ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 11, November 2023

Plant Disease Detection using Deep Learning Concepts

[1] Pranav Simha R, [2] Raksha K Moorthy, [3] Rajat Gupta, [4] C Gururaj

[1] Department of Electronics and Telecommunication Engineering, BMS College of Engineering, Bengaluru, India
[2] [3] Department of Electronics and Communication Engineering, BMS College of Engineering, Bengaluru, India
[4] Senior Member IEEE, Department of Electronics and Telecommunication Engineering, BMS College of Engineering, Bengaluru, India
Corresponding Author Email: [1] pranavsimha.te19@bmsce.ac.in, [2] Rakshakm.ec19@bmsce.ac.in, [3] Rakshakm.ec19@bmsce.ac.in, [4] gururaj.tce@bmsce.ac.in

Abstract*— The yield and quality of plants are significantly impacted by plant diseases and pests. The effects of plant diseases on plant growth are more pronounced. It is regarded as a natural catastrophe since it also has an impact on and significantly lowers the yield. The identification of plant diseases may be done via digital image processing. Our project's major emphasis is image processing of plants, namely determining if they are healthy or sick and whether they are infected with germs and viruses. The classification network analyses the input picture after receiving the test image and outputs a label designating the image's categorization. Three models have been used in our study. Three models have been used in our study. Model-1 uses a CNN architecture, Model-2 will use VGG19, and Model 3 will use ResNet50. Consequently, a comparison of three models will be done.

Keywords- CNN, VGG19, ResNet50, Accuracy, Plant diseasedetection , precision, F1score, Confusion matrix.

I. INTRODUCTION

Deep learning principles and several architectures, including CNN, ResNet50, and VGG19, are used to create the model. In this case, supervised learning is being used, and the dataset is tagged. Consequently, models are also utilized to compare the results for the provided input data/test image. As a result, the model can determine if the provided data is healthy or unhealthy.

The project's primary goal is to forecast and identify if the sample dataset of plant leaves contains disease or not, as well as whether it is healthy or ill. As a result, we used the CNN method and Python programming to build our model 1 design.

Therefore, model 2 will be created using VGG19, and model 3 will be created using ResNet50 in order to conduct comparative analyses and choose the most effective model for categorization and result prediction given the input data.

II. LITERATURE SURVEY

The research article "Identification of Plant-Leaf Diseases Using CNN and Transfer-Learning Approach" was the source for the project. The essay provided a quick overview of the approaches we should use to successfully complete the project, including the CNN Approach and Transfer-Learning Approach.

The project was picked from the blog that "Towards Data Science" publishes. This article made it easier to comprehend how to use Kaggle to pick the right dataset.

The academic article titled "Plant Disease Detection using Convolutional Neural Network" was the source for the project. Pest infections affect the country's agricultural production and identifying and detecting plant diseases may be quite helpful to farmers. Because of this major issue in the sector of agriculture, we were motivated to develop a project that would help that industry.

"Detection of plant diseases and pests using deep learning" is the title of a research article. This study aided in the development of our model's approach. We were able to clearly see our models thanks to the methodology's definition and extensive explanation. The project has been chosen from the research paper named “Fine tuning models for plant disease detection”. This article helped us in the Fine tuning of our project

The paper "Leaf Disease Detection Using Deep Convolutional Neural Networks Review and New Approach by VGG19" was the source for the project. This paper aided in our comprehension of the VGG 19 pretrained model.

From the article "Understanding the VGG19 Architecture," the project has been chosen. Understanding the architecture of the VGG 19 model was made easier by this article.

The article "Understanding and Coding a ResNet in Keras" was the source for the project. We learned more about the pre-trained ResNet model thanks to this post.

III. PLANT DISEASE DETECTION IMPLEMENTATION

A. Block Diagram Implementation of Plant Disease Detection

Images from the input data are sent to the appropriate model, such as CNN, VGG19, or ResNet50. As a result, we

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 11, November 2023

can determine if the input (test) picture data is healthy or unhealthy by using our categorized output.

The model is developed using Python programming and deep learning methods. Kaggle is therefore utilized to choose the dataset. 14,607 picture datasets make up the dataset. The dataset was split up so that the model could be well-trained and produce reliable results. Training, validation, and testing datasets were created from the dataset. The training dataset makes up 70% of the entire dataset (10,224 photos), the validation dataset 20% (2,922 images), and the test dataset 10% (1,461 images). The picture is 256 × 256 in size. As a result, we employed three channels to classify data. Convolution2D and MaxPooling on Channel 1; Convolution2D and MaxPooling on Channel 2; and Flatten and Dense layer on Channel 3.

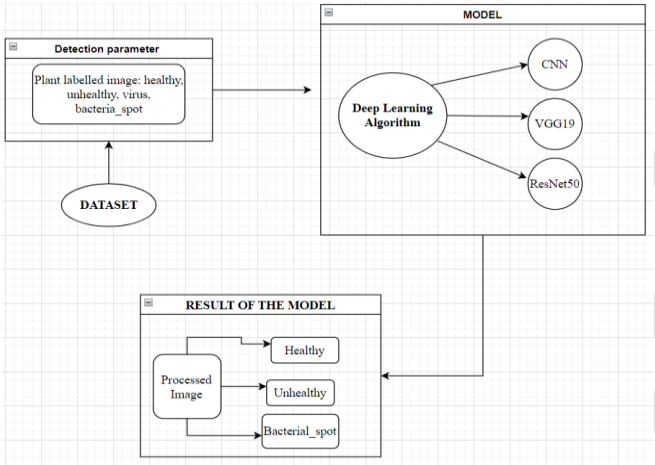


Fig 3.1: Block Diagram of implementation

B. Implementation of Model -1- CNN Architecture

Here, CNN architecture is used to build the model. The CNN architecture is implemented as shown below. The 256 × 256 input dataset is then downsized to 64 x 64. It travels via a separate layer in the following stage. Channel 1: The picture size of 64 × 64 is now convolved and decreased to 62 x 62 after being subjected to 2D and MaxPooling convolution. The picture is still shrunk to 31 by 31 at the MaxPooling step. Conv 2D -1: 29 x 29, and MaxPooling: 14 x 14 on Channel 2. It is flattened and densed in the last stage to provide the desired result.

C. Implementation of Model -2-VGG19 architecture

This model was created using the VGG19 architecture. Below is a picture of the VGG19 architecture in use. The input data picture is 256 × 256 in size. It travels via a separate layer in the following stage. It then goes via MaxPooling and 2D convolution, where the original 256 × 256 picture size is now convolved and shrunk to 128 x 128. The picture is then further reduced to 64 × 64 and subjected to additional convolution and MaxPooling blocks, with the final convolution block's image size being 16 x 16 and the MaxPooling block's size being 8 x 8. Therefore, it travels through Flatten and Dense to provide the desired result.

D. Implementation of Model -3: ResNet50 Architecture

The ResNet50 architecture is used in this model's construction. The ResNet50 architecture is implemented as shown below. The input data picture is 256 × 256 in size. It travels through many ResNet50 layers in the next stage. Convolutional and identity blocks make up ResNet50. The model's core element is the identity block. When a layer's activation is forwarded to deeper layers of neurons, identity block is used. Convolutional and pooling layers make up the convolution block. ResNet50 is a 50-layer deep CNN architecture. Convolution layer is therefore utilized for operations like picture edge improvement as well as image sharpening and blurring. It also provides information on the number of kernels. To reduce the dimensionality of feature maps, pooling layers are utilized. Because of this, it later travels through Flatten and Dense to provide the desired result.

IV. RESULT

This section will describe model assessment, model EDA, and model output predictions. Then, to determine which model is the best and produces accurate results, a comparison analysis of three distinct models will be conducted afterwards. The model therefore includes the f1-score, precision, recall, validation accuracy, and validation loss.

A. Steps Followed for EDA in Plant Disease Detection Project

The data collection, which contained nine different illness classifications and disease categories specific to Apple, was evaluated. The bar plot was used to display the quantity of photos in each class. The class "Frogeye_Resized," which included 3181 photographs, had the most, while "Mosaic_Resized," which contained 371 images, had the fewest. Each class typically had 1623 photographs, which is a lot of pictures.

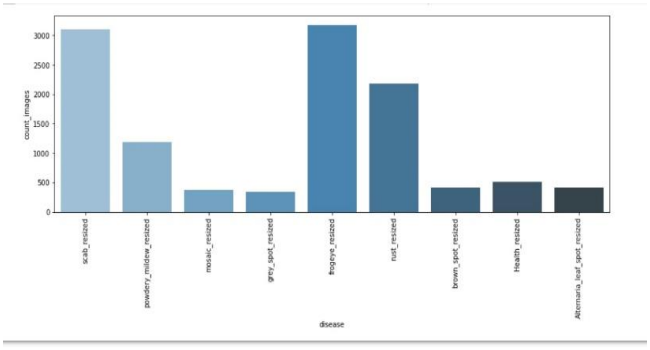


Figure 4.1.1 Bar plot of dataset classes

Each category's photos, which are all from Apple, were noticed.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 11, November 2023

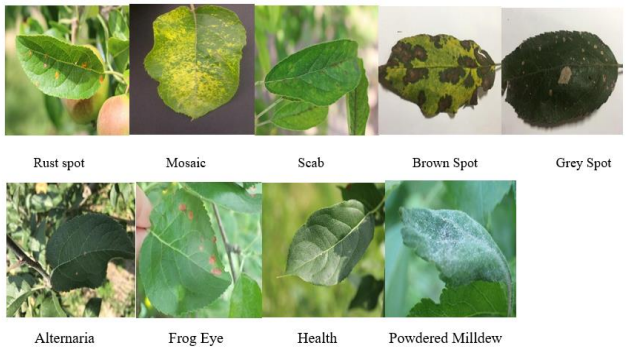


Fig 4.1.2: Apple dataset images

B. Evaluation of Models

1) Evaluation of CNN Model: The CNN architecture is used to implement the model in this case. The 256 × 256 input dataset is then downsized to 64 x 64. It travels via a separate layer in the following stage. Channel 1: The picture size of 64 × 64 is now convolved and decreased to 62 x 62 after being subjected to 2D and MaxPooling convolution. The picture is still shrunk to 31 by 31 at the MaxPooling step. Conv 2D -1: 29 x 29, and MaxPooling: 14 x 14 on Channel 2. It is flattened and densed in the last stage to provide the desired result.

2) Evaluation of VGG19 Model: VGG19 is a variation kind of the VGG architecture, which is a typical deep learning algorithm approach used by CNN. It has 19 deep layers, including 1 soft maximum layer, 3 fully connected layers, 5 max pooling layers, and 16 convolutional layers. Although there are 25 layers total in this case, only 19 of them have learnable parameter layers.

As a result, our model travels through many levels. The input data picture is 256 × 256 in size. It travels via a separate layer in the following stage. It then goes via MaxPooling and 2D convolution, where the original 256 × 256 picture size is now convolved and shrunk to 128 x 128. The picture is then further reduced to 64 × 64 and subjected to additional convolution and MaxPooling blocks, with the final convolution block's image size being 16 x 16 and the MaxPooling block's size being 8 x 8. Therefore, it travels through Flatten and Dense to provide the desired result.

3) Evaluation of RESNET50 Model: Here, 48 convolutional layers, 1 MaxPooling layer, and 1 Average pooling layer are applied to the ResNet50 model. Thus, there are 5 steps in the ResNet50 design. Later, a thick and flat layer form. Each layer receives the picture input in order to forecast the desired result. In stage 1, the raw input dataset of its original size is subjected to convolution, batch normalization, and layer pooling; in stages 2, 3, and 4, convolution and identity blocks of layers are applied; and in stages 5, flatten and dense layers are applied. These layers are used to transform the 2D pooled feature mapped image into a 1D array for the next stage's input, and they take all of the output from the layer before and apply it to neurons to produce a single output.

C. Analysis of Models

1) Analysis of Model -1: The CNN architecture is being used to implement the model -1, and it is now undergoing training and epochs. The model -1 is put into practice, and the model's compilation is finished using the metrics for predictions together with the loss function, optimizer, and model. The callbacks are put into place to avoid overfitting. The model is now fitted, and the epoch is set to 20 with steps per epoch of 1032. The model ends at 98% accuracy in this case.

2) Analysis of Model -2: The model -2 is currently undergoing training and epochs and is implemented using the VGG19 architecture. The model-2 is put into practice, and the model's compilation is completed using the metrics for predictions, the optimizer, and the loss function. The callbacks are put into place to avoid overfitting. The model is now fitted, and the steps per epoch is set to 16 and epoch to 50.

3) Analysis of Model -3: The ResNet50 architecture is being used to implement Model-3, which is now undergoing training and epochs. The model version three is put into practice, and the model's compilation which employs a loss function, an optimizer, and metrics for predictions is completed. The model is now fitted, and the epoch step size is set to 452 with a value of 10.

D. Final Prediction of the Model

1) Outcome of Model – 1: The average f1-score for Apple dataset is around 68%.

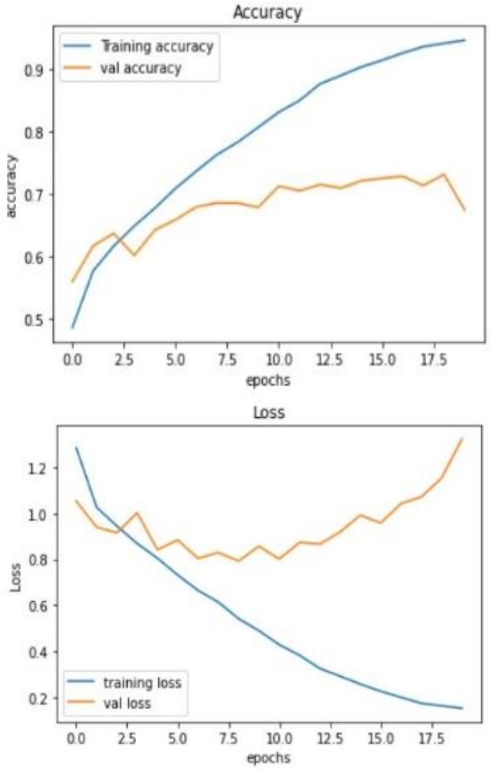


Fig 4.4.1 Plot of accuracy vs val_accuracy and loss vs val_loss for each epoch

ISSN (Online) 2394-2320

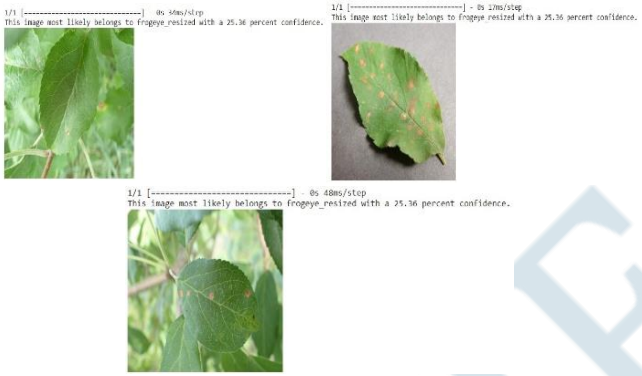
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 11, November 2023

	precision	recall	f1-score	support
Powdery mildew	0.80	0.60	0.69	234
Rust	0.66	0.80	0.72	518
Grey spot	0.57	0.64	0.60	61
Health	0.43	0.40	0.42	105
Scab	0.82	0.62	0.70	1141
Alternaria leaf spot	0.46	0.57	0.51	72
Mosaic	0.65	0.75	0.70	75
Frogeye leaf spot	0.56	0.73	0.63	641
Brown spot	0.95	0.91	0.93	78
accuracy			0.67	2925
macro avg	0.65	0.67	0.65	2925
weighted avg	0.70	0.67	0.68	2925

Fig 4.4.1.2 Metrics calculation of the model

Hence, the model is built and has an accuracy of 68% and predicts proper output for 6 out of 9 classes. Grey_Spot, Alternaria_leaf_spot and Powdery_Mildew.



Hence, from the above output of model -1, the test data given were Apple Dataset. The model predicts correct result for 6 classes out of 9 classes of Apple. For 3 classes mentioned above, the model predicts as Frogeye.

2) Outcome of Model – 2:

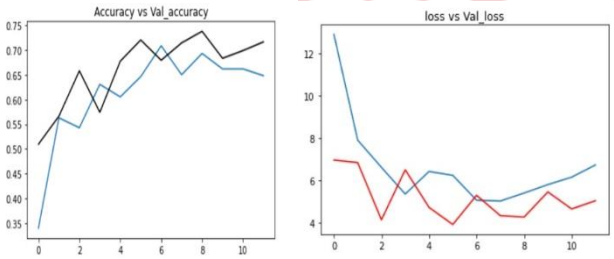


Fig 4.4.2. Plot of accuracy vs val_accuracy and loss vs val_loss for each epoch

```
Training Accuracy    : 0.6484375
Training Loss        : 6.737403869628906
Validation Accuracy  : 0.716796875
Validation Loss      : 5.046938896179199
```

Fig 4.4.2.1 Accuracy and loss results of model – 2

As a result, a model is created, which has a 71% accuracy rate and predicts the right results for the provided test dataset. Therefore, the model accurately forecasts the right outcome for the Apple Dataset.

Rust_Spot Mosaic Scab

As a consequence, from the output of model 2 given above, the test results were correspondingly Rust Spot, Mosaic, and Scab. We can see that model -2 using VGG19 successfully predicts the output for all test data since the model appropriately predicts the outcomes. The model therefore performs better and is more accurate than model -1

3) Outcome of Model – 3:

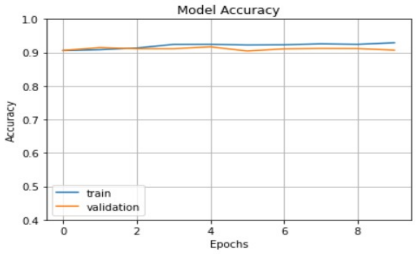
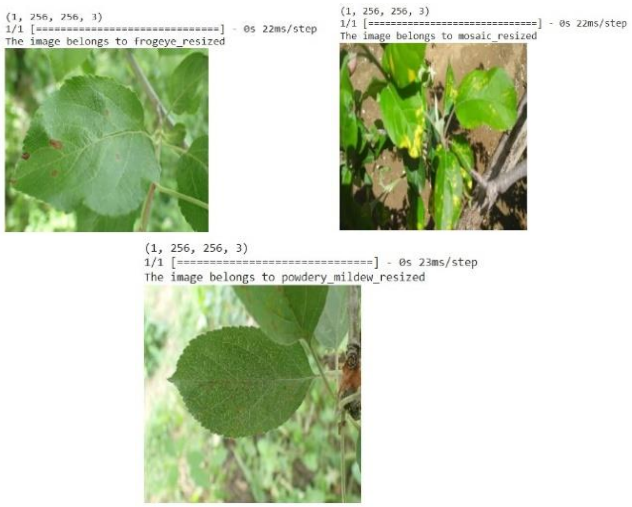


Fig 4.4.2 Plot of accuracy vs val_accuracy for each epoch

```
Training Accuracy ResNet   : 0.929286777973175
Training Loss ResNet       : 0.20518885552883148
Validation Accuracy ResNet : 0.9071856141090393
Validation Loss ResNet     : 0.2993146777153015
```

Fig 4.4.3.1 Accuracy and loss result of model – 3

As a result, a model is created, which has a 91% accuracy rate and predicts the right results for the provided test dataset. As a consequence, the model accurately predicts the right outcomes for all the classes in the Apple Dataset.



As a result, the test results were Frogeye, Mosaic and Powdery_Mildew as shown by the above output of model -2. The model predicts accurate outcomes, hence it is clear that

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 11, November 2023

model -3 utilizing RESONATE50 predicts accurate results for all test data. Therefore, in comparison to both model-1 and model-2, the model is effective and precise.

E. Comparative Study of Three Models

This section examines the three models' comparative analysis. The models utilised are ResNet50, VGG19, and CNN.

We can analyze the most effective and reliable model for detecting plant diseases through comparison studies.

The table analyses each model's attributes in order to assess performance and identify the top model for making predictions with accuracy.

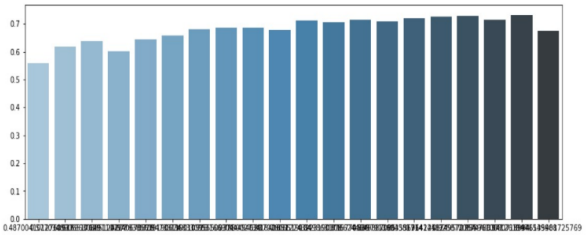


Fig 4.5.1: plot of accuracy vs val_accuracy for model – 1 (CNN)

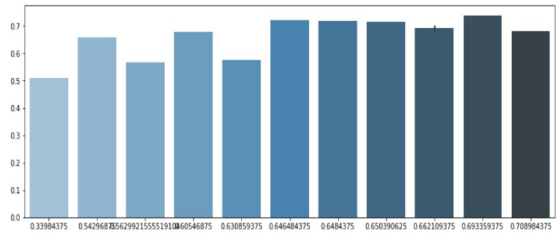


Fig 4.5.2: plot of accuracy vs val_accuracy for model – 2 (VGG19)

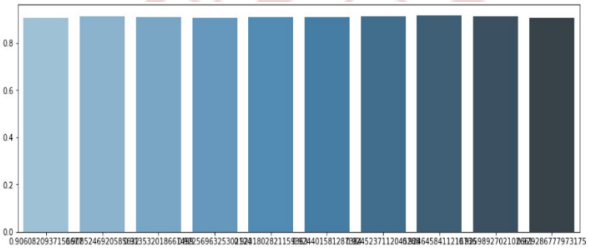


Fig 4.5.3: plot of accuracy vs val_accuracy for model – 3 (ResNet50)

SLNo	Performance Characteristics	Model 1 - CNN	Model 2 – VGG19	Model 3 – RESNET50
1	Image Size	256×256	256×256	256×256
2	Final image size after the end of layer	14×14 (at final max pooling layer)	8×8 (at final max pooling layer)	256×256 (at ResNet50 Functional layer)
3	Number of Kernels	16	3 (at initial input layer) and 512 (at final layer)	3
4	Number of layers in the model	Convolution, MaxPooling, Flatten and Dense layer	19 Deep Layers: 16 Convolutional layers, 3 Hidden layers, 5 MaxPooling layers and 1 SoftMax layer	48 Convolutional layers, 1 MaxPooling layer and 1 Average Pooling layer
5	Overall Accuracy	98.08%	64.47%	92.92%
6	Validation Accuracy	68.48%	70.41%	90.71%
7	Overall Loss	19.45%	6.73%	20.51%
8	Validation Loss	80.09%	5.74%	29.29%
9	Epochs	20/20	12/50	10/10
10	Steps per Epoch	293	16	256

Table 4.5.4: Comparative Study

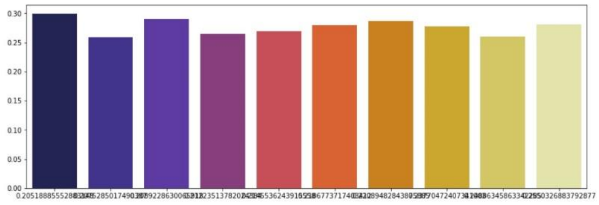


Fig 4.5.4: plot of loss vs val_loss for model – 1 (CNN)

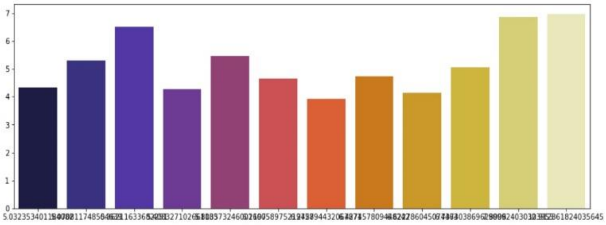


Fig 4.5.5: plot of loss vs val_loss for model – 2 (VGG19)

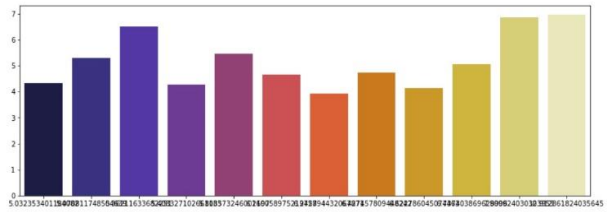


Fig 4.5.6: plot of loss vs val_loss for model – 3 (ResNet50)

V. CONCLUSION AND FUTURE ENHANCEMENT

As a consequence, all three models have been effectively used, and the outcomes are reliable. For each model, the output predictions are also made. Model -1's overall accuracy is 68%, Model -2's overall accuracy is 71%, and Model -3's overall accuracy is 91%. As a result, the model 3 that employs the ResNet50 architecture is the best and most accurate, according to observations and analyses of all the distinctive performance of all the models. With a 91% accuracy rate, this model predicts the output accurately and effectively given the test input data.

The model may be improved in the future by creating a complete model that is beneficial for farmers and other agricultural operations. Additionally, we are developing our own website and app so that farmers can simply take a photo, upload it, and receive information whenever they need to know about a plant's condition. Another feature that may be added is the ability to upload new dataset categories, allowing farmers to get the results through an app or graphical user interface (GUI) for any plant they require for disease prediction.

REFERENCES

[1] Sk Mahmudul Hassan, Arnab Kumar Maji , Michał Jasinski , Zbigniew Leonowicz and Elzbieta Jasinska , Identification of Plant-Leaf Diseases Using CNN and Transfer-Learning Approach ,MDPI ,2021 https://www.mdpi.com/2079-9292/10/12/1388/pdf

[2] https://towardsdatascience.com/plant-ai-plant-disease-detection-using-convolutional-neural-network-9b58a96f2289

[3] https://www.researchgate.net/publication/349162145_PLANT_DISEASE_DETECTION_USING_CONVOLUTIONAL_NEURAL_NETWORK
[4] https://www.kaggle.com/emmarex/plant-disease-detection-using-keras
[5] https://www.kaggle.com/code/deepmalviya7/plant-disease-detection-using-cnn-with-96-84
[6] https://www.frontiersin.org/articles/10.3389/fpls.2016.01419/full
[7] https://plantmethods.biomedcentral.com/articles/10.1186/s13007-021-00722-9
[8] https://www.tensorflow.org/hub/tutorials/cropnet_on_device
[9] https://ijcrt.org/papers/IJCRTH020014.pdf
[10] https://iq.opengenus.org/vgg19-architecture/
[11] https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d3
[12] S. S, R. B N, A. B R, P. S. R and C. Gururaj, "Deep Learning Based Plant Disease Detection," 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), Mysuru, India, 2022, pp. 1-6, doi: 10.1109/MysuruCon55714.2022.997239.