PLANT DISEASE CLASSIFICATION USING CONVOLUTION NEURAL NETWORK(CNN)

Dasari Srinivas[1], Eluri Abhishek Gupta[2], Yempalla Rakesh Reddy[3], P Rajeshwari[4]

[1,2,3]Student, Department of Computer Science and Engineering,

Anurag Group of Institutions.

[4]Assistant Professor, Department of Computer Science and Engineering,

Anurag Group of Institutions.

Abstract

Plant disease is an ongoing challenge for current day to day life situation, which impacts in the reduction of food production.Image classification in agriculture has craved its opportunity by the advancements in computer vision models .Convolutional Neural Networks (CNNs) is the bestfor image recognition and provide accurate model. In this paper, the performance of a pre-trained AlexNet weightsare used to create a model . The implemented model is built and deployed as a web application which runs on the server and is able to classify the identified diseases among the 38 total classes present as healthy or not. A controlled environment is established and 87,867 images are collected for training and validation of the model. Validation results are the key holders showing that the proposed method is achieving an accuracy of 94% and F1 score greater than 94%. In this paper we can see the accurate prediction given by using CNN.

Key words - *Deep learning , Plant disease classification, AlexNet architecture, CNN*

1.Introduction

Leaves play a important role in providing us the insights on nature and the amount of the crop yield. Many factors like climate change, fertility, seasons, presence of weed affects the production of food. Leaving that,these plant diseases are held responsible in the reduction of agricultural products and loss of economy[1]. The delay or failure to identifythese virus/bacteria/any other form of disease leads to insufficient insecticides/pesticides use over them. Hence, the analysis of this has got a start.In detecting these various forms of illness, the conventional methods of human investigation i.e manual collection and the visual assessment is not a practical option. The improvement of PC vision models offers a fast,normalized and exact answer for this issue. Once prepared, a model can likewise be sent as an application . Simple to use, everything necessary is a web application and camera equipped cell phone.
Exploiting the digital image processing techniques prevailed such as colour and thresholding[2] were used to achieve the plant disease detection and its classification. In recent studies Deep Learning (DL) has come into picture for real-life object detection, recognition and classification purposes and used on the plant disease detection and classification[3].

The utilization of CNNs in plant sickness arrangement and detection[4] has accomplished astounding outcomes lately . Complex capacities like ReLU nonlinearity and pooling are a common components prevailing within the present day design. Such advancements have assisted with decreasing preparing time and error rate. Most importantly, the development of engineering has been a fundamental interest of enormous and complex 21st century datasets.Moreover, the research community focus shifted for better optimisation of weight parameters of newralnetworks[5].

Data pre-processing is vitally critical to a model's execution. Viral, bacterial and contagious contaminations can be hard to recognize, regularly sharing a cover of indications. These side effects can be any quantifiable distinction in shading, shape or capacity which result as the plant reacts to the microbe. Due to this intricacy, it is desirable over use RGB information. This produces clear, commotion free pictures which may take longer than greyscale information to prepare, yet generally are more appropriate for plant sickness distinguishing proof models[6].
Further more, many neural networks were constructed which were tested on this plant diseases too, which included the GoogLeNet, AlexNet, VGG-16, Inception-v3 and ResNet, which achieved notable outcomes.

2.Goals of research

1) Determining the model's overall performance in classification of the disease using both the validation and the test dataset.
2) Comparing the accuracy of model with tests using various sizes of images and augmentation variations.
3) Deploying the trained model as a web application interface.

Because of a lopsided class dissemination, the precision and f1-score measurements together will be analyzed in getting the execution of the model. When the model arrives at a precision and F1score more prominent than 90%, it will be acknowledged.

3.Methodology

3.1. Architecture

A diseased plant leaf image is uploaded by the user from their web browser. The uploaded image is then cropped in so that it's final dimensions after cropping are 224*224*3 which are in accordance with the model which is trained over the same dimensional and pre-processed images. Now this image is fed into the model for prediction. The output is a single disease name which is one of the 38 classes as mentioned in table 1. This prediction is displayed to the user.

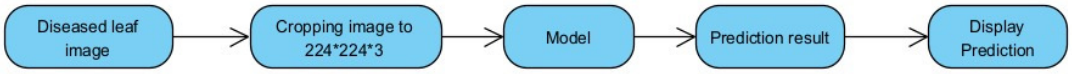


Figure 3.1.1 : Architecture Diagram

AlexNet[7]weights are used in our research .We use sequential model as the base neural network to build layers on one other. A total of 8 layers were built in this model. The input size of each image is considered as 224*224*3 pixels. The parameters used to fit model are: categorical cross entropy, Softmaxactivation ,ReLU activation function with learning rate of 0.01 and number of epochs are 11(optimized using EarlyStopping and checkpoints) .The principal layer in our model is a full convolutional layer. All layers are trailed by group normalization and ReLU as our activation function. As a part of transfer learning, all layers are frozen with the exception of lastfour layers. Now these frozen layers containing the new updated weights are utilized for the plant disease detection and classification task.Restricting the backpropagation of gradients, the freezing of layers allows us to train disease specific.

3.2. Dataset

The dataset is contains a total of 87,867 images(augmented) collected from Kaggle repository[8]. Model bias is to be expected since all images are captured in a controlled environment .And an extra set of test images.

The following was the annoting of our dataset for the plant disease identification done by the DL architecture. The images were annotated using Labelling, which is a method used for image annotation.

Annotation Label	No.of.Images(train set)	No of Images(valid set)
Apple__Apple_scab	2016	504
Apple__Black_rot	1987	497
Apple__Cedar_apple_rust	1760	440
Apple__healthy	2008	502
Blueberry__healthy	1816	454
Cherry_Powdery_mildew	1683	421

Cherry_healthy	1826	456
Corn_Cercospora_leaf_spotGray_leaf_spot	1642	480
Corn_(maize)__Common_rust_	1907	472
Corn_(maize)__Northern_Leaf_Blight	1908	477
Corn_(maize)__healthy	1859	465
Grape__Black_rot	1888	472
Grape__Esca_(Black_Measles)	1920	480
Grape__Leaf_blight_(Isariopsis_Leaf_Spot)	1722	430
Grape__healthy	1692	423
Orange__Haunglongbing_(Citrus_greening)	2010	502
Peach__Bacterial_spot	1838	454
Peach__healthy	1728	432
Pepper_bell__Bacterial_spot	1913	478
Pepper,_bell__healthy	1988	497
Potato__Early_blight	1939	485
Potato__Late_blight	1939	485
Potato__healthy	1824	456
Raspberry__healthy	1781	445
Soybean__healthy	2022	505
Squash__Powdery_mildew	1736	434
Strawberry__Leaf_scorch	1774	444
Strawberry__healthy	1824	456
Tomato__Bacterial_spot	1702	425
Tomato__Early_blight	1920	480
Tomato__Late_blight	1851	463
Tomato__Leaf_Mold	1882	470
Tomato__Septoria_leaf_spot	1745	436
Tomato__Spider_mitesTwo-spotted_spider_mi	1741	435
Tomato__Target_Spot	1827	457
Tomato__Tomato_Yellow_Leaf_Curl_Virus	1961	490
Tomato__Tomato_mosaic_virus	1790	448
Tomato__healthy	1926	481

Table 3.2.1:**Image Distribustion Under train data and validation data**

3.3.DataPreprocessingand Data Augmentation

Data pre-processing consists of series of steps to transform raw data derived from data extraction into “clean” and “tidy” dataset prior to any form of analysis. Assessing and improving the quality of data are the aims of this pre-processing to allow for a reliable analysis and results.

We have induced the technique of dataaugmentation[9] which is by the application of various distinct transformations to the original images which generates multiple transformed copies of the original image provided as input. Based on the augmentation techniques applied the images differ from each other in certain aspects.

In our model generation we applied the techniques of random rotation, vertical and horizontal shifts of pixels, flips both vertical and horizontal, zoom variants and the level of brightness to the images in our dataset.

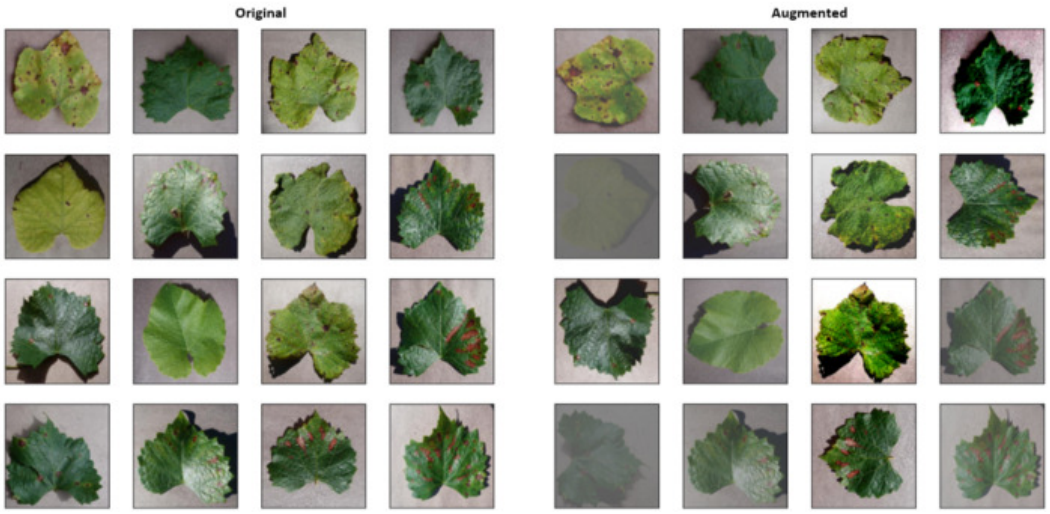


Figure 3.3.1: Random sample of Grape___Esca_(Black_Measles) disease – 16 images, to show the image augmentation

This is provided by the KerasImageDataGenerator[10] class which generates the augmented images by the model at that momentas a part of the training stage. Also these newly formed images are not added to the original store of images. Original images duplication would have caused anoverfitto our model.

These image augmentation techniques expands the size of your dataset, provides a generated level of variation to the dataset allowing generalisation over unseen data. Thismakesthe model robusti.e reducing the overfit of model[11] when trained over the new and slightly transformed and altered images.ImageDataGenerator requires lower memory usage as all the images are loaded at once. By using this, the images are loaded batch-wisehence saving memory.

3.4.Neural Network Training

For the purpose of this research, AlexNet architecture is used, combined with the weights learned on a very large dataset, ImageNet. This AlexNet consists of a total 8 layers: 5 convolution layers and 3 fully-connected layers .AlexNet uses Rectified Linear Units (ReLU) instead of tanh function, which was astandard previously. ReLU's advantage is shown training time, it runs six times faster than a CNN using tanh as the activation function which was tested on the CIFAR-10 dataset.

These ReLU[12] are used as substitute for saturating nonlinearities. This activation function learns the parameters of the rectifies and improves accuracy at a negligible computational cost[13]. It is defined as

$$f(z_i) = \max(0, z_i)$$

where, z_i is the input of the nonlinear activation function f on the i^{th} channel.

This architecture can also help to cut down the training time by allowing multi-GPU training, by putting half of model's neurons on one GPU and the other half on another GPU. It was found to reduce the error by 0.5% with the help of 'Overlapping Pooling'. With all these advantages this architecture was used.Every layer is accompanied by BatchNormalization layer, a regularization technique to improve the model and allows it to converge faster. To reduce the overfitting, a dropout layer[14] is added to the fully connected layers. The usage of ReLU and dropout together is advantageous which is proved the the experiments of Bayesian Optimisation which indicates that they have synergy effects[15].

The dataset is divided into 80% for training and 20% for validation. Augmentation techniques are applied to the training data.Each image size is considered as 224*224*3 with 150 as the batch size.

The concept of transfer learning freezed layers is used. The first 2 convolution blocks are freezedand leaving the remaining layersunfreezed. This allows the model to perform identical computation through the freezed layers hence accelerate neural network training progressively[16]. The end layers are open to modification meaning their weights are bound to change during training And, when further increase of the data occurs we can then reuse that model by just replacing or adjusting the last layer and have the model classify some new image categories. The more data we have for our task the more we can unfreeze the layers of the original model and fine tune them for the specific task.

We considered a total of 11 epochs that is adjusted by the model by using EarlyStopping[17] method.EarlyStopping is performed on the val_loss parameter. This helps to find the optimal number of epochs to use. Early stopping is a method that allows you to specify an arbitrary large number of training epochs and stop training once the model performance stops improving on a monitored validation dataset.

epochs	loss	acc	vali loss	vali acc	test acc
5	4.88	73.2	0.1378	96.05	0.95
8	4.51	89.4	0.1782	95.58	0.94
10	4.51	66.1	0.1842	95.63	0.94
12	4.34	85.2	0.2075	95.34	0.94
15	4.37	68.8	0.24	95.07	0.93
18	4.25	73.6	0.2601	94.67	0.93
20	4.27	64.9	0.2668	94.98	0.93
25	4.2	66.2	0.3013	94.45	0.92

Table 3.4.2 : loss and accuracy changes related to epochs

To verify over the EarlyStopping method result, the model is generated over various considerations of epochs count. Based on the results which can be seen in the Table 2. The EarlyStopping found the model to go overfit or underfit and this can be seen in the training accuracy with a great fluctuation and from the increase over validation loss.

An additional callback of Keras known as ModelCheckPoint[17] is used to save the model with the bestperformance on the validation dataset as the model at the end of training may not be the one with the best performance. The method monitors validation accuracy and maximizing the performance measure. And this model is stored in .hdf5/.h5 format which is used in the generated web application for prediction.

3.5. User Interface – Web Application

The web application is developed using Flask which provides a connectivity between the user interface and the model. It has three main steps. First step involves loading of the model when the user launches the application. It hardly takes 30 seconds for the model to load. In the second step, the user is prompted to upload an image for the purpose of detection of the plant disease. The third step involves displaying the results. Once when the user uploads the image and clicks on submit/analyse, the image is passed as the input to the model and the predicted output from the model is displayed as the result.

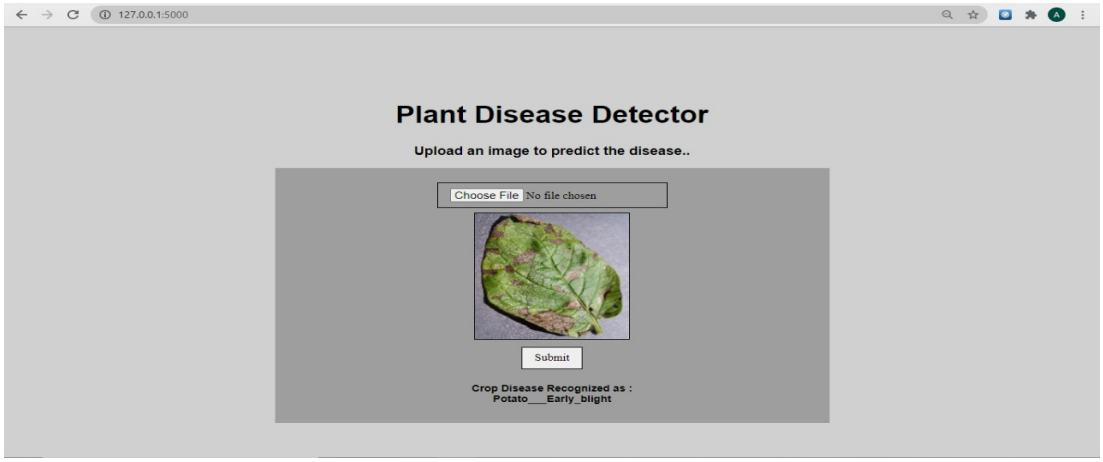
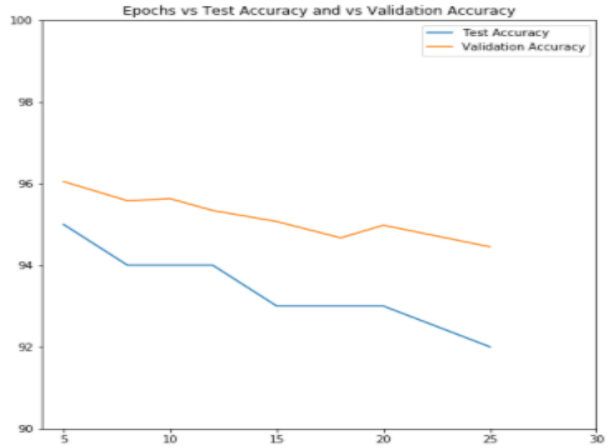


Figure 3.5.1: Graphical User Interface of Web Application

4. Results and Discussions

The accuracy obtained with the generated best fit and best performance model is 94%. The below data shows the accuracy of model while training and validation across epochs.



The below data shows the loss of the model while training and validation across epochs.

What we can infer from this is that the validation loss took a slight increase when it crossed the 10 epochs point and the decrease in the training loss which became steady without any further noticeable decrease. These lead to the conclusion led by the callback of kerasEarlyStopping class.

And the high values of precision and recall achieved by the model for every classes prove that the error during the model creation is kept to its minimum and stating the high probability over the correct classification by the model.

5. Conclusion

Many methods in automated and computer vision are prevailing in the market for plant disease classification process, yetthis field of research is behind and needs improvement. In this paper, an approach using the AlexNet neural network architecture is open to discussion in order to classify the plant diseases and differentiate or distinguish the healthy and diseased between the 38 classes. The whole procedure from images collection used in training and validation, the techniques performed over the dataset collected, image pre-processing and augmentation, and the training procedure carried out involving the methods to generate the best-fit model having the best performance is discussed. The dataset is acquired from Kaggle repository which holds over 87,000 images. The model generated over this produced achieved the precision of 88% to 98% with few classes producing 100% precision, for separate class tests. The achieved accuracy of our generated model is 94%. The fine-tuning can be expanded with the increased range of the considered hyper-parameters and the model can be checked for improvement on the performance on this generated model. The enhancement over this will be to fine-tune and trying to improve the model value and moving the web application to an mobile application where the server side holds the model trained and the user can use their hand held devices to capture the images using the camera and displaying the recognized disease. This application will be helpful to the farmers without the consideration of their experience, faster and efficient recognition and making decisions over the time, quantity and type of pesticides to use to control and eradicate the disease. Further it can be extended to use by detecting these diseases over the captured images using drones which covers larger areas. This helps in sustainable development and increases the crop quality for future generations.

References:

1. Sankaran, S.; Mishra, A.; Ehsani, R.; Davis, C. A review of advanced techniques for detecting plant diseases. Comput. Electron. Agric. 2010, 72, 1–13.
2. J. G. ArnalBarbedo, “Digital image processing techniques for detecting, quantifying and classifying plant diseases,” SpringerPlus, vol. 2, article 660, pp. 1–12, 2013
3. Saleem, M.H.; Potgieter, J.; Arif, K.M. Plant disease detection and classification by deep learning. Plants 2019, 8, 468.
4. Chen, J.; Liu, Q.; Gao, L. Visual Tea Leaf Disease Recognition Using a Convolutional Neural Network Model. Symmetry 2019, 11, 343.
5. Saleem, M.H.; Potgieter, J.; Arif, K.M. Plant Disease Classification: A Comparative Evaluation of Convolutional Neural Networks and Deep Learning Optimizers. Plants 2020, 9, 1319.
6. K.Padmavathi.;K.Thangadurai.; Implementation of RGB and Grayscale Images in Plant Disease Detection-ComparitiveStudy. IJST – 2016, 9, 6, 1-6.
7. Alex Krizhevsky, IlyaSutskever, Geoffrey E. Hinton.; Imagenet Classification with Deep Neural Networks. NeurIPS 2012
8. The repository of the collected data.; https://www.kaggle.com/vipoooool/new-plant-diseases-dataset
9. Fatma Marzougui.; Mohamed Elleuch.; Monji Kherallah.; Evaluation of data Augmentation for Detection Plant Disease. DOI: 10.1007/978-3-030-73050-5_47, 2021.
10. https://www.analyticsvidhya.com/blog/2020/08/image-augmentation-on-the-fly-using-keras-imagedatagenerator/
11. D. M. Hawkins, “The problem of over-fitting,” Journal of Chemical Information and Computer Sciences, vol. 44, no.1, pp. 1–12, 2004.
12. A. M. Reddy, V. V. Krishna, L. Sumalatha and S. K. Niranjan, "Facial recognition based on straight angle fuzzy texture unit matrix," 2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC), Chirala, 2017, pp. 366-372, doi: 10.1109/ICBDACI.2017.8070865. (
13. A. M. Reddy, K. SubbaReddy and V. V. Krishna, "Classification of child and adulthood using GLCM based on diagonal LBP," 2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Davangere, 2015, pp. 857-861, doi: 10.1109/ICATCCT.2015.7457003.

14. G. E. Dahl, T. N. Sainath, and G. E. Hinton, “Improving deep neural networks for LVCSR using rectified linear units and dropout,” in Proceedings of the 38th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’13), pp. 8609–8613, Vancouver, Canada, May 2013.
15. C. Ciresan Dan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, “Flexible, high performance convolutional neural networks for image classification,” in Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '11), vol. 22, no. 1, pp. 1237–1242, 2011.
16. S. Han, J. Pool, J. Tran, and W. J. Dally, “Learning both weights and connections for efficient neural network,” in Proceedings of the Advances in Neural Information Processing Systems (NIPS '05), NIPS Proceedings, pp. 1135–1143, 2015.
17. Swarajya Lakshmi V Papineni, SnigdhaYarlagadda, HaritaAkkineni, A. Mallikarjuna Reddy. Big Data Analytics Applying the Fusion Approach of Multicriteria Decision Making with Deep Learning Algorithms *International Journal of Engineering Trends and Technology*, 69(1), 24-28, doi: 10.14445/22315381/IJETT-V69I1P204.
18. Srinivasa Reddy, K., Suneela, B., Inthiyaz, S.,Kumar, G.N.S., Mallikarjuna Reddy, A.” Texture filtration module under stabilization via random forest optimization methodology “International Journal of Advanced Trends in Computer Science and Engineering, Volume 8, No.3, May - June 2019
19. G. E. Dahl, T. N. Sainath, and G. E. Hinton, “Improving deep neural networks for LVCSR using rectified linear units and dropout,” in Proceedings of the 38th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '13), pp. 8609–8613, Vancouver, Canada, May 2013.
20. Andrew Brock, Theodore Lim, & J.M. Ritchie.; FreezeOut: Accelerate Training by Progressively Freezing Layers. arXiv:1706.04983v2 [stat.ML] 18 Jun 2017.
21. https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/