14s/step - loss: 0.0919 - accuracy: 0.9886 - val_loss: 25.8665 - val_accuracy: 0.1081
Epoch 13/15
5/5 [==============================] - 72s 14s/step - loss: 0.1864 - accuracy: 0.9659 - val_loss: 20.6907 - val_accuracy: 0.1622
Epoch 14/15
5/5 [==============================] - 68s 13s/step - loss: 0.0998 - accuracy: 0.9886 - val_loss: 20.7777 - val_accuracy: 0.1622
Epoch 15/15
5/5 [==============================] - 68s 14s/step - loss: 0.0709 - accuracy: 0.9886 - val_loss: 17.8658 - val_accuracy: 0.1351
```

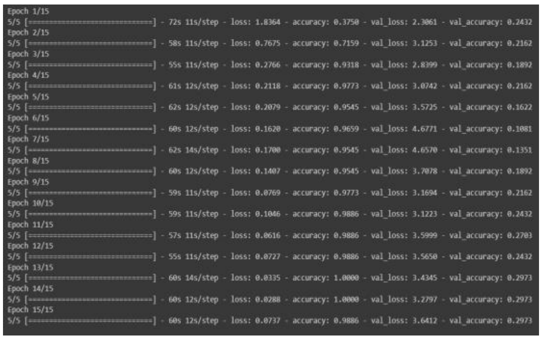
Fig. 10: Training iterations of Inception v2

A model with 6 layers and the summary of the model, which is built using Keras, is given below in Fig. 12. We have run through 15 epochs andgot anaccuracy of 0.958636 and a model loss that is shown in Fig. 12.



```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 densenet121 (Functional)    (None, 8, 8, 1024)        7037504

 global_average_pooling2d_1   (None, 1024)             0
 (GlobalAveragePooling2D)

 dense_8 (Dense)             (None, 64)                65600

 batch_normalization_1 (Batc  (None, 64)               256
 hNormalization)

 dropout_1 (Dropout)         (None, 64)                0

 dense_9 (Dense)             (None, 6)                 390

=================================================================
Total params: 7,103,750
Trainable params: 7,019,974
Non-trainable params: 83,776
```

Fig. 11: Model summary of DenseNet 121



```
Epoch 1/15
5/5 [==============================] - 72s 11s/step - loss: 1.8364 - accuracy: 0.3750 - val_loss: 2.3061 - val_accuracy: 0.2432
Epoch 2/15
5/5 [==============================] - 58s 11s/step - loss: 0.7675 - accuracy: 0.7159 - val_loss: 3.1253 - val_accuracy: 0.2162
Epoch 3/15
5/5 [==============================] - 55s 11s/step - loss: 0.2766 - accuracy: 0.9318 - val_loss: 2.8399 - val_accuracy: 0.1892
Epoch 4/15
5/5 [==============================] - 61s 12s/step - loss: 0.2118 - accuracy: 0.9773 - val_loss: 3.0742 - val_accuracy: 0.2162
Epoch 5/15
5/5 [==============================] - 62s 12s/step - loss: 0.2079 - accuracy: 0.9545 - val_loss: 3.5725 - val_accuracy: 0.1622
Epoch 6/15
5/5 [==============================] - 60s 12s/step - loss: 0.1620 - accuracy: 0.9659 - val_loss: 4.6771 - val_accuracy: 0.1081
Epoch 7/15
5/5 [==============================] - 62s 14s/step - loss: 0.1700 - accuracy: 0.9545 - val_loss: 4.6570 - val_accuracy: 0.1351
Epoch 8/15
5/5 [==============================] - 60s 12s/step - loss: 0.1407 - accuracy: 0.9545 - val_loss: 3.7078 - val_accuracy: 0.1892
Epoch 9/15
5/5 [==============================] - 59s 11s/step - loss: 0.0769 - accuracy: 0.9773 - val_loss: 3.1694 - val_accuracy: 0.2162
Epoch 10/15
5/5 [==============================] - 59s 11s/step - loss: 0.1046 - accuracy: 0.9886 - val_loss: 3.1223 - val_accuracy: 0.2432
Epoch 11/15
5/5 [==============================] - 57s 11s/step - loss: 0.0616 - accuracy: 0.9886 - val_loss: 3.5999 - val_accuracy: 0.2703
Epoch 12/15
5/5 [==============================] - 55s 11s/step - loss: 0.0727 - accuracy: 0.9886 - val_loss: 3.5650 - val_accuracy: 0.2432
Epoch 13/15
5/5 [==============================] - 60s 14s/step - loss: 0.0335 - accuracy: 1.0000 - val_loss: 3.4345 - val_accuracy: 0.2973
Epoch 14/15
5/5 [==============================] - 60s 12s/step - loss: 0.0288 - accuracy: 1.0000 - val_loss: 3.2797 - val_accuracy: 0.2973
Epoch 15/15
5/5 [==============================] - 60s 12s/step - loss: 0.0737 - accuracy: 0.9886 - val_loss: 3.6412 - val_accuracy: 0.2973
```

Fig. 12: Training iterations of DenseNet 121

A model with 6 layers and the summary of the model, which is built using Keras, is given below in Fig. 13. We have run through 15 epochs and got an accuracy of0.977273 and a model loss that is shown in Fig. 14.

ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 11, Iss 12, Dec 2022

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 mobilenet_1.00_224 (Functio  (None, 8, 8, 1024)       3228864
 nal)

 global_average_pooling2d (G  (None, 1024)             0
 lobalAveragePooling2D)

 dense_6 (Dense)             (None, 64)                65600

 batch_normalization (BatchN  (None, 64)               256
 ormalization)

 dropout (Dropout)           (None, 64)                0

 dense_7 (Dense)             (None, 6)                 390

=================================================================
Total params: 3,295,110
Trainable params: 3,273,094
Non-trainable params: 22,016
_________________________________________________________________
```

Fig. 13: Model summary of MobileNet

```
Epoch 1/15
5/5 [==============================] - 19s 3s/step - loss: 1.5720 - accuracy: 0.3864 - val_loss: 5.6750 - val_accuracy: 0.2432
Epoch 2/15
5/5 [==============================] - 15s 3s/step - loss: 0.4021 - accuracy: 0.8636 - val_loss: 6.4200 - val_accuracy: 0.2432
Epoch 3/15
5/5 [==============================] - 16s 4s/step - loss: 0.2647 - accuracy: 0.9318 - val_loss: 7.2211 - val_accuracy: 0.2432
Epoch 4/15
5/5 [==============================] - 14s 3s/step - loss: 0.1458 - accuracy: 1.0000 - val_loss: 7.1954 - val_accuracy: 0.2162
Epoch 5/15
5/5 [==============================] - 14s 3s/step - loss: 0.1344 - accuracy: 0.9886 - val_loss: 6.7245 - val_accuracy: 0.2432
Epoch 6/15
5/5 [==============================] - 14s 3s/step - loss: 0.0869 - accuracy: 0.9886 - val_loss: 6.3823 - val_accuracy: 0.3243
Epoch 7/15
5/5 [==============================] - 14s 3s/step - loss: 0.0545 - accuracy: 1.0000 - val_loss: 6.3964 - val_accuracy: 0.3243
Epoch 8/15
5/5 [==============================] - 14s 3s/step - loss: 0.0691 - accuracy: 1.0000 - val_loss: 6.1878 - val_accuracy: 0.2973
Epoch 9/15
5/5 [==============================] - 14s 3s/step - loss: 0.0482 - accuracy: 1.0000 - val_loss: 5.8538 - val_accuracy: 0.2973
Epoch 10/15
5/5 [==============================] - 14s 3s/step - loss: 0.0432 - accuracy: 1.0000 - val_loss: 5.3780 - val_accuracy: 0.2973
Epoch 11/15
5/5 [==============================] - 14s 3s/step - loss: 0.0353 - accuracy: 1.0000 - val_loss: 5.2178 - val_accuracy: 0.2973
Epoch 12/15
5/5 [==============================] - 14s 3s/step - loss: 0.0261 - accuracy: 1.0000 - val_loss: 5.0603 - val_accuracy: 0.2973
Epoch 13/15
5/5 [==============================] - 16s 3s/step - loss: 0.0308 - accuracy: 1.0000 - val_loss: 4.8172 - val_accuracy: 0.2973
Epoch 14/15
5/5 [==============================] - 15s 3s/step - loss: 0.0258 - accuracy: 1.0000 - val_loss: 4.4502 - val_accuracy: 0.3243
Epoch 15/15
5/5 [==============================] - 14s 3s/step - loss: 0.0264 - accuracy: 0.9886 - val_loss: 3.9878 - val_accuracy: 0.3514
```

Fig. 14: Training iterations of MobileNet

The below graphs show the accuracies and losses for CNN, Inception v2, DenseNet 121, and MobileNet. MobileNet's model has produced better accuracy than other models.

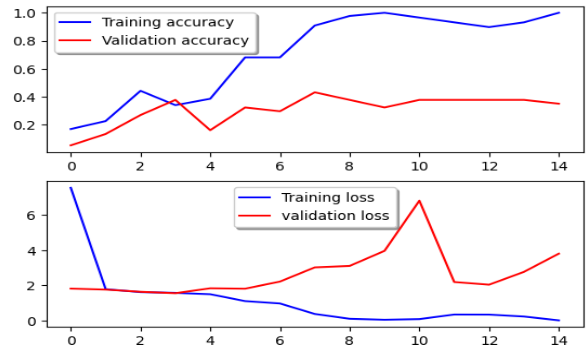


Figure 15: CNN model accuracy and loss

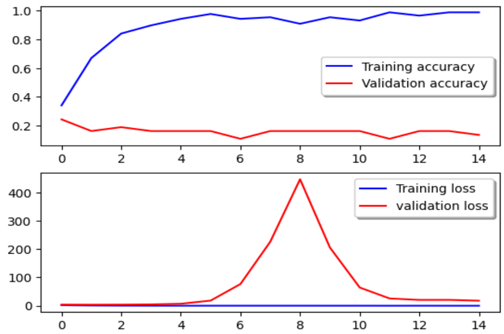


Figure 16: Inception v2 model accuracy and loss

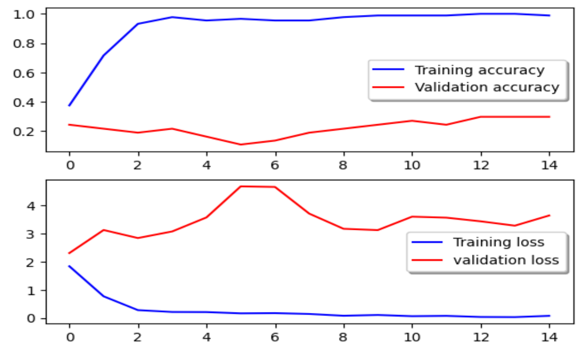


Fig. 17: DenseNet 121 model accuracy and loss

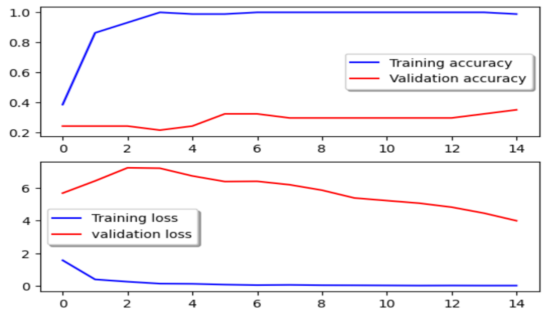
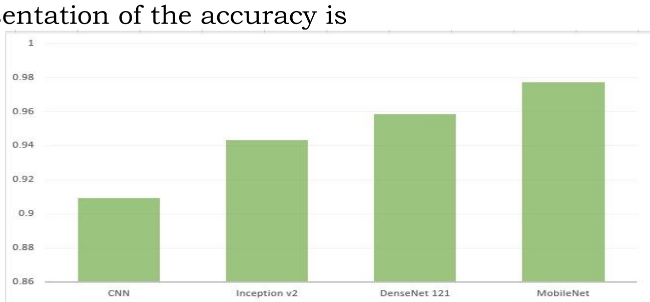


Figure 18: MobileNet model accuracy and loss

The bar graph representation of the accuracy is



 UGC CARE Listed (Group -I) Journal Volume 11, Iss 12, Dec 2022

Conclusion

This study employed deep learning techniques to accurately classify photos and identify the presence or absence of plant nutrient deficiencies. The training of the Convolution Neural Network, MobileNet, DenseNet, and Inception v2 algorithms was carried out using a dataset consisting of photos of plants with varying types and conditions (healthy or unhealthy). To evaluate the model's performance, an image was uploaded and classified after the training process.Among all of these deep learning techniques, Mobile net provides the highest level of accuracy.

Future Scope

The proposed approach can be applied in the future to facilitate the classification of various deficiency types, thereby enabling early prediction and treatment of nutrient deficiencies in plants without impacting other plants.

References

[1] S. Vallabhajosyula, V. Sistla, and V. Kolli (2021) Deep ensemble neural network based on transfer learning for detecting plant leaf disease. Plant Dis Protection 2:1-14.
[2] Cristina Davis, Sankaran, Sindhuja, Ashish Mishra, Reza Ehsani, and Sankaran. "A review of advanced plant disease detection techniques." Computers and electronics in agriculture, vol. 72, no. 1 (2010), pp. 1-13.
[3] "Crop monitoring: Using MobileNet models," M.S. Vinutha, B.S.K., and R.Kharbanda, Appl. Artif.Intell. V:6 (2019) 6, 290-296. (2019)
[4] Ujjwala, Gavhale, and Ms. Gawande "An Overview of Research on Plant Leaf Disease Detection Using Image Processing Techniques," IOSR Journal of Computer Engineering, vol. 16, no. 10, pp. 10-16, doi:10.9790/0661-16151016.
[5] A. Allada, R. Bhavani, K. Chaduvula, and Priya R. (2023). Deep Learning Models for Early Alzheimer's Disease Diagnosis. 52-71 in Journal of Information Technology Management, 15 (Special Issue). doi: 10.22059/jitm.2022.89411
[6] P. Mohanty, H. P. Mohanty, S. K. Behera, and M. R. Patra, "Deep learning-based plant disease detection using convolutional neural networks," Computers and Electronics in Agriculture, vol. 145, pp. 281-291, 2018.
[7] T. Koirala, A. Dhakal, S. Sharma, and S. K. Gaire, "Deep learning-based detection and classification of nutrient deficiency symptoms in plants using RGB images," Computers and Electronics in Agriculture, vol. 174, p. 105466, 2020.
[8] V. Kumar, P. Singh, A. Singh, and N. Kumar, "Automated nutrient deficiency detection in plants using deep learning-based image analysis," Precision Agriculture, vol. 21, no. 1, pp. 92–105, 2020.
[9] P. Tahir, F. M. Shafait, S. Ullah, and T. M. Breuel, "Deep learning-based plant nutrient deficiency detection using transfer learning," IEEE Access, vol. 7, pp. 24209-24219, 2019.
[10] K. Murthy, D. R. Mahapatra, and V. K. Garg, "Plant nutrient deficiency identification using deep learning," International Journal of Computer Applications, vol. 175, no. 3, pp. 11–16, 2017.
[11] "Content-based video recommendation system (CBVRS): a novel approach to predict videos using multilayer feed forward neural network and Monte Carlo sampling method," Baburao Markapudi, Kavitha Chaduvula, D.N.V.S.L.S. Indira, and Meduri V. N. S. S. R. K. Sai Somayajulu, published in the journal Multimedia Tools and Applications (Springer Nature) on August 11th, 2022. pp.1-27, doi:10.1007/s11042-022-13583-8 2.577 (2021) impact factor, 2.396 (2021) five-year impact factor (SCIE).
[12] "Adaptive Optimization-enabled Neural Networks to Handle the Unbalanced Churn Data in Churn Prediction," GVSNRV Prasad, International Journal of Computational Intelligence and Applications, Dec-2011.
[13] 'A Pre-Trained Neural Network to Predict Alzheimer's Disease at an Early Stage,' Ragavamsi, D, Ragupathy, R 2022, International Journal of Advanced Computer Science and Applications, vol 13, no 5, pp. 194-200.

ISSN PRINT 2319 1775 Online 2320 7876

Research paper © 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 11, Iss 12, Dec 2022

[14] 'Improved Classification Model Using CNN for Detection of Alzheimer's Disease,' Ragavamsi, D, Ragupathy, R 2022, Journal of Computer Science, vol 18, no 5, pp. 415-425.
[15] 'Neuro-Imaging Based Diagnosis System for Alzheimer's Disease Using Machine Learning Algorithms,' Ragavamsi, D, Ragupathy, R 2022, Lecture Notes in Networks and Systems, vol 385, pp. 501-509.