ISSN: 2633-4828 Vol. 6 No.1, January, 2024

International Journal of Applied Engineering & Technology

EFFECTIVE DEEP LEARNING MODEL TO DIAGNOSE PLANT DISEASE AND IDENTIFY THE LOCATION OF THE SPREAD

Ms. Vidhya R[1] and Ms. G Sandhya[2]

[1]Associate Professor and [2]Assistant Professor,T Kapil, Jayadev J B, Jude Raehan R, Kousalya J

Computer Science and Engineering, Sri Krishna College of Technology, Coimbatore, India

ABSTRACT

Food security is very important nowadays with the rising impact of global warming, climate change, and overpopulation. In such a critical situation food shortage can have a huge impact on all of us. One of the challenges to this food security is plant diseases. Early detection of these spreadable plant diseases can limit the damage of their impact. Normal plant disease and evaluation is generally slow and not very efficient in containing the spread. The data collected are not centralized to have a clear view of the disease spread. This paper aims to solve those problems by proposing a highly accurate and efficient deep learning model stored in the cloud along with a centralized database to store the disease spread along with their location and mapping of the spread of the disease in a map. The research explores the use of the convolutional neural network (CNN) for the classification and identification of diseases using an open dataset of both diseased and healthy plant leaf images. In this paper, we have compared our deep learning model with another already existing pre-trained model for accurate plant disease recognition. Their performances were examined and it was found that our model built on top of the EfficientNet outperformed the other existing models with an astonishing 99.82% accuracy. Once this plant disease identification has been done, we can use the geotagging to identify the location for identifying the spread of the disease. This paper highlights the huge potential of using the EfficientNet model with transfer learning in plant disease classification to offer an effective model with high accuracy with reliability and also provide a view of the spread of the disease.

Keywords: Convolutional Neural Network, Image processing, Plant Disease Classification, Deep Learning, Visual symptoms, Geo-tagging, Transfer Learning

1. INTRODUCTION

Every country be it a developed country or developing country would have had agriculture as its backbone initially as it serves as the food security and also as an employment generator. The goods produced from the agricultural sector later would help propel the economy of a nation by leading to the development of the secondary and tertiary sectors of the country. This shows how important the agricultural sector also known as the primary sector is important for any country.

Any issues in this industry would be catastrophic for not only the people but also the economy dependent on it. This crucial industry faces many challenges that need to be addressed and yet this industry uses primitive outdated methodology still which is not very effective in this current world with modern challenges.

One of the modern problems that this industry faces is global warming which causes climate change and indirectly causes sudden outbreak of many plant disease outbreaks which have a huge impact and even threaten the food security of many nations. Yet such problems are not faced or solved properly due to the lack of proper utilization of modern technology in this sector.

The method used to currently identify plant disease is by manually checking the plants and leaves individually which is time-intensive and manually expensive. Also, this method is only effective if the identifier knows about all the existing diseases of the plant. When he is met with a new variety of plant diseases, he might not be able to identify the disease in a short period before the disease spreads to all plants in the field. Due to the lack of early identification and mapping the disease can spread to vast lands and impact the agricultural output of the whole region.

These plant diseases may be due to several factors including climate change and global warming but also other factors including pathogens and biotic factors. Climate change is responsible for the changes in abiotic factors such as temperature, water, and soil health, which have a huge impact on plant defense and are often irreversible. Hence there is a huge need to immediately classify diseases with high accuracy and reliability to solve them, and contain them to minimize the impact of plant diseases on the agricultural sector of the economy

Plant diseases affect the quantity and quality of the crops produced which can affect the consumers who consume these products and also the reduced quantity can affect the livelihood of the people who depend on this sector as their only source of income.

Hence it is important to identify such diseases in their early stages and map their spread to minimize their impact. With technology improving every day we can use many new technologies to solve such problems. The deep learning field has shown promising outcomes in solving these new problems.

This field has however had its challenges such as it is relatively new and to develop such a model, we need a very vast set of datasets which must be of top quality, and the computing resources to train this model on. Also, they must be reliable, efficient, and accurate enough to be used on a large scale and be commercially viable.

The goal of this research is to build such a model with all the requirements and compare it with previous models. This study includes a large and diverse dataset of images to compare the effectiveness of this model. This paper aims to improve the performance of the model with proper development and use deep learning algorithms to improve the model and reduce human dependence to create an efficient model, and also deploy that model in the cloud so that it can be accessed easily from anywhere via proper internet and the data of the disease collected can be used to map the spread of the disease for early diagnosis and containing the spread of the disease.

2. RELATED WORK

The traditional way to classify plant diseases has been slow, time-consuming, and labor-intensive and not being able to find and fix these issues can affect the output of the produce. There have been several research in this field previously and we will look at what has been previously done.

Ahmed et al. [1] developed a model which has an accuracy of around 98.1% approx. and was trained with a large dataset. However, the model they developed had a huge size of around 10mb which could have been less as the size of the model plays a huge part in integrating the model into technologies such as Drones and the IoT field.

Arsenovic et al. [2] built a CNN model for plant disease classification using real-time photographs from real environments with different environmental scenarios such as in different lighting conditions and angles. They have generated 93.6% accuracy.

Kim et al. [3] built a CNN model for an apple leaf image classification model using the dataset from Plant Village and achieved an accuracy of 92.4%.

Ulutas et al. [4] built a model to train on the Plant Village dataset and their model when compared to other pre-trained models such as DenseNet and ResNet achieved around 94.5% accuracy. This model was both fast and efficient.

Yang et al. [5] showed how CNN models can help farmers in the diagnosis of diseases by analyzing images of sick plants and images of healthy plants. In addition, he put forward a proposal to develop a model that can be used to distinguish diseased and healthy leaves using image processing during the preprocessing stage of model development.

Zhang et al [6] used approx. 3000 images and evaluated many models such as ResNet, VGG, Inception v4, AlexNet, EfficientNet, and DenseNet for identification of diseases of cucumber. Image augmentation techniques were used to increase the size of the dataset to approx. 20,000 images. Their model generated 94% accuracy with the help of the RMSprop optimizer.

Chen et al. [7] proposed an advanced VGG model in which they replaced the activation layer, which resulted in a 92% increase in accuracy for disease classification of corn plants.

Xu et al. [8] have used a VGG1 pre-trained model to classify maize plant disease with a small dataset. This pre-trained model generated an accuracy of 95%.

Chowdhury et al. [9] proposed disease detection of tomato plants using EfficientNet and the accuracy achieved was 99.5%.

Hernández et al. [10] detected plant diseases using a Bayesian deep learning algorithm. The model was shown to achieve a harmonic mean accuracy of 53.4% with zero-shot learning.

Vasavi et al. [11] developed a Crop disease classification using a deep learning model but they trained their model only for very few species of plants and their disease.

Srivastava et al. [12] used AlexNet for plant disease classification with ReLU (rectified linear units) in a hidden layer instead of the tanh function normally used. ReLU helped them with faster execution than Tanh and their model achieved 94% accuracy.

Algulivev et al. [13] proposed a technique which was to combine CNN with GRU (Gated recurrent units) for the classification of disease in plants from the Plant leaf Dataset. The combined model generated an accuracy of 91.1%

3. MATERIALS USED AND METHODS

This section describes the materials used and the model developed for this study. This will cover all used technologies and background knowledge necessary to understand the proposed technique.

3.1 DATASET COLLECTION

The dataset used was collected from Kaggle which has 87867 images of plant leaves spread across 38 classes. This data was split into two folders for training and validation. 80% for training the model and the remaining for validation and testing the model. All images have dimensions of 256x256x3 representing the RGB channel format. Some of the images were resized according to the requirements. Microsoft Power Toys was used to resize the images so that they could fit the training model.

3.2 DATA PREPROCESSING

Data preprocessing is important to obtain as much data as possible to diversify the set of training as it will be useful to create accurate results. Color transformations were used to remove the noise from the images. To make the image fit the model and to reduce the size of the image Microsoft powertoys was used. Image preprocessing techniques used in this paper include color transformation, image enhancement, flipping, rotating, rescaling, and cropping. Image segmentation was used to split the image into various zones as it helps to extract useful information.

3.3 MODEL TRAINING

Pre-processed data is then used to train two pre-trained EfficientNet models which are EfficientNetB0 and EfficientNetB5 models which are customized to fit our dataset. We have used Jupyter Notebook to run our code via Anaconda Navigator to utilize the GPU power available in the PC to make the training faster. TensorFlow Keras package was used in Python to develop and train the model. We used Nvidia CUDA and Nvidia CuDNN software to make use of the local GPU resources available. Once the model has achieved higher accuracy and efficiency with low validation loss after certain epochs, we saved the model using the h5 extension for easy deployment in the cloud. Before deployment, the model was made to undergo rigorous testing to check its accuracy.

3.4 MODEL DEPLOYMENT

After the model was saved using the h5 extension we created a GCP (Google Cloud platform) account to deploy the model in the cloud. Once the account was created then the new project was created and inside it, we created a bucket to store the Deep Learning Model. Once the bucket was created, we deployed the model h5 file in the bucket. Then we created a cloud function which when called will accept the image as data and convert it into a 2D array to pass that data to the DL model to get the output. Once the predicted class and confidence its generated by the DL model it will return the output.

3.5 INTERFACE AND CENTRALISED DATABASE

The interface is important as it abstracts the user from the logic used behind it and makes sure that the user can easily use the interface without knowing how it works. Interface is built to allow the user to capture the image or browse the image from this local storage which then triggers the cloud function to receive the image from the user. Once the input image is received by the cloud the model uses that image to predict the class with confidence and once the plant disease has been identified then this plant disease name data along with the geolocation (latitude and longitude) of the user data and date of prediction is stored in a centralized database. Then the user is displayed the output in the interface. The spread of the disease can also be viewed from the user interface.

4. METHODOLOGY

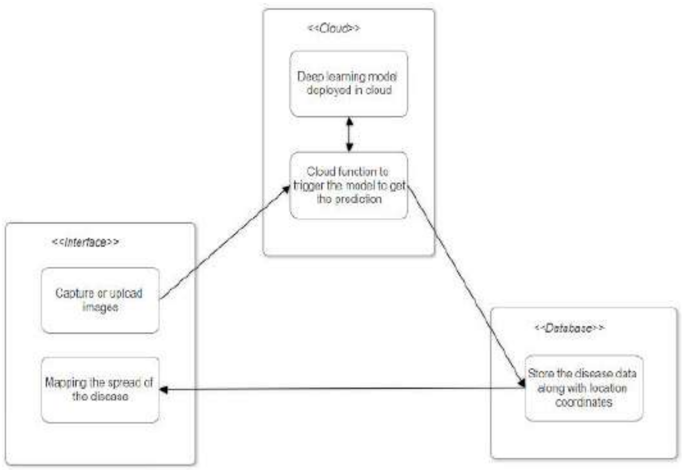

Figure 1: Block Diagram

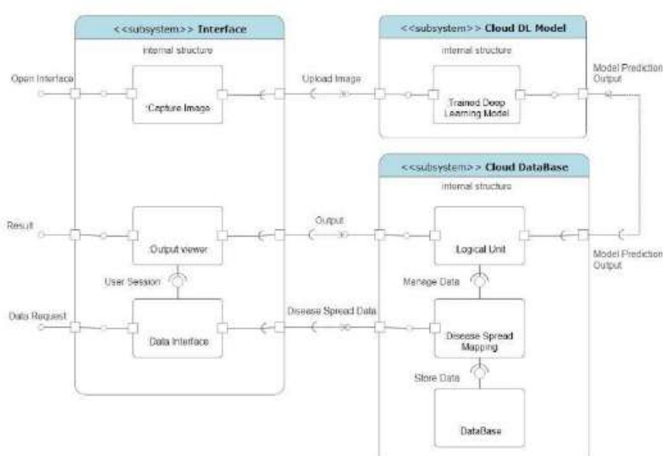

Figure 2: Component Diagram

4.1 CONVOLUTIONAL NEURAL NETWORK

CNN (Convolutional neural network) is a popular algorithm used in the field of Deep Learning for image recognition. Its unique feature is to apprehend unique patterns without the need for labor and time-intensive human intervention. CNN is used in many fields such as video feature extraction, speech recognition, image recognition and segmentation, computer vision, etc. CNN's early inspiration was from the working of neurons in the human brain.

CNN is a feed-forward network that consists of a pooling layer, a convolutional layer, and finally a fully connected layer in the end. CNN consists of the input layer, output layer, and a hidden layer in between the input and output layer. The hidden layer has ReLU as its activation function that sets all negative inputs to zero which makes it consume less computational resources and it introduces nonlinearity in the model.

The final layer uses a softmax activation layer to generate the probability values. These values are used to correspond to the probability of a certain class matching with the input.

4.2 TRANSFER LEARNING APPROACH

Transfer learning is useful because it helps train the model with less input and achieve high accuracy. The transfer learning approach is the usage of pre trained model to develop another model with a similar task. It helps reduce the time taken to develop a model from scratch with less computation power needed and less input to feed the model.

4.3 EFFICIENTNETB5

EfficientNetB5 belongs to the EfficientNet convolutional neural network family. This model strikes the balance between model size and computational efficiency. It has more parameters compared to other models present in the family which also makes it a more complex model. This model also requires more computational resources for training and inference because of its larger model size and complexity. But we can expect an improved accuracy on certain tasks. It is made to handle high-dimensional data which makes it suitable for image classification, segmentation tasks, and object detection. EfficientNetB5's pre-trained capabilities help it to use transfer learning to apply it across various domains.

4.4 EFFICIENTNETB0

EfficientNetB0 is the smallest variant in the EfficientNet family. It has fewer parameters compared to other members of its family. This also makes it suitable for scenarios where there are limited resources for training and inference of the model. It requires less computation resources and time to train a model. Since this model is also less complex with few parameters it is suitable for deployment in mobile devices and IoT. Even with its less complex model, its accuracy can be equally as good as other bigger and more complex models of its family. It is known for its compound scaling method that balances dimensions of width, depth, and resolution. It manages to reduce computational costs while maintaining expressive power. It is also popular in edge computing and computer vision tasks due to its need for less computation power.

5. IMPLEMENTATION

The study focuses on developing an efficient DL (Deep Learning) model using EfficientNet for plant disease classification and building a cloud-based model user interface for disease prediction and disease prevalence mapping. The input image is through data preprocessing, data multiplication, and data partition. Training images are overlaid at different resolutions. Layer construction is done to reproduce complex features to distinguish leaf objects from others. Each output produces a score that allows the disease with the highest probability to be selected as the classification result.

The steps of the Algorithm are described below

1) Upload the image dataset which has undergone data preprocessing, data augmentation, and data segmentation into the model.

2) Split the dataset into 80% for the training of the model and the remaining dataset images for the testing and validation of the model.

3) Choose either the EfficientNetB5 or EfficientNetB0 model as the base model for your model training.

4) Personalize the EfficientNetB5 with a customized dense layer of 256 neurons followed by an activation layer of ReLU which is followed by a set of normalization layers and finally with a dropout layer with the value of 0.3 and one more layer of dense layer.

5) Personalize the EfficientNetB0 model with a customized GlobalAveragePooling2D layer with 1280 neurons followed by an output layer (Dense).

6) Combine all these layers with the same number of neurons as the number of classes in the output layer. Then use the softmax function.

7) Develop a model that accepts the input and gives the output and make sure the model is using the categorical cross-entropy loss function and Adams optimizer.

8) Run and evaluate the EfficientNet model with a learning rate of 0.0005 initially and use the Adams optimizer for improved training performance and control.

9) Run and evaluate the model for 10-13 epochs and assess the performance for further evaluation.

10) Examine the overall performance with the data of the pre-trained model and new data collected from our customized model.

11) Upload the most suitable model in the cloud and write a cloud function to trigger the model when an input is provided.

12) Develop a central database that can store the disease name, the geolocation of the user, and the date of capture so that these data can be used to map the spread of the disease.

13) Develop the interface and connect it with the DL model in the cloud and with the central database.

Our model achieves its highest accuracy within 13 epochs which demonstrates the potential efficiency of our model. The EfficientNetB0 model achieves 99.82% validation accuracy in 10 epochs with each epoch taking 2 minutes and the EfficientNetB5 model achieves 99.37% validation accuracy

	precision	recall	f1-score	support
Apple___Apple_scab	1.00	1.00	1.00	58
Apple___Black_rot	1.00	1.00	1.00	53
Apple___Cedar_apple_rust	1.00	1.00	1.00	40
Apple___healthy	0.96	1.00	0.98	53
Blueberry___healthy	1.00	1.00	1.00	40
Cherry_(including_sour)___Powdery_mildew	1.00	0.97	0.99	37
Cherry_(including_sour)___healthy	1.00	1.00	1.00	47
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot	0.92	0.97	0.95	36
Corn_(maize)___Common_rust_	1.00	1.00	1.00	44
Corn_(maize)___Northern_Leaf_Blight	0.98	0.94	0.96	54
Corn_(maize)___healthy	1.00	1.00	1.00	48
Grape___Black_rot	1.00	1.00	1.00	35
Grape___Esca_(Black_Measles)	1.00	1.00	1.00	50
Grape___Leaf_blight_(Isariopsis_Leaf_Spot)	1.00	1.00	1.00	48
Grape___healthy	1.00	1.00	1.00	38
Orange___Haunglongbing_(Citrus_greening)	1.00	1.00	1.00	48
Peach___Bacterial_spot	1.00	1.00	1.00	44
Peach___healthy	1.00	1.00	1.00	43
Pepper,_bell___Bacterial_spot	1.00	1.00	1.00	47
Pepper,_bell___healthy	1.00	1.00	1.00	42
Potato___Early_blight	1.00	1.00	1.00	50
Potato___Late_blight	1.00	1.00	1.00	54
Potato___healthy	1.00	1.00	1.00	39
Raspberry___healthy	1.00	1.00	1.00	55
Soybean___healthy	1.00	0.98	0.99	53
Squash___Powdery_mildew	1.00	1.00	1.00	41
Strawberry___Leaf_scorch	1.00	1.00	1.00	43
Strawberry___healthy	1.00	1.00	1.00	45
Tomato___Bacterial_spot	0.98	1.00	0.99	46
Tomato___Early_blight	1.00	0.97	0.99	40
Tomato___Late_blight	0.98	1.00	0.99	52
Tomato___Leaf_Mold	1.00	1.00	1.00	51
Tomato___Septoria_leaf_spot	1.00	1.00	1.00	37
Tomato___Spider_mites Two-spotted_spider_mite	0.97	1.00	0.99	38
Tomato___Target_Spot	1.00	0.98	0.99	61
Tomato___Tomato_Yellow_Leaf_Curl_Virus	1.00	0.98	0.99	51
Tomato___Tomato_mosaic_virus	1.00	1.00	1.00	49
Tomato___healthy	1.00	1.00	1.00	48
accuracy			0.99	1758
macro avg	0.99	0.99	0.99	1758
weighted avg	1.00	0.99	0.99	1758

Figure 3: Classification Report of Customized EfficientnetB0 Model

Test Labels	Test Classes	Prediction Labels	Prediction Classes	Path	Prediction Probability
1	Apple___Black_rot	1	Apple___Black_rot	Dataset/Database/Images/valid\Apple___Black_ro...	0.999799
29	Tomato___Early_blight	29	Tomato___Early_blight	Dataset/Database/Images/valid\Tomato___Early_b...	0.998091
37	Tomato___healthy	37	Tomato___healthy	Dataset/Database/Images/valid\Tomato___healthy...	0.999985
24	Soybean___healthy	24	Soybean___healthy	Dataset/Database/Images/valid\Soybean___health...	0.999594
31	Tomato___Leaf_Mold	31	Tomato___Leaf_Mold	Dataset/Database/Images/valid\Tomato___Leaf_Mo...	0.999733
35	Tomato___Tomato_Yellow_Leaf_Curl_Virus	35	Tomato___Tomato_Yellow_Leaf_Curl_Virus	Dataset/Database/Images/valid\Tomato___Tomato_...	0.999994
31	Tomato___Leaf_Mold	31	Tomato___Leaf_Mold	Dataset/Database/Images/valid\Tomato___Leaf_Mo...	0.999638
19	Pepper,_bell___healthy	19	Pepper,_bell___healthy	Dataset/Database/Images/valid\Pepper,_bell___h...	0.999435

Figure 4: Prediction Probability of Customized EfficientnetB0 Model

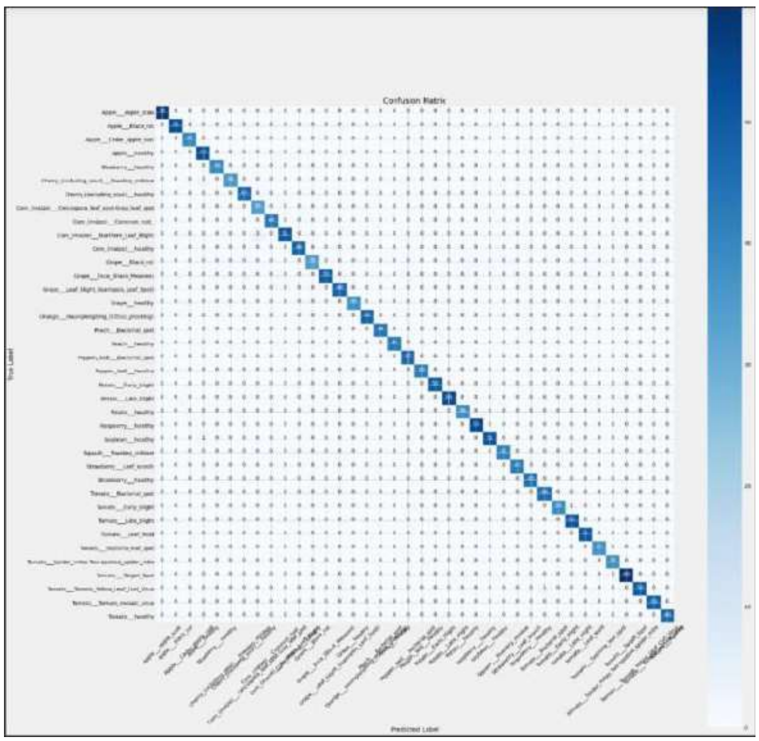

Figure 5: Confusion Matrix of Customized EfficientnetB0 Model

```
550/550 [==============================] - 32s 57ms/step - loss: 0.0078 - accuracy: 0.9982

[0.007836543023586273, 0.9981788992881775]
```

Figure 6: Accuracy of customized EfficientNetB0 Model

6. EXPERIMENTAL RESULTS

The model was developed, trained, and evaluated on Jupyter Notebook, using a GPU, python, TensorFlow, Keras, and SciKit. Adams optimizer along with regularisation techniques were used to prevent overfitting

After evaluating both EfficientNet models, EfiiciennentNetB0 was selected because it achieved the highest accuracy of 99.82% among other models. The uniqueness of this research is that we have used transfer learning along with the feature-extracting capabilities of pre-trained models. Our model with customized layers has contributed to the higher accuracy. It strikes a balance between accuracy and efficiency. We have also successfully developed an interface that connects with both the Deep learning model present in the cloud as well as the centralized database which is then used to map the spread of the disease.

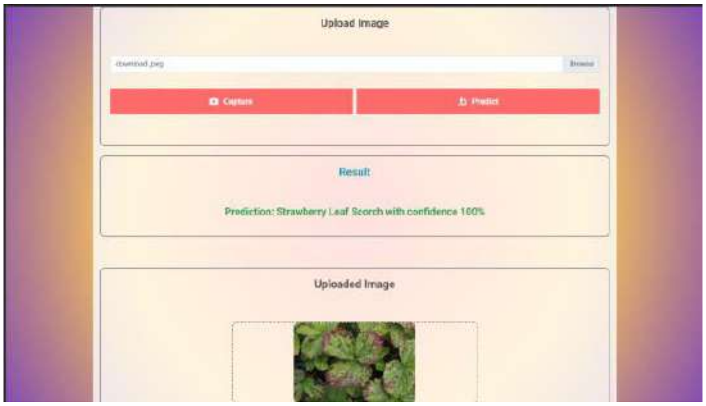

Figure 7: User Interface

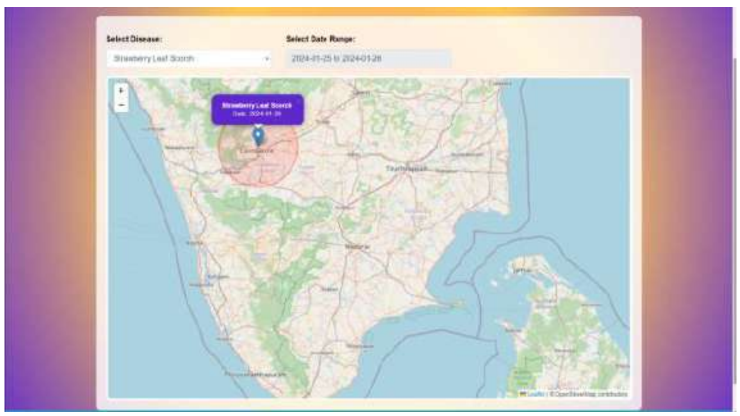

Figure 8: Visualization of data from centralized cloud database to identify the spread of the location of the plant disease

	Models	Accuracy
1	Ahmed et al.	98.10%
2	Arsenovic et al.	93.60%
3	Kim et al.	92.40%
4	Ulutas et al.	94.50%
5	Zhang et al.	94.00%
6	Chen et al.	92.00%
7	Xu et al.	95.00%
8	Srivastava et al.	94.00%
9	Algulivev et al.	91.10%
10	EfficientNetB5	99.32%
11	EfficientNetB0	99.82%

Figure 9: Accuracy VS Models

7. CONCLUSION

This paper aimed to develop an effective cloud-based deep learning model to diagnose plant disease and identify the location of the spread by building an interface to map and visualize the spread of the disease. The study compared the performance of the previously built models and pre-trained models with the customized EfficientNetB5 and EfficientNetB0 models for effective and accurate disease classification. The customized

EfficientNetB0 achieved the highest accuracy and validation accuracy at 99.98% and 99.82% respectively followed by customized EfficientNetB5 with 99.16% and 99.37% respectively followed by other pre-trained models and models used from references. The customized EfficientNetB0 model was selected to be deployed in the cloud as it has the highest accuracy and validation accuracy among all other models along with its added advantage of less time and computational resources required to train and inference. The model was deployed in a cloud bucket, a cloud function was created to trigger the function to get prediction results, the centralized database was created to map disease data with geolocation coordinates and date of capture, and an interface to make all these work together by getting image data from user and use the collected disease data from database to visualize the spread of the disease on a map allowing the user to quickly identify the disease and know about its current spread across a location quickly compared to traditionally manual processes. Future research can focus on using drones and IoT to capture real-time live data from the field and integrate it with the cloud in a similar way as we did with image classification.

REFERENCES

[1] Ahmad, Aanis & Saraswat, Dharmendra & Gamal, Aly. (2022). A Survey on Using Deep Learning Techniques for Plant Disease Diagnosis and Recommendations for Development of Appropriate Tools. Smart Agricultural Technology. 3. 100083. 10.1016/j.atech.2022.100083.

[2] Arsenovic, Marko, Mirjana Karanovic, Srdjan Sladojevic, Andras Anderla, and Darko Stefanovic. 2019. "Solving Current Limitations of Deep Learning Based Approaches for Plant Disease Detection" *Symmetry* 11, no. 7: 939. https://doi.org/10.3390/sym11070939

[3] Kim, Manbae. (2021). Apple Leaf Disease Classification Using Superpixel and CNN. 10.1007/978-3-030-71051-4_8.

[4] Ulutaş, Hasan, and Veysel Aslantaş. 2023. "Design of Efficient Methods for the Detection of Tomato Leaf Disease Utilizing Proposed Ensemble CNN Model" *Electronics* 12, no. 4: 827. https://doi.org/10.3390/electronics12040827

[5] Zhang, Pan & Yang, Ling & Li, Daoliang. (2020). EfficientNet-B4-Ranger: A novel method for greenhouse cucumber disease recognition under natural complex environment. Computers and Electronics in Agriculture. 176. 105652. 10.1016/j.compag.2020.105652.

[6] Ma, Juncheng & Du, Keming & Zheng, Feixiang & Zhang, Lingxian & Gong, Zhihong & Sun, Zhongfu. (2018). A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. Computers and Electronics in Agriculture. 154. 18-24. 10.1016/j.compag.2018.08.048.

[7] Chen, Junde & Chen, Jinxiu & Zhang, Defu & Sun, Yuandong & Nanehkaran, Yaser. (2020). Using deep transfer learning for image-based plant disease identification. Computers and Electronics in Agriculture. 173. 105393. 10.1016/j.compag.2020.105393. [8] J.-H. Xu, M.-Y. Shao, Y.-C. Wang, and W.-T. Han, ‘‘Recognition of corn leaf spot and rust based on transfer learning with convolutional neural network,’’ Trans. Chin. Soc. Agricult. Mach., vol. 51, no. 2, pp. 230–236, Feb. 2020.

[9] Chowdhury, Muhammad E. H., Tawsifur Rahman, Amith Khandakar, Mohamed Arselene Ayari, Aftab Ullah Khan, Muhammad Salman Khan, Nasser Al-Emadi, Mamun Bin Ibne Reaz, Mohammad Tariqul Islam, and Sawal Hamid Md Ali. 2021. "Automatic and Reliable Leaf Disease Detection Using Deep Learning Techniques" *AgriEngineering* 3, no. 2: 294-312. https://doi.org/10.3390/agriengineering3020020

[10] Hernández, S. & López, Juan. (2020). Uncertainty quantification for plant disease detection using Bayesian deep learning. Applied Soft Computing. 96. 106597. 10.1016/j.asoc.2020.106597.

[11] Vasavi, P., Punitha, A. and Rao, T.V.N., 2022. Crop leaf disease detection and classification using machine learning and deep learning algorithms by visual symptoms: A review. International Journal of Electrical and Computer Engineering, 12(2), p.2079.

[12] Srivastava, Pallavi & Shukla, Aasheesh & Bansal, Atul. (2021). A comprehensive review of soil classification using deep learning and computer vision techniques. Multimedia Tools and Applications. 80. 10.1007/s11042-021-10544-5.

[13] Alguliyev, Rasim & Imamverdiyev, Yadigar & Sukhostat, Lyudmila & Bayramov, Ruslan. (2021). Plant disease detection is based on a deep model. Soft Computing. 25. 10.1007/s00500-021-06176-4.