 https://doi.org/10.1051/itmconf/20224403067

Plant Disease Identification – A portable mobile application system

Diksha Nigam[1, *], *Anurag* Patil[1], *Pratik* Murari[1] and *Puja Padiya*[1]

[1]Ramrao Adik Institute of Technology, Computer Engineering Department

Abstract. Plant Disease Classification System is software that takes the image of the plant leaf and identifies the possible plant disease at an early stage. In plants, disease indications usually occur on leaves, fruit, buds, and young branches. This situation causes fruit to be wasted (to drop) or be damaged. In addition, these diseases lead to the formation of new infections and the spread of the disease for reasons such as seasonal conditions. For this reason, it is very important to determine the disease in advance and to take the necessary precautions before it spreads to other trees. This project proposes a user-friendly, portable, scalable, and accurate way for identifying, and treating plant diseases at an early stage using Convolution Neural Networks. This will help in the timely and accurate diagnosis of plant diseases, which plays an important role in preventing the loss of productivity and quantity of agricultural products.

1 Introduction

Food safety and plant health are closely related. Pests and diseases threaten food security by causing the loss of 20–40% of global food production. Plants can be protected from these infestations by using pesticides. However, the use of such substances is environmentally harmful. In plants, the indication of disease is typically found on leaves, buds, fruits, and young branches. As a result, fruit is wasted or damaged when it falls from the plant. Plant diseases are typically detected manually by botanists or agricultural engineers, first by visual inspection and then by testing in a laboratory. It is often time-consuming and complicated to use these traditional methods. This is why machine learning and image processing have become important for automatically identifying diseases. Using a visual inspection to diagnose plant diseases can be of great benefit to users who have little to no experience in farming.

The main objectives of the proposed system are early and easily accessible detection of plant disease and suggestion of precautions to avoid the loss of agricultural goods.

1.1 Related Work

1.1.1 Survey of existing systems

In Zhang, K.; Wu, Q.; Liu, A.; Meng, X. [1], experimental result shows that multilayer convolutional neural networks (MCNN) are effective in identifying tomato leaf disease, and they could be used to identify other plant diseases.

In M, TURKO ̈ GLU.; D, Hanbay. [2], the results ̆ show that deep feature extraction and SVM/ELM classification produced better results than transfer learning but both of them had complex computations not suitable for the type of system we are aiming for.

Singh, U.P.; Chouhan, S.S.; Jain, S.; Jain, S.[3] validated on a real time dataset of mango leaves using Multilayer Convolutional Networks(MCNN).

J, Chen; Q, Liu.; L, Gao.[4] used accuracy and mean accuracy indices to evaluate the algorithm. In order to extract the disease-related characteristics of tea plants automatically from images, a CNN model named LeafNet was developed using different sized feature extractors.

Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E.[5] introduced AlexNet which achieved a winning top-5 test error rate of 15.3% compared to 26.2% achieved by the second-best entry in the ImageNet LSVRC-2010 contest. A large dataset of 1.2 million high-resolution images was used for training.
The rest of the papers referred are the review papers which helped as gain an insight into various possible algorithms and their uses.

1.1.2 Limitations of existing systems

Study of existing systems have shown that Existing software is complicated and does not provide an easy, intuitive experience. It is not reliable because it is prone to human error. Accurate ML implementations require high processing power, they are not scalable hence not commercial. The disadvantage of current systems is that they require a lot of resources and do not give fast performance. Another method is with the help of experts

* Corresponding author: nigamdiksha02@gmail.com

but not everyone has access to resident experts in this field.

1.2 Problem Statement

To develop a user-friendly and portable system that uses image analysis to identify the plant disease using AlexNet (CNN) and suggest the precautions that need to be taken in order to save the plant at an early stage. Proposed method uses one of the Convolution Neural Network's variation, AlexNet to analyze and visualize imagery of 38 different plant diseases.

1.2 Scope

The system for Plant Disease Detection involves training the AlexNet model on a dataset of 300,000 plant leaf images consisting of 26 diseases and 11 healthy samples. To make the system a user-friendly, cross-platform application made using Flutter to run on Windows, Linux, MacOS and Web. Build a Simple UI to upload the image of the plant leaf and get the results which include the identified disease, probability of the identified disease compared to other diseases, essential resources, etc. Make the application multilingual to make it accessible across the globe. Provide an in-app chat option for users to make networks with other users and plantsman(https://en.wikipedia.org/wiki/Plantsman).

2 Proposed Work

Proposed method uses one of the Convolution Neural Network's variation, AlexNet to analyze and visualize imagery of 38 different plant diseases. An artificial neural network, such as a convolutional neural network, is used primarily for applications involving image recognition. Several layers called perceptron's are used to learn the features present in images with great detail. The term "convolution" refers to the process of understanding an image's features. Convolution is required to extract these features. Filters are used to extract these features, we specify how many filters to use (Kernels). Alexnet contains a total of eight layers with weights; where five of them are convolutional and three are maxpooling layers. These are followed by three layers that are fully connected. A 38-way softmax layer produces distribution over 38 class labels using the output of the last fully-connected layer.

2.1. Proposed system

The system proposes to develop a user-friendly and portable system that uses image analysis to recognize the plant disease using AlexNet (CNN) model and suggest the precautions that need to be taken in order to save the plant at an early stage. Proposed method uses one of the Convolution Neural Network's variation, AlexNet to analyze and visualize imagery of 38 different plant diseases

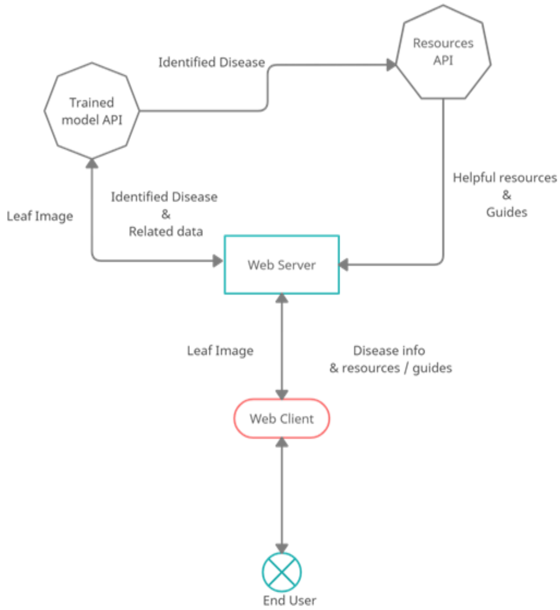


Fig. 1. The system architecture is depicted in the figure. The end user inputs a leaf image through the user interface. This input is processed and sent to trained model for prediction. The model predicts the disease and sends it to resources api which in turn fetches helpful resources and guides.

2.2 Details of Hardware/Software Requirement

The surveyed papers show that most of the good performing techniques are either computationally expensive to implement or have a huge error margin. Thus we wanted to implement such a technique which can be scaled and provides better results while requiring less complex computations. We used SQLAlchemy as the database. To make the application scalable we wrapped the backend written in python with starlette api.

Table 1. Hardware Specifications

Processor	**Intel Core i5**
RAM	**4GB**
Operating System, Server	**Windows 10,Linux,Unix**

Table 1. Software Specifications

Database	**SQLAlchemy**
Client	**Google Chrome, Microsoft Edge or any other web browser**
Development Tools	**Microsoft Visual Studio**
Programming Language	**Python, HTML, CSS, Java, Dart, Kotlin**

2.3 Implementation Details

In the back-end, Python API written in Starlette is used to serve the CNN Model. Client facing API to be written in Java (Kotlin) using SpringBoot. PostgresSQL will be used to store the backend side data of users. Redis is used for cache. Nginx server is used for load-balancing. We built a user-friendly cross-platform application with Flutter to work with Windows, Linux, MacOS and Web. SharedPrefrences is used as clientside database. Dart is used as the language because of its null-safety, robust and strict nature. We implemented AlexNet CNN model for the disease prediction system. We compared the performance of AlexNet, GoogleNet and VGG models as these are most widely used high performing models in competitions.

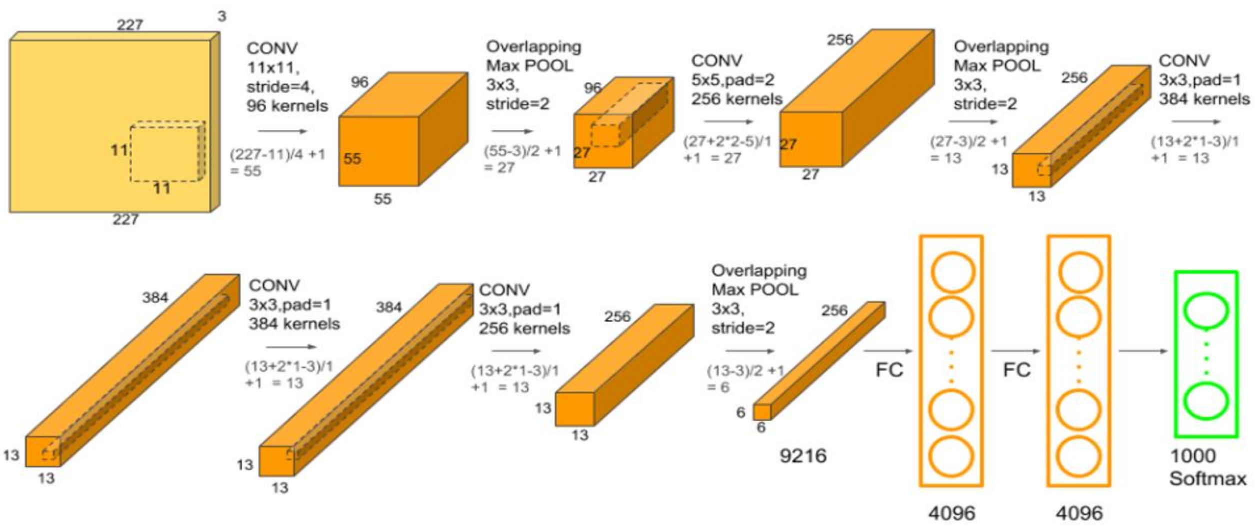


Fig. 2. AlexNet Architecture[10]. The figure depicts the architecture of AlexNet with it's 5 convolution layers, 3 max-pooling layers and 3 fully connected layers.

As depicted in Fig. 2, there are 11 layers in AlexNet architecture as:
Layer C1: Convolution Layer (96, 11*11)
Layer S2: Max Pooling Layer (3*3)
Layer C3: Convolution Layer (256, 5*5)
Layer S4: Max Pooling Layer (3*3)
Layer C5: Convolution Layer (384, 3*3)
Layer C6: Convolution Layer (384, 3*3)
Layer C7: Convolution Layer (256, 3*3)
Layer S8: Max Pooling Layer (3*3)
Layer F9: Fully-Connected Layer (4096)
Layer F10: Fully-Connected Layer (4096)
Layer F11: Fully-Connected Layer (36)

In the first convolutional layer, 96 kernels of 11*11*3 sizes with a stride of 4 pixels filter the 224*224*3 input image (this represents the distance between the receptive field centers of neighboring neurons). 256 kernels of size 5*5*48 are used to filter the output of the second convolutional layer. It takes the normalized and pooled output of the first convolutional layer as input. There are no intervening layers between the third, fourth, and fifth convolutional layers. There are 384 kernels in the third convolutional layer of 3 * 3 * 256 with the outputs of layer two connected (normalized, pooled). There are 384 kernels in the fourth convolutional layer, and 256 kernels in the fifth convolutional layer. Each of the 5 fully connected layer has 4096 neurons. Alexnet has around 60 million parameters.
Therefor the overall system consists of a user interface, a machine learning model and a resource API. All these components are programmed in Python language. The system is also integrated with an in-app chat system using web sockets which allows user to interact with fellow users in the system. We built a scalable backend, which can even be hosted on low end machines as prediction is very fast and memory efficient.

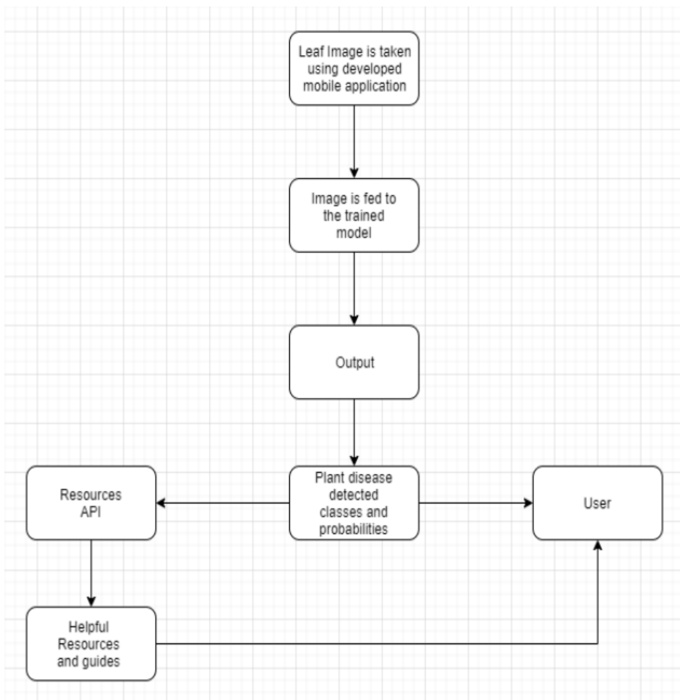


Fig. 3. Flow Chart for the system

Figure 2 shows a flowchart for the overall disease identification system implemented. A leaf image is

provided as an input through the mobile application. Image is fed to the trained model. Output of the system is in the form of array of key-value pairs of plant disease detected classes mapped to their probabilities. This information is displayed to the user in the form of bar graph with predicted classes probablitites. These probabilities is again fed to the resource api to derive helpful resources and guides for the highest probability predicted disease.
The dataset used is Plant Village Dataset [9]. It consists of images of leaves of diseased plant with their corresponding labels. It has 300,000 plant leaf images consisting of samples of 26 plant diseases and 11 healthy samples. Plants included in the dataset are commonly found all over the world, hence the application can be used globally.

3 Result Analysis

We Operated with 3-channel images that were (224*224*3) in size. We used max pooling and ReLU activations when subsampling. We used ReLu activations for convolutions. We used (3*3) kernels for max pooling. We used either (11*11), (5*5), or (3*3) kernels for convolutions. All this was done to classify images into one of 38 classes. We built a user friendly mobile application, in which we added support for multiple languages in the application. It also includes a realtime chat feature with other users. We added better and improved resources when the user is shown predicted disease and other results. It also shows news / posts for gardners from internet.

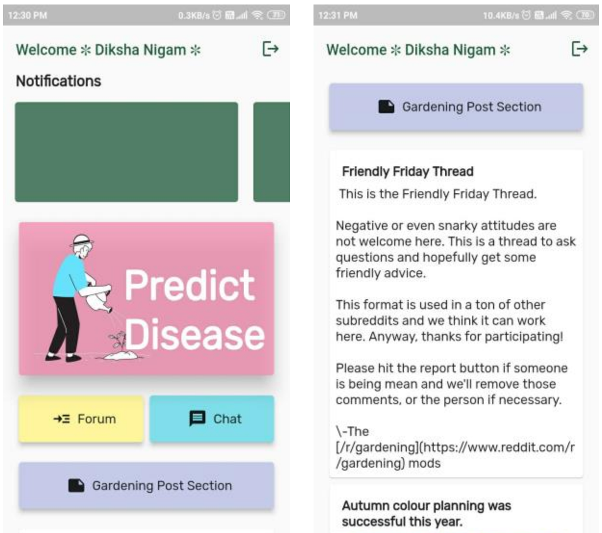


Fig. 4. The first screen shows the start of main page which contains notifications feature, disease prediction option and chat and forum options. The second screen shows the below section of main page containing the daily newsletters and posts.

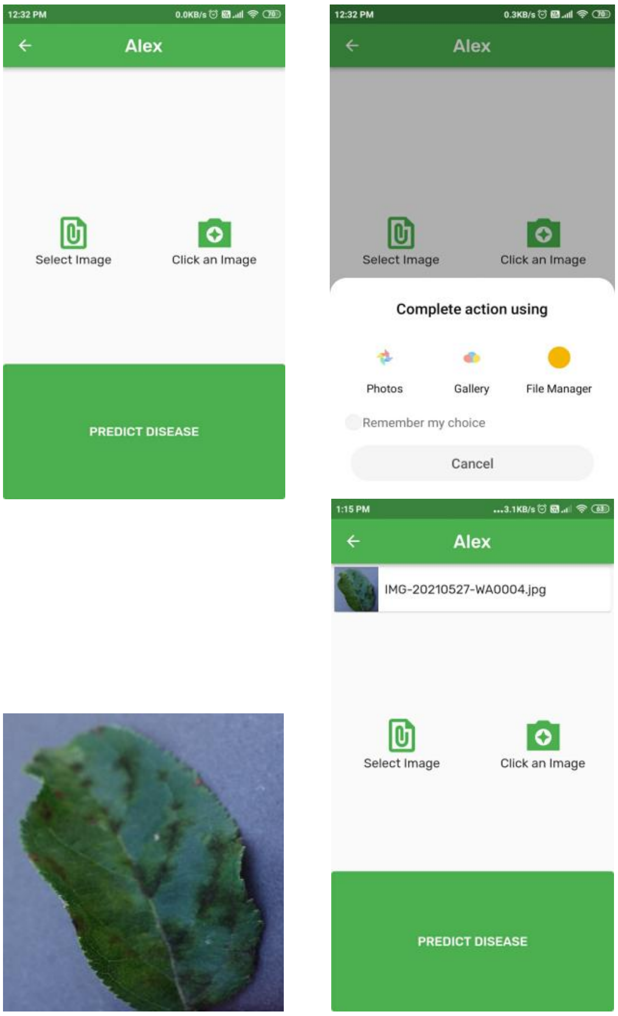
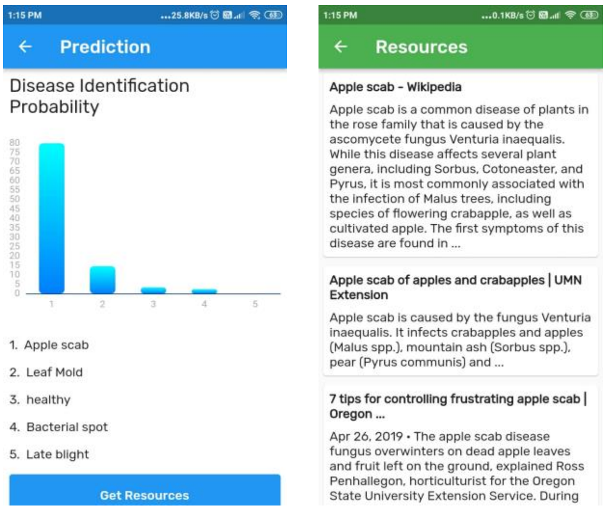


Fig. 5. The process of disease prediction is depicted in the picture. The image can be taken through a camera or from the image gallery. The prediction shows the top 5 probabilities of diseases.



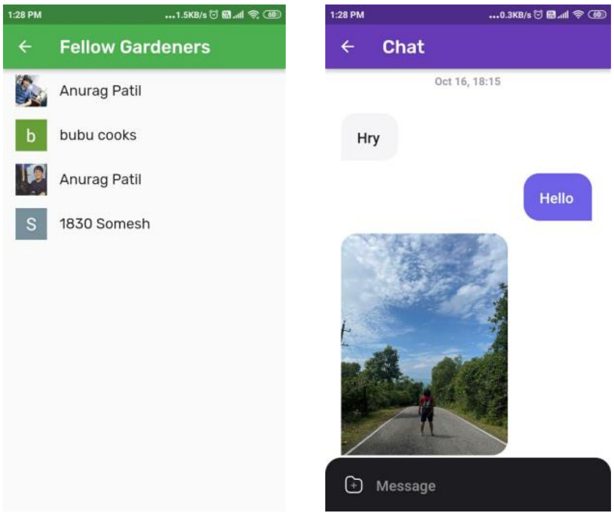


Fig. 6. Resources related to the predicted disease is made available to the user. The user can also interact with other users through personal chats and chat forums.

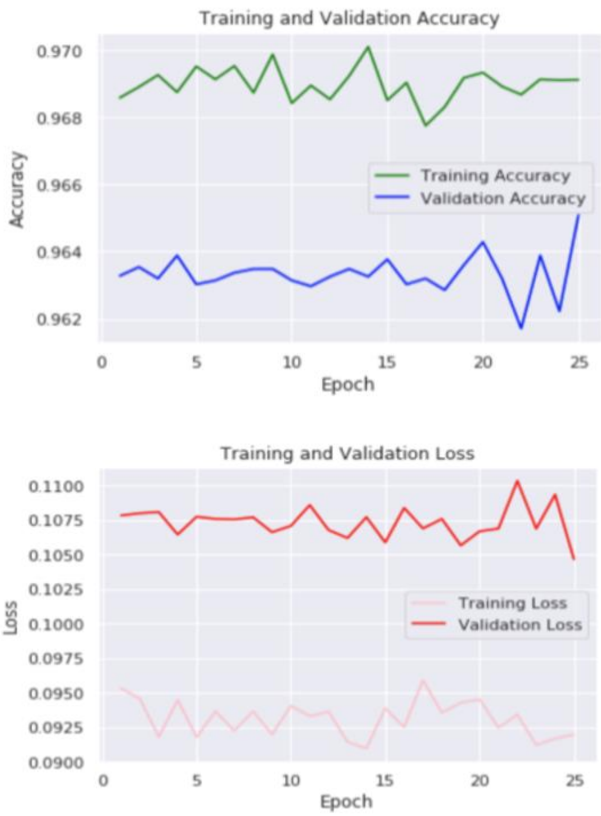


Fig. 7. Graphs showing the trend of accuracy over epoch and validation loss vs epoch.

In order to avoid overfitting, we used a technique called cross-validation to test whether the model is over-fitting. The data is divided into two parts - a training and a validation set. A model is trained on a training set, and then it is evaluated on a validation set. With this in mind, loss and accuracy are calculated on the training set and validation set. As seen from figure 5, the training accuracy and validation accuracy achieved by the model are 97% and 96.6% respectively.

References

1. K. Zhang, Q. Wu, A. Liu, and X. Meng, “Can deep learning identify tomato leaf disease?”, Advances in Multimedia, vol. **2018** (2018)
2. M. Turko¨ glu and D. Hanbay, “Plant disease and pest detection using deep learning-based ˘ features,” Turkish Journal of Electrical Engineering & Computer Sciences, vol. **27**, no. 3, pp. 1636–1651 (2019)
3. U. P. Singh, S. S. Chouhan, S. Jain, and S. Jain, “Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease,” IEEE Access, vol. 7, pp. 43721–43729 (2019)
4. J. Chen, Q. Liu, and L. Gao, “Visual tea leaf disease recognition using a convolutional neural network model,” Symmetry, vol. 11, no. 3, p. 343 (2019)
5. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Communications of the ACM, vol. 60, no. 6, pp. 84–90 (2017)
6. K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556 (2014)
7. J. Kaur, R. Chadha, S. Thakur, and E. R. Kaur, “International journal of engineering sciences & research technology a review paper on plant disease detection using image processing and neural network approach” (2016)
8. S. Kumar and R. Kaur, “Plant disease detection using image processing-a review,” International Journal of Computer Applications, vol. 124, no. 16, (2015)
9. Plant Village Dataset- Dataset of diseased plant leaf images and corresponding labels https://github.com/spMohanty/PlantVillage-Dataset
10. https://neurohive.io/wp-content/uploads/2018/10/AlexNet-1.png