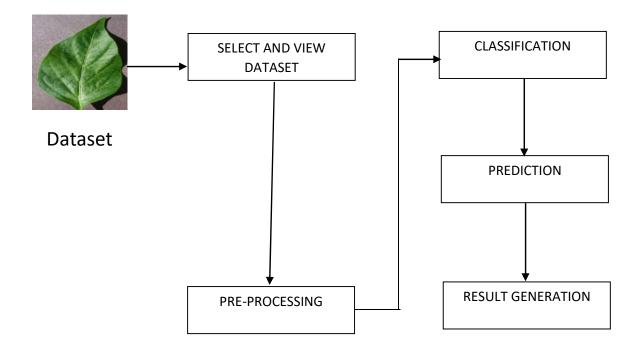
Plant Disease Identification using Densenet Architecture Dr.K.Kavitha¹, A.Prasanna Venkatesh², R.K.Naveen Prasad³, J.Praveen⁴ ¹Professor,²³⁴UG students Department of Electronics and Communication Engineering Velammal College of Engineering and Technology(Autonomous), Madurai-625009

Abstract:

Last year during monsoon season, plants in our house were affected regularly and we were not able to identify what kind of disease it was. So, we decided to develop a project to overcome this difficulty. The main aim of this project is to identify the type of disease affected in the plant using the Plant Village Dataset, The objective of this project is to find out what kind of plant it is and to check whether the plant is healthy or not healthy. If it is unhealthy then it will detect what kind of disease the plant is affected. The software used is Anaconda-Spyder and DenseNet architecture is used to achieve higher accuracy for classifying different plant images. The images are pre-processed and it gets classified and the outcome will be whether the leaf is infected or not. If it is infected it classifies what kind of disease the plant is affected. This project can be further improvised by reducing the run time, and also the accuracy can be further improved.



1.Introduction:

Plant diseases are a major threat to food security, but their rapid identification remains difficult in many parts of the world due to the lack of the necessary infrastructure. Due to the Advancement in computers it has been made possible by deep learning using various architectures. Using a public dataset of 39,443 images of diseased and healthy plant leaves collected under controlled conditions, we can train a deep convolutional neural network to identify 9 crop species and 24 diseases.

Leaves being the most sensitive part of plants show disease symptoms at the earliest. The crops need to be monitored against diseases from the very first stage of their life-cycle to the time they are ready to be harvested. Initially, the method used to monitor the plants from diseases was the traditional naked eye observation that is a time-consuming technique which requires experts to manually monitor the crop fields. In the recent years, a number of techniques have been applied to develop automatic and semi-automatic plant disease detection systems and automatic detection of the diseases by just seeing the symptoms on the plant leaves makes it easier as well as cheaper. These systems have so far resulted to be fast, inexpensive and more accurate than the traditional method of manual observation by farmers In most of the cases disease symptoms are seen on the leaves, stem and fruit. The plant leaf for the detection of disease is considered which shows the disease symptoms. There are many cases where farmers do not have a fully compact knowledge about the crops and the disease that can get affected to the crops.

S.No	Class	Disease	No.of normal	No.of unhealthy		
			images	images		
1	Apple	3	1645	3000		
2	Cherry	1	1000	1052		
3	Corn	3	1162	3192		
4	Grape	3	1000	3639		
5	Peach	1	1000	2297		
6	Pepper Bell	1	1478	997		
7	Potato	2	1000	2000		
8	Strawberry	1	1000	1109		
9	Tomato	9	1500	11372		

Considering our South Indian culture we have taken Potato ,Tomato and Pepper Bell in our project as these plants are suitable for growing in our climatic conditions.

2.Existing Methodology:

In existing system propose a Long Short Term Memory neural network algorithm to accomplish the leaf disease classification task. Plant disease recognition is an interesting and practical topic. However, this problem has not been sufficiently explored due to the lack of systematically investigation and large-scale dataset. The most challenging step in constructing such a dataset is providing a reasonable structure from both the agriculture and image processing perspective.

2.1. Disadvantages:

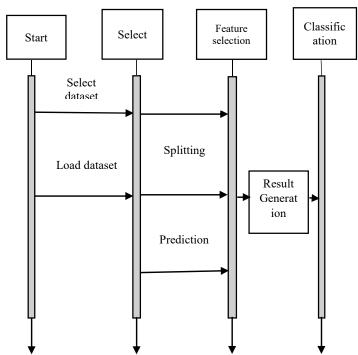
• It convert the images as a data frame and predict using data mining technique.

- LSTMs are prone to over fitting and it is difficult to apply the dropout algorithm to curb this issue.
- Dropout is a regularization method where input and recurrent connections to LSTM units are probabilistically excluded from activation and weight updates while training a network.

3.Proposed Methodology:

The proposed model is introduced to overcome all the disadvantages that arise in the existing system. This system will increase the accuracy of the neural network results by classifying the leaf disease digital image dataset using Deep learning algorithm. It enhances the performance of the overall classification results. Predict the leaf disease image is to find the accuracy more reliable.In deep learning algorithm we are using DENSENET121 which improves the screening accuracy using digital images and also uses a less time duration to identify the disease from the leaf images.

3.1 Sequence Diagram:



3.2 Modules Description:

DATA SELECTION AND LOADING:

- The data selection is the process of selecting the data for PLANT VILLAGE dataset.
- In this project, LEAF digital images are used to classify the disease.
- The dataset which contains the information about the plant leaf disease digital images.

DATA PREPROCESSING:

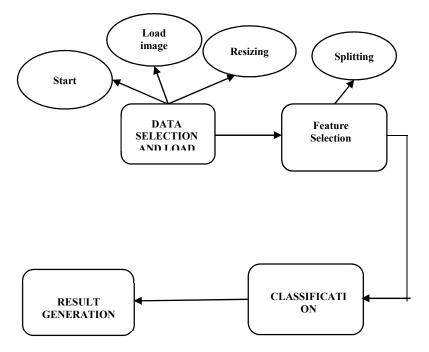
- Image Data pre-processing is the process of getting rescale data from the dataset.
- Resize image dataset: Rescale the leaf digital images size into 64x64.
- Getting data: That categorical data is defined as variables with a finite set of rescaled values. That most deep learning algorithms require array input and output variables.

SPLITTING DATASET INTO TRAIN AND TEST DATA:

- Data splitting is the act of partitioning available data into two portions, usually for cross-validate purposes.
- One Portion of the data is used to develop a predictive model and the other to evaluate the model's performance.

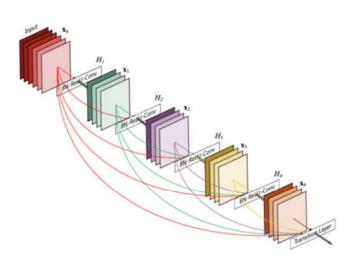
- Separating image data into training and testing sets is an important part of evaluating image processing models.
- Typically, when you separate a data set into a training set and testing set, most of the image data is used for training, and a smaller portion of the data is used for testing

Entity Relationship Diagram:



3.4.DENSENET121:

DenseNet121 (Dense Convolutional Network) is an architecture that focuses on making the deep learning networks go even deeper, but at the same time making them more efficient to train, by using shorter connections between the layers. DenseNet121 is a convolutional neural network where each layer is connected to all other layers that are deeper in the



DenseNet Structure

$$a^{[l]} = g\left([a^{[0]}, a^{[1]}, a^{[2]}, \dots, \dots, a^{[l-1]}\right)$$

network, that is, the first layer is connected to the 2nd, 3rd, 4th and so on, the second layer is connected to the 3rd, 4th, 5th and so on. This is done to enable maximum information flow between the layers of the network. To preserve the feed-forward nature, each layer obtains inputs from all the previous layers and passes on its own feature maps to all the layers which will come after it.

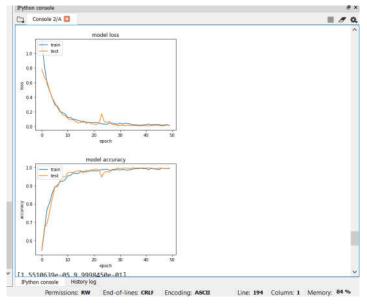
4.Experimental Results:

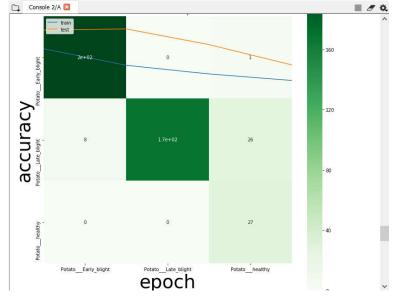
The Experiment is carried out using Plant village dataset and following images are from the output using densenet121 algorithm and their accuracy is classified in a tabular column and it predicts the leaf disease from the dataset and the final result will be generated based on

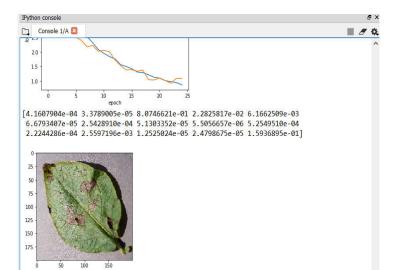
- *1* 0

the overall classification and prediction and the final accuracy will be displayed.

4.1 Output for Potato:







Prediction: Potato

Early blight

Epoch 00046: val_loss did not improve from 0.00628 Epoch 47/50
30/30 [=======] - 84s 3s/step - loss: 0.0160 - accuracy: 0.9958 - val_loss: 0.0070 - val_accuracy: 0.9960
Epoch 00047: val_loss did not improve from 0.00628
Epoch 48/50 30/30 [] - 84s 3s/step - loss: 0.0169 - accuracy: 0.9943 - val loss: 0.0090 - val accuracy: 0.99399
Epoch 00048: val loss did not improve from 0.00628
Epoch 49/50

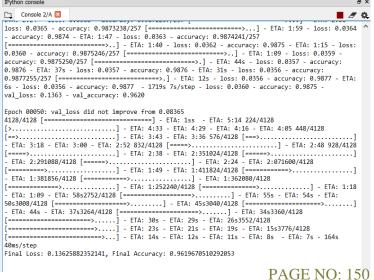
- Epoch 00049: val_loss did not improve from 0.00628
- Epoch 50/50

IPython console

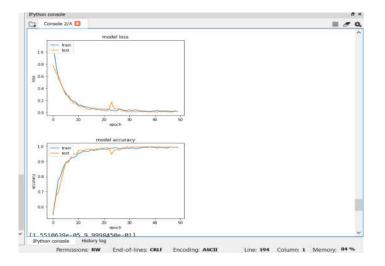
Console 2/A 🗵

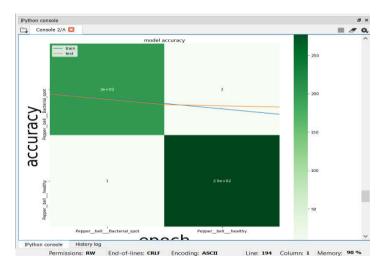
- 0/30 [========] 88s 3s/step loss: 0.0122 accuracy: 0.9963 - val_loss: 0.0125 - val_accuracy: 0.9939
- Epoch 00050: val_loss did not improve from 0.00628

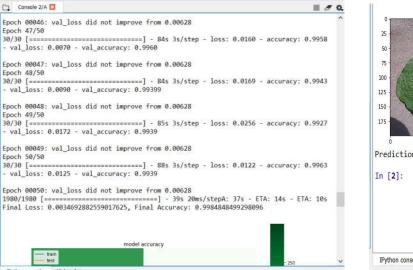




4.2 Output for Pepperbell:





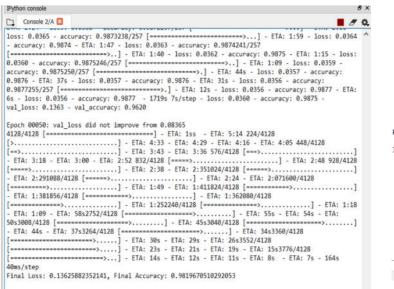


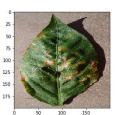
IPython console History log

IPython console

Permissions: RW End-of-lines: CRLF Encoding: ASCII Line: 194 Column: 1 Memory: 92 %







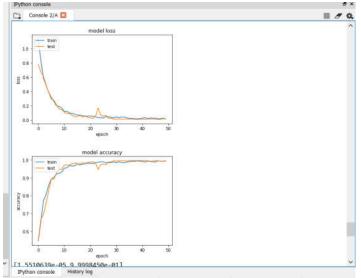
Prediction: Pepper_bell__Bacterial_spot

In [3]:

ð ×

IPython console History log Permissions: RW End-of-lines: CRLF Encoding: ASCII Line: 201 Column: 9 Memory: 93 %

4.3.Output for Tomato:

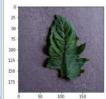


Permissions: RW End-of-lines: CRLF Encoding: ASCII Line: 194 Column: 1 Memory: 84 %

IPython console	8:
Console 2/A 🖸	
loss: 0.0365 - accuracy: 0.9873230/257 [1:15 - loss: 0.0359 - accuracy: acy: 0.9877 - ETA:
Epoch 00050: val_loss did not improve from 0.08365 4128/4128 [] - ETA: 1ss - ETA: 5:14 224/4128 [] - ETA: 4:39 - ETA: 4:29 - ETA: 4:16 - ETA: 4:05 448, [] - ETA: 3:43 - ETA: 3:26 576/4128 [,] - ETA: [] - ETA: 2:29 - ETA: 3:20 - ETA: 5:10 22/4128 [,] - ETA: - ETA: 3:18 - ETA: 3:00 - ETA: 2:32 832/4128 [,] - ETA: [] - ETA: 2:29088/4128 [,] - ETA: 1:41824/4128 [,] - ETA: - ETA: 1:381856/4128 [,] - ETA: 5:4080/4128 [,] - ETA: 1:362080/4128 [,] - ETA: 5:4080/4128 [,] - ETA: 5:54 - ETA: 556 - 503:3008/4128 [,] - ETA: 5:453040/4128 [,] - ETA: 555 - ETA: 556 - ETA: 4:00 - ETA: 58:272/4128 [,] - ETA: 45:33040/4128 [,] - ETA: 556 - ETA: 5570:3008/4128 [,] - ETA: 556 - ETA: 5566 - ETA: 556 - ETA: 556 - ETA: 556 - ETA: 556	

Final Loss: 0.13625882352141, Final Accuracy: 0.9819670510292053

console										-	8
.0050HE 3/A 🔛						model a	ccuracy			1 2	
	Ternato_Target_Spot	tion test	0	1	16	2	Ŧ	1	0	$\left i \right $	
	Tomato_Tomato_mesak_virus	к.	48	Ø	0	0	0	0	2	3	
Tomatu	_Tomato_YellowLeaf_Curl_Virus -	e	6	5.9e+02	12	1	0	2	0	0	
•	Tomato_Bacterial_spot ·	¢	0	3	410+02	8	¢	2	0	0	
	Tomato_Early_blight -	u.	.0	3	в	1.1e+02	1	41	0	1	
	Tomato_healthy -		.0	0	1	.0	376+00.	3	0	0	
	Tomato_Late_blight -	2	0	0	υ	7	2	3 Am+02	Ø	1	
	Tomato_Leaf_Mold -	3	0	7	15	10	8	6	11e+02	5	
	Tomato_Septoria_leaf_spot	29	2	3	20	н	5	58	12	3.8e+02	
Tomato_Spider	mites_Two_spatted_spider_mite		0	1	2	1	1	3	0	0	3
		et Spot -	ic while -	ri Vins.	- spot -	Y_DIDH-	testitry -	e blight -	ef Mold -	- toot, foot -	
	Tomato	innole 3/4 D Tumuto_Target_Spat Tumuto_Teneto_mosac_virus Teneto_Teneto_Mosac_virus Teneto_Teneto_HowLeaf_Carl_Virus Teneto_Bacterial_spot Teneto_factorial_picativy Teneto_targeting Teneto_targeting	Cancole 3/A Timata_Target_Spat Timata_Target_Spat Timata_Tomata_Tomata_masaC_virus Timata_Tomata_Tomata_MasaC_virus Timata_Tomata_Statend_spat Tomata_State_Might Timata_Late_Might Timata_Late_Might Timata_Late_Might Timata_Septore_lead_spat Timata_Septore_lead_spat Timata_Septore_lead_spat Timata_Septore_lead_spat	Zanobie 3/A Timuta: Targer, Seet Timuta: Targer, Seet Timuta: Timuta: Instact. Instact. Yeau Timuta: Timuta: Instact. Yeau Timuta: Timuta: Instact. Yeau Timuta: Timuta: Instact. Yeau Timuta: Instac	Cancole 3/A Timuta_Target_Spar Timuta_Tomata_Target_Spar Timuta_Tomata_Tomata_rusac_yeus A 40 Timuta_Tomata_test_Curl_Vous Domata_Tomata_Statest_Corl_Vous Domata_Tomata_Statest_Curl_Vous Domata_Statest_Spart Domata_Late_test_Vest_Vest Domata_Late_test_Vest_Vest_Vest_Vest_Vest_Vest_Ve	Console 3/A Immain_Targat, Spat Immain Immain	Canacole 3/A Image fraget part Image fraget part <t< td=""><td>Console 3/A Image: Target Spate Image: Target Spate</td><td>Console 3/A O Model accuracy Tunuta_Target_Set Image Set <td< td=""><td>Standole 3/A Image: standole 3/A</td><td>Canado 3/A Image: page: pa</td></td<></td></t<>	Console 3/A Image: Target Spate Image: Target Spate	Console 3/A O Model accuracy Tunuta_Target_Set Image Set Image Set <td< td=""><td>Standole 3/A Image: standole 3/A</td><td>Canado 3/A Image: page: pa</td></td<>	Standole 3/A Image: standole 3/A	Canado 3/A Image: page: pa



Prediction: Tomato_Spider_mites_Two_spotted_spider_mite

5.Accuracy Classification:

Train/test ratio	Classes	Classwise classification accuracy (%)	Average classification accuracy
50-50	Pepper bell bacterial spot	98	98.9
	Pepper bell healthy	99.8	
50-50	Potato Early blight	98.5	
	Potato healthy	99	
50-50	Tomato healthy, Tomato Spider mites Two spotted spider mite	97.5	97.5

6.Conclusion:

In this study, the deep learning classifier is analyse the plant leaf disease images. The PLANT VILLAGE data is taken as input data and applied into pre-processing method. In preprocessing method the images are resized and converted into array. Then it processed into feature selection method, in this method the dataset is split into training dataset and testing dataset. After that all the images are resized and convert into array. Finally the classification method is used to analysis the remote sensing scene from images. Deep learning algorithm of DENSENET is implemented and predict the result based on accuracy.

7.Future Enhancement

In feature, the implementation of work is enhance in web application or graphical user interface model. And it is easy to identify the diseased leaf images and provides the quick and better result by using this model.

8.References:

- Z. Li et al., "Non-invasive plant disease diagnostics enabled by smartphonebased fingerprinting of leaf volatiles," Nature Plants, vol. 5, no. 8, pp. 856– 866, Aug. 2019.
- G. Litjens et al., "A survey on deep learning in medical image analysis," Med. Image Anal., vol. 42, pp. 60–88, Dec. 2017.
- W. Min, S. Jiang, L. Liu, Y. Rui, and R. Jain, "A survey on food computing," ACM Comput. Surv., vol. 52, no. 5, pp. 92:1–92:36, 2019.

- E. Moen, D. Bannon, T. Kudo, W. Graf, M. Covert, and D. Van Valen, "Deep learning for cellular image analysis," Nature Methods, vol. 16, no. 12, pp. 1233–1246, 2019.
- Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Proc. Neural Inf. Process. Syst., 2012, pp. 1106–1114.
- Bauer et al., "Combining computer vision and deep learning to enable ultra-scale aerial phenotyping and precision agriculture: A case study of lettuce production," Horticulture Res., vol. 6, no. 1, pp. 1–12, Dec. 2019.
- S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," Comput. Intell. Neurosci., vol. 2016, pp. 1–11, May 2016.
- J. Wang, L. Chen, J. Zhang, Y. Yuan, M. Li, and W. Zeng, "CNN transfer learning for automatic image-based classification of crop disease," in Image and Graphics Technologies and Applications. Beijing, China: Springer, 2018, pp. 319–329.
- K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Comput. Electron. Agricult., vol. 145, pp. 311–318, Feb. 2018.

- G. Wang, Y. Sun, and J. Wang, "Automatic image-based plant disease severity estimation using deep learning," Comput. Intell. Neurosci., vol. 2017, pp. 1–8, Jul. 2017.
- M. RuBwurm and M. Korner, • "Temporal vegetation modelling using long short-term memory networks for identification from crop mediumresolution multi-spectral satellite images," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jul. 2017, pp. 11-19.
- M. Brahimi, M. Arsenovic, S. Laraba, S. Sladojevic, K. Boukhalfa, and A. Moussaoui, "Deep learning for plant diseases: Detection and saliency map visualisation," in Human and Machine Learning. Cham, Switzerland: Springer, 2018, pp. 93–117.