ASSESSMENT OF DATA VISUALIZATION AND PRE-PROCESSING TECHNIQUES ON THE PERFORMANCE OF DEEP NEURAL NETWORKS

Problem Statement

The project on "Assessment of data visualization and pre-processing techniques on the performance of deep neural networks" deals with visualization using different techniques while trying to obtain excellent accuracy and to identify diseased leaves in images taken under real-world conditions.

An attempt to construct a robust model that is focused on obtaining accurate results even when unseen data is passed through it and also visualizing and rectifying the data where the accuracy is low has been made. Visualizing heatmaps, plotting confusion matrix, charts, and graphs, etc. have been discussed in the paper. Also, we have randomly shuffled the data, resized and reshaped the images, and normalized the dataset in hopes to obtain better results in every aspect.

In this paper, we expect to provide a thorough investigation of several visualization methods. These methods are presented in terms of algorithms, and experiment results. Based on these visualization methods, we also discuss their practical applications to demonstrate the areas of network design, optimization, security enhancement, etc.

Review of Literature

The purpose of the literature survey is to identify various studies, models, and papers in our proposed research area in an attempt to appreciate, make use of as well bridge a missing gap, if any, between different research. Many researchers are trying to use data visualization strategies to gain better evaluation metrics or to reduce the network parameters. With the help of these research strategies, the internal behavior of Convolutional Neural Networks is studied.

Table Literature survey

Ref No.	Visualization Technique	Author Name	Year	Conclusion from Strategy
[1]	LIFT-CAM	Hyungsik Jung et al.	2021	●It uses the additive feature attribution method, a novel analytic framework that helps in deciding the weight coefficients. ●It approximates the SHAP values of activation maps precisely with a single backward pass. ●Provides better quantitative and qualitative results than other CAM implementations.
[2]	Integrated GRAD-CAM	Sam Sattarzadeh et al.	2021	●Gradient-based CNN visualization technique, which effectively calculates the features' value. ●This technique applies a path integral to calculate the features' value. ●Results cited it improves GRAD-CAM both in precise localization and while interpreting predictions.
[3]	ADA-SIE	Mahesh Sudhakar et al.	2021	●It is an improvement to the SISE method. ●It gives faster results without compromising the visualization quality.
[4]	Disentangled Masked Backpropagation	Adria Ruiz et al.	2021	●It is a gradient-based approach for attribution map generation. ●DMBP leverages the linear nature of ReLU neural networks to disentangle positive, negative, and nuisance factors from the attribution maps. ●It produces fine-grained attribution maps that give better visual interpretations.
[5]	Ablation-CAM	Saurabh Desai et al.	2020	●It is a class-discriminative localization map for explaining individual decisions of CNN-based models. ●It is a gradient-free technique.

[6]	Axiom-based Grad-CAM	Ruigang Fu et al.	2020	●XGrad-CAM is motivated by the axioms of sensitivity and conservation. ●It enhances Grad-CAM in terms of sensitivity and conservation and significantly improves the visualization performance compared with Grad-CAM.
[7]	Eigen-CAM	Mohammed Bany Muhammad et al.	2020	●It provides a visual explanation irrespective of the accuracy of the model or the presence of adversarial noise. ●It is robust and reliable in producing consistent visual explanations.
[8]	MFPP	Qing Yang et al.	2020	●It explains the prediction of black-box models with multi-scale morphological fragment perturbation modules. ●It can generate a finer-grained interpretation of critically shaped objects on an intuitive basis.
[9]	Zoom-CAM	Xiangwei Shi et al.	2020	●To generate high-quality pseudo labels by integrating visual maps overall intermediate layers in classification. ●It uses weight masks to linearly combine the feature maps at any intermediate CL.
[10]	SmoothTaylor	Gary S. W. Goh et al.	2020	●It was proposed by an integration of integrated gradients and smooth GRAD ●It introduces multi-scaled average total variation as a new measure for the noisiness of saliency maps. ●It can produce attribution maps that are more relevance-sensitive and with much less noise as compared to IG.
[11]	Smoothed Score-CAM	Haofan Wang et al.	2020	●It significantly enhances the localization of the features of the target class in an image. ●It significantly increases the score as the pixels get inserted.
[12]	Score-CAM	Haofan Wang et al.	2020	●It uses Increase in Confidence for the weight of each activation map, removes the dependence on gradients, and has a more reasonable weight representation.
[13]	Backpropagation Saliency Method	Ruth Fong et al.	2020	●It unifies several existing backpropagation-based methods and allowed us to systematically explore the space of possible saliency methods.
[14]	Smooth Grad-CAM++	Daniel Omeiza et al.	2019	●It is an enhanced visual saliency map that can help increase our understanding of the internal workings of trained deep convolutional neural

				network models at the inference stage. ● Its results disclosed improvements in the generated visual maps when compared to existing methods.
[15]	CHIP	Xinrui Cui et al.	2019	● It can provide visual interpretations for the predictions of networks without requiring re-training.
[16]	RISE	Vitali Petsiuk et al.	2018	● It estimates the importance of input image regions for the model's prediction.
[17]	Grad-CAM++	Aditya Chattopadhyay et al.	2018	● Grad-CAM++ method addresses the shortcomings of Grad-CAM - especially multiple occurrences of a class in an image and poor object localization.
[18]	Grad-CAM	Ramprasaath R. Selvaraju et al.	2017	● It is a class-discriminative localization technique—Gradient-weighted Class Activation Mapping (Grad-CAM)—for making any CNN-based models more transparent by producing visual explanations.
[19]	CAM	Bolei Zhou et al.	2016	● It enables classification-trained CNNs to learn to perform object localization, without using any bounding box annotations. ● Class activation maps allows to visualize the predicted class scores on any given image, highlighting the discriminative object parts detected by the CNN

Existing System

The models built using deep learning at times don’t give good results on the unseen dataset, whereas the evaluation metrics for. Example accuracy performs outstandingly well on the training dataset. At such times a need arises to evaluate the neural network what is happening in each layer. Researchers try to visualize the layers, activation to know how exactly a model is learning.

Visualization strategies were also used for the increasing demand for understanding the internal behavior of the convolutional neural network. All this work has been documented as well as implemented, the optimal solution is yet to be discovered. Techniques are being applied to regularizing training and to unravel implicit attention of CNN while working with images.

Researchers implemented techniques that produce visual explanations for decision making and making the CNN model more transparent and explainable. Also, studies related to fine-grained visualizations to create a high-resolution class-discriminative visualization were being implemented. Techniques that can provide better visual explanations of CNN model predictions, in terms of better object localization as well as explaining occurrences of multiple object instances in a single image, when compared to the state-of-the-art.

Methodology

As discussed in the earlier sections, the existing approach is disadvantageous. Therefore, an advanced system is required which overcomes the drawback of the existing approach.
Thus, the implementation of the new proposed system in any given situation will reduce the time and help the user interpret the functionality of CNN through visualization strategies. It will help to increase the evaluation metrics of the model and give better results. It will be capable to reduce the network of the model, deleting those parameters that do not contribute to the give optimal results.

The automated system will consist of some extravagant features such as:

- To train the system for natural complex images and give good accuracy.
- To train the system on multiple models and to find the optimal model that suits a particular domain.
- To use multiple Deep Learning Models for prediction.
- To assess the results of each layer of the CNN as well as produce heatmaps, and functioning in each layer.

5.2.1 Dataset

1. PLANT VILLAGE DATASET

Our project mainly focuses on visualization strategies, tomato plant dataset is used to test various visualization strategies. The dataset is a collection of leaf images of tomato plants that are diseased. In all, there were images of tomato disease of 10 types. Following is the list of classes which we considered for our task:

- bacterial_spot
- leaf_mold
- septoria_leaf_spot
- yellow_leaf_curl_virus
- early_blight
- mosaic_virus
- spider_mites
- leaf_blight
- powdery_mildew
- target_spot

For a training algorithm to be valid we must collect as much data as possible. The more training data that you can give to a network, the more training iterations you can make, the more weight updates you can make, and the better tuned to the network is when it goes to production. This dataset was divided into a training dataset, validation dataset, and testing dataset.

2. INTERNET AUGMENTED

A dataset comprising of internet plant leaf images. These images were then augmented using data visualization strategies. After augmentation, the dataset was enriched. The dataset has the following classes.

- Blight Late – 773
- Early Blight – 613
- Healthy – 642
- Septoria Leaf- 694

3. REAL WORLD AUGMENTED DATASET

A real-world dataset plant leaves samples were collected from dapoli village. This dataset comprises 12 classes. These images were augmented using augmentation strategies. The dataset has the following classes:

- Alt + miner - 535
- BLF - 467
- B-wilt - 126
- E-blight - 586
- Healthy - 350
- Insect - 348
- L blight - 108
- Leaf – miner - 383
- Mosaic - 372
- Physio - 375
- Powdery mildew - 313
- Septoria leaf spot - 1132

5.2.2 Data Augmentation

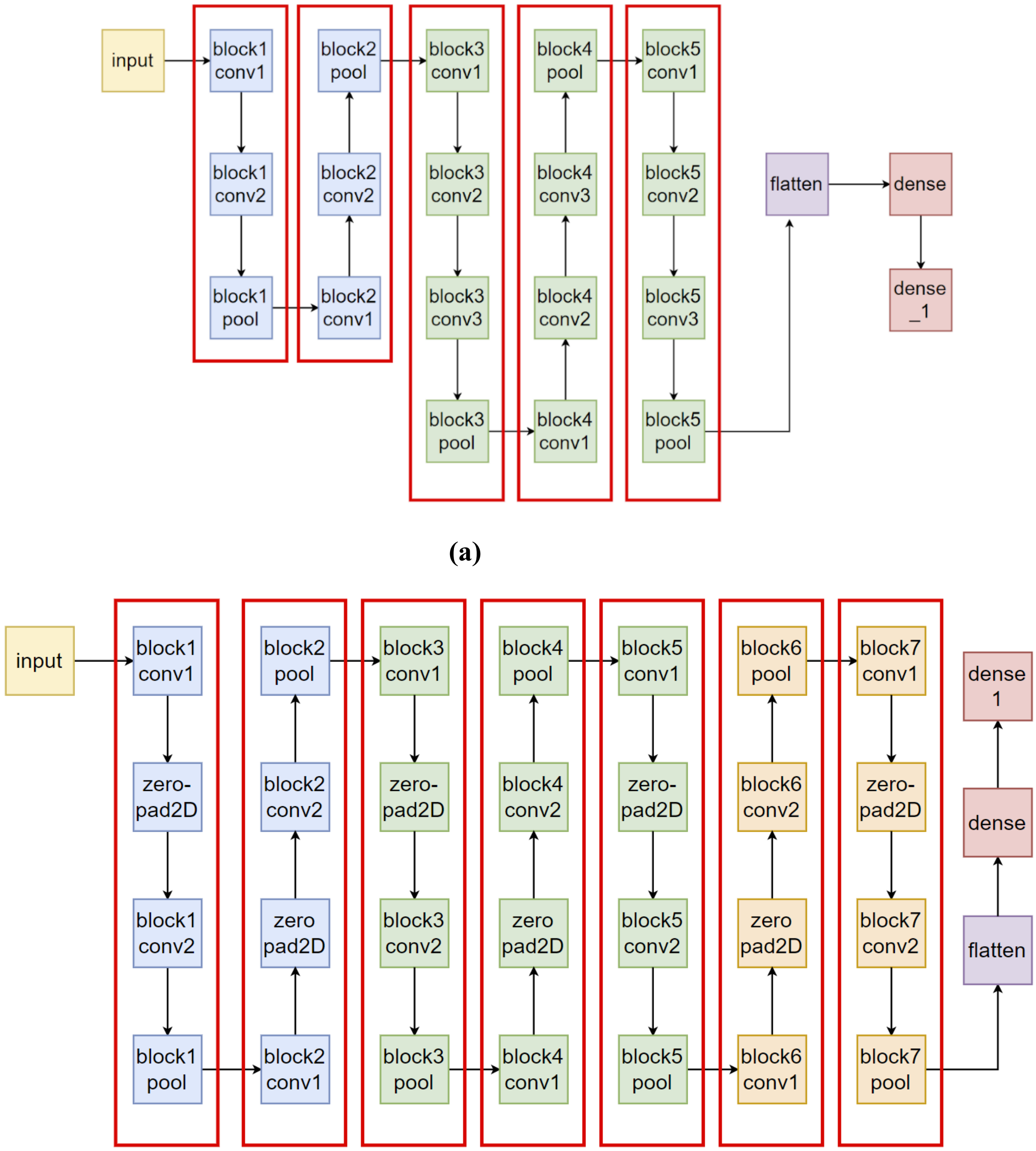
Data Augmentation refers to randomly applying various kinds of transforms to the images in our dataset. These transforms help to introduce varieties in our dataset. Different techniques of data augmentation such as rotation of images at an angle, the random crop of images, brightness/contrast of images, etc have been employed in our data augmentation of captured images.

Fig.5.2 Augmented Leaves

5.2.3 Training of data:

Training is done using CNN's. In machine learning, a convolutional neural network (CNN, or ConvNet) is a class of deep, feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery.

Experiment 1



(b)

Fig.5.3. VGG16 models

The VGG16 architecture was used as the main reference model.

(a): A VGG16 model is used, using transfer learning where the last 3 layers were modified, and pre-trained imagenet weights are used. The model was trained on 264 epochs. There are loopholes in the model architecture on experimental analysis. The model fails to depict the actual behaviour of the neural network. The heat maps generated are of poor quality. A new improvised model was required to increase accuracy and remove redundant layers and parameters from the network.

(b): Constructing the VGG16 architecture using the sequential function and training it on random initialised weights. This model gave better results than the transfer learning model It failed to give results due to the excessive training and failed to evaluate the max values required to generate a heatmap It gave significant results for the upper layers of this architecture. later on, as the pixel size decreased, and the model starts working on single pixel, and a need for better improvised model was required. This model was bulky in nature and failed to depict the network behaviour.

(c): The final proposed model is an improvised version of the intermediate model which required less training. This model achieved better results by eliminating drawbacks of the previous model. In this system padding done throughout in the same block to restore the dimensions of the image and few layers were eliminated. The convolutional layers block3_conv3, block4_conv3, block5_conv3 were removed and two blocks comprising 4 convolutional layers i.e. block6_conv1, block6_conv2, block7_conv1 and, block7_conv2 are added.

Experiment 2

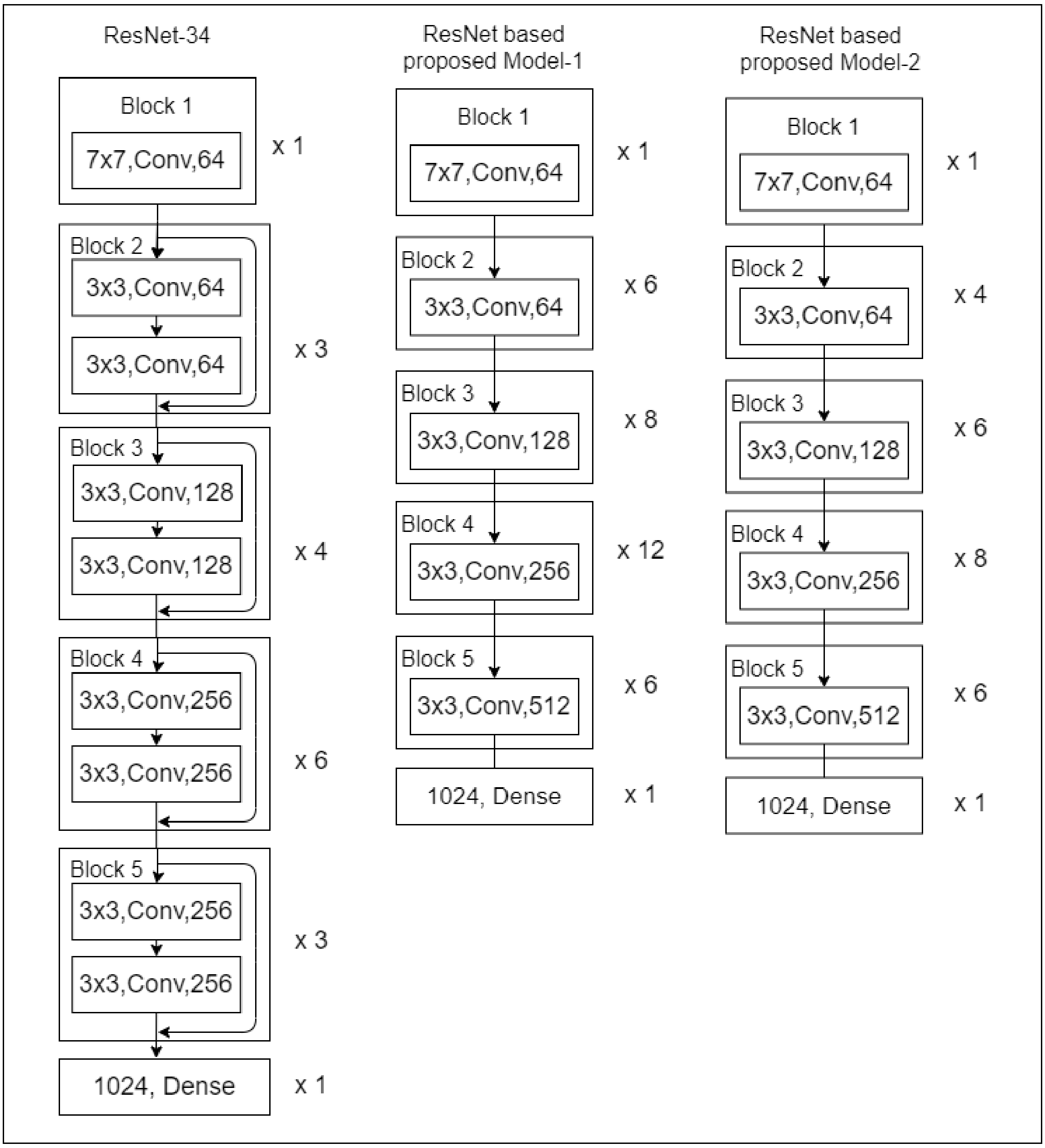


Fig.5.4. Resnet models

The 34-layer ResNet architecture was used as the main reference model.
(a): A custom model was trained and the visualization technique used was Grad-CAM, the output obtained had multiple similar blocks hence eliminating layers would help reduce the architecture size while maintaining and/or improving the accuracy.
(b): Considering, the drawbacks of the custom model an improvised model was made in which the blocks i.e., block2_conv5, block2_conv6, block3_conv7, block3_conv8, and from block4 the last four layers were eliminated as they all had similar outputs.
Hence by eliminating layers an improvement in the accuracy was observed.

5.2.4 Feature Extraction

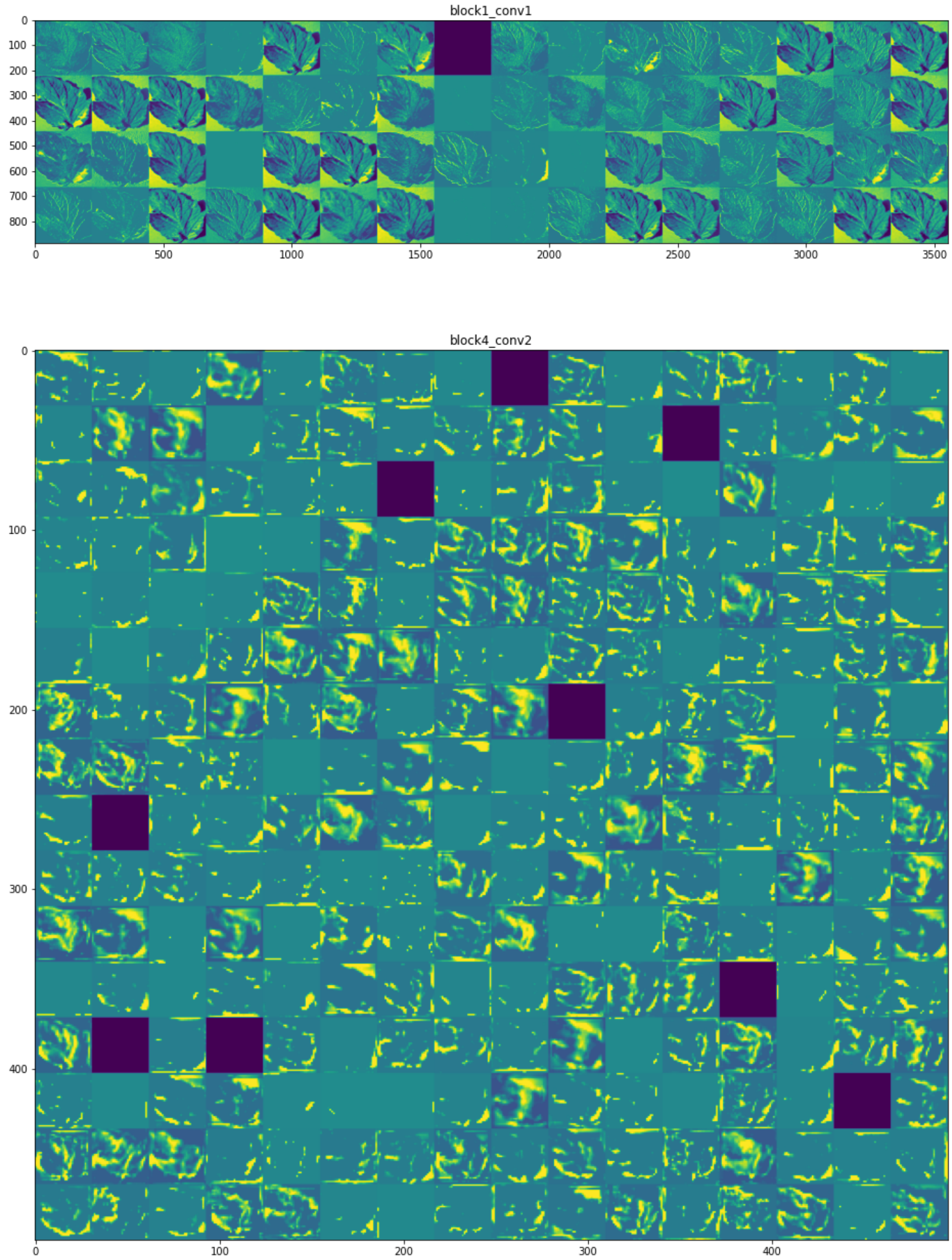


Fig.5.6 Feature Extraction of leaves

Feature extraction helps in describing the relevant shape information contained in a pattern so that the task of classifying the pattern is made easy by a formal procedure. In pattern recognition and image processing, a feature extraction is a special form of dimensionality reduction. Thus, feature extraction is a pivotal step in understanding the image. For feature extraction, we have used the VGG16 model and proposed models giving the best results. We have first divided our dataset into training, validation, and testing with the ratio of 70-30. Next, we have resized all the images in our dataset to the size 224 by 224. These resized images are further passed along the different layers of the VGG16 model and proposed models network layers. VGG16 model and proposed models consist of a combination of convolution layers along with batch normalization and dropout with Relu as the activation function. We have used the loss function as cross-entropy and chosen Adam optimizer. Convolutional neural networks detect visual cues in an image, and it does this by processing a given image with many filters. Figure 5.3 represents the output of the filters and highlights cues that the neural network determines to be important.

5.2.5 Data Visualization Strategies

Various data visualization strategies like Grad-CAM, Grad-CAM++, Score-CAM, and Faster-Score-CAM, generate the heatmaps and to gain insights into how the neural network processes the images and to study the network model.

Experiment 1

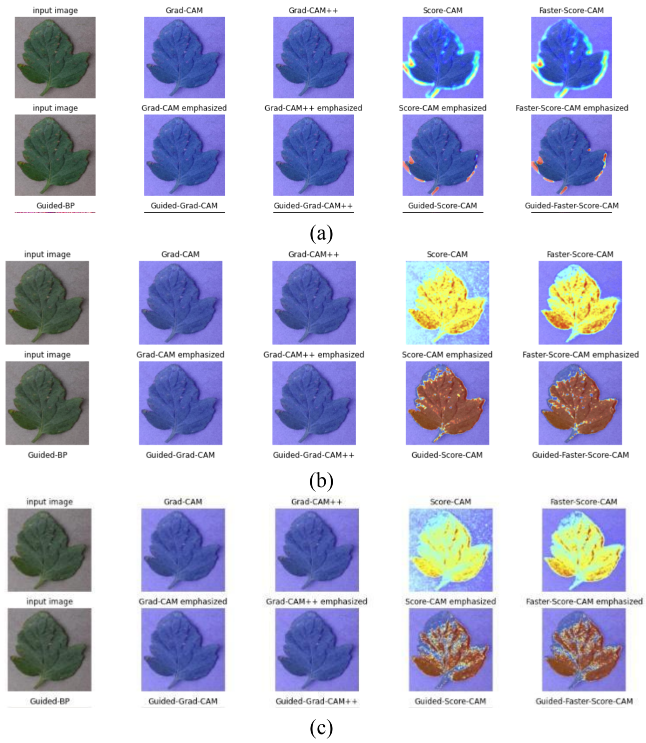
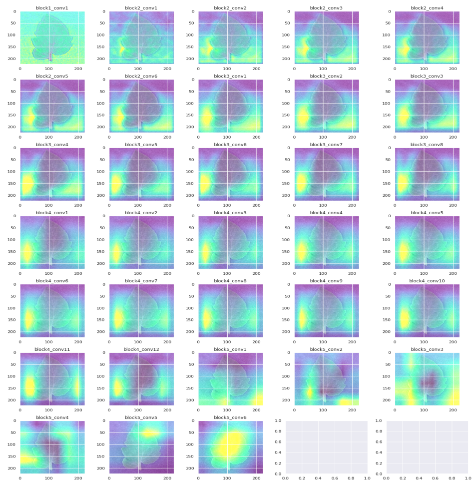


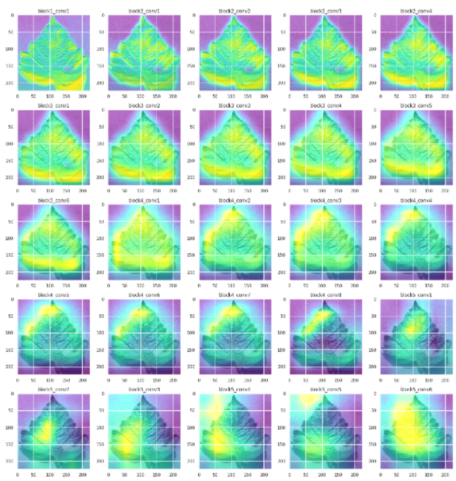
Fig.5.7. Visualizing the first convolution layer (a) model a visualization (b) model b visualization (c) model c visualization

Experiment 2

Various data visualization strategies like Grad-CAM, GradCAM++, Score-CAM, Faster-ScoreCAM, Ablation-CAM, and Occlusion were used to visualize the heatmaps and to gain insights into how the neural network processes the images and to study the network model.



(a)



(b)

Fig.5.8. Model (a) visualisation on plant village dataset, Model (b) visualisation on plant village dataset

From the visualisation we can conclude that similar blocks were obtained and hence the need to eliminate them was seen.

On comparing the visualisation obtained from Fig. Model (a) visualisation and Fig. Model (b) visualisation, it is clearly seen that by eliminating the layers an improvement in the accuracy is seen as well as the architecture size has been reduced.

Results

Trying all network architectures

Plant Village Dataset

Below are the results acquired **without** using pre-trained weights

Table 8.1. Results on all network without using pre-trained weights

Models	loss	accuracy	Validation loss	Validation accuracy
DenseNet121	0.1620	0.9434	4.6237	0.3950
DenseNet169	0.2926	0.8887	7.6172	0.0762
DenseNet201	11.4295	0.2909	9.7511	0.3950
EfficientNetB0	0.3782	0.8750	5.6718	0.0513
EfficientNetB1	0.5367	0.8418	9.5671	0.1058
EfficientNetB2	12.0256	0.2539	14.8647	0.0778
EfficientNetB3	1.8171	0.5412	14.2914	0.1133
EfficientNetB4	12.5584	0.2209	9.9140	0.3849
EfficientNetB5	11.6125	0.2795	9.7511	0.3950
InceptionResnetV2	11.7825	0.2690	9.7511	0.3950
InceptionV3	11.5871	0.2811	9.7511	0.3950
MobileNet	14.2735	0.1144	14.8647	0.0778
MobileNetV2	11.6136	0.2795	9.7511	0.3950
MobileNetV3Large	1.6444	0.6309	2.6605	0.1058
MobileNetV3Small	0.4544	0.8610	2.3283	0.0513
NASNetLarge	14.1018	0.1251	13.7294	0.1482
NASNetMobile	14.0740	0.1268	13.7294	0.1482
Resnet101	11.9728	0.2572	9.7511	0.3950
Resnet101V2	14.7856	0.0827	14.3617	0.1090
Resnet152	12.5064	0.2241	9.7511	0.3950
Resnet152V2	0.4367	0.8516	3.8711	0.1058
Resnet50	14.6220	0.0928	14.8647	0.0778
Resnet50V2	12.1577	0.2457	9.7511	0.3950
VGG16	**0.1936**	**0.9648**	**0.4285**	**0.8904**
VGG19	1.3826	0.2900	1.3802	0.3333
Xception	11.9770	0.2569	9.7511	0.3950

Hence, we can conclude V6616 is the best suited architecture.

Final results on Experiment 1

The proposed algorithm's performance is tested and evaluated on Plant village dataset (Mohanty et al., 2016), internet downloaded image dataset and, our gathered image dataset. All the three datasets are split as train, test and validate in ratio 70:15:10.

(a): The model used pre-trained imagenet weights. The model was trained on 264 epochs. Using visualization strategies like Score-CAM and faster-Score-CAM the loopholes in the model architecture were examined, after gaining insights from the above techniques. The model gives decent accuracy on the test dataset, but it fails to visualize few classes of the dataset also, it fails to depict the actual behaviour of the neural network. The heat maps generated are of poor quality and fail to denote the internal functioning of the model.

Plant village dataset results

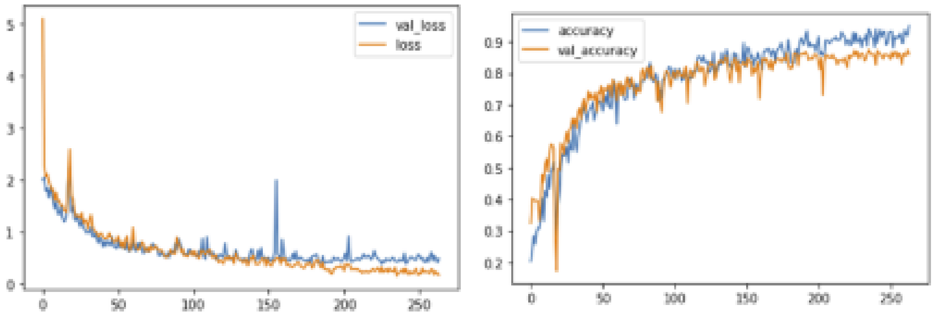
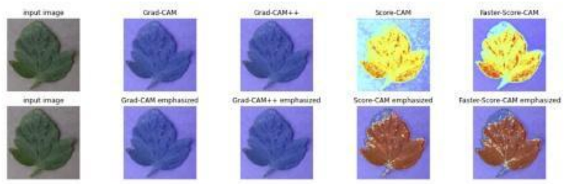


Fig.8.1 model a accuracy and loss curve

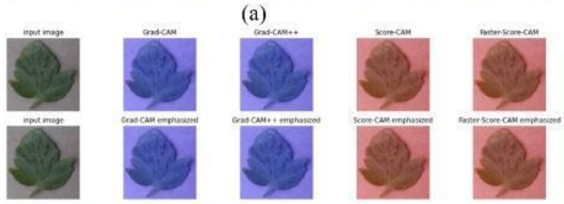
(a)

(b)

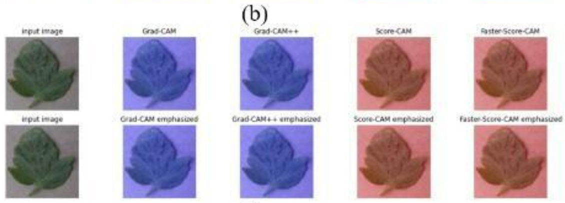
Fig.8.2 model a (a) first layer,(b)last layer



(a)

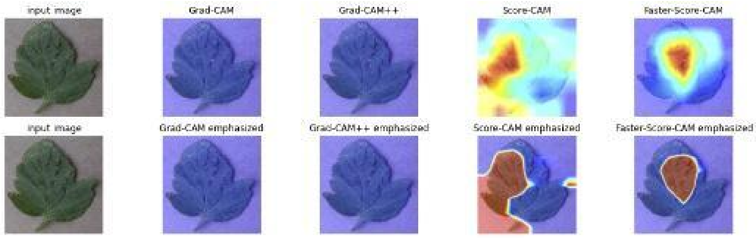


(b)

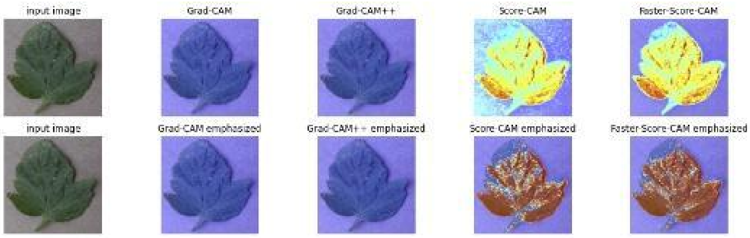


(c)

Fig 8.3. Model b (a)first layer, (b) b4c2 layer, (c) b4c3 layer



(a)



(b)

Fig. 8.4. model c (a) first layer, (b)last layer

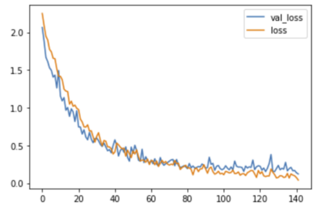
From the visualisation we can conclude Grad-CAM and Grad-CAM++ failed to visualise the activations in the neural network and Score-CAM and Faster-Score-CAM no significant changes were observed in the model.

(b): This model gave better results than the transfer learning model. It gave decent accuracy along with heatmaps. The model was trained on 739 epochs and gave an accuracy of 86%. On applying visualisation strategies Grad-CAM and Grad-CAM ++ failed to give results due to the excessive training and failed to evaluate the max values required to generate a heatmap. Score-CAM and faster-Score-CAM gave significant results for the upper layers of this architecture. From the above diagram we can see there is no change in the subsequent layers and hence 1 layer is removed. This method failed to visualise the last subsequent layers while working on single pixel.

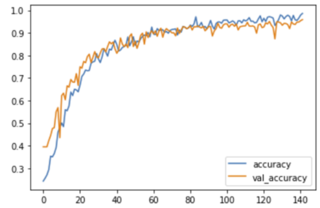
(c): The final proposed model is an improvised version of the intermediate model which required less training, It is trained on 142 epochs. This model gives good validation accuracy and 95% accuracy on unseen dataset. The redundant layers were removed.

```
[[ 51  13  24   7  26  33  30 115   5  29]
 [ 16   8   9   3   4   8  13  37   2   6]
 [ 30  11  11  12  20  17  12  52   4  12]
 [ 21   4   4   5   9   7   9  34   3   4]
 [ 21  13  16  13  25  25  12  79   6  10]
 [ 33  14  21  13  22  21  17  64   1  18]
 [ 22  11  25   9  15  20  21  60   2  13]
 [117  38  56  33  58  83  53 212  14  55]
 [  9   1   2   2   4   3   5  14   0   3]
 [ 17   8  20   8  18   8  20  52   1  17]]
```

(a)



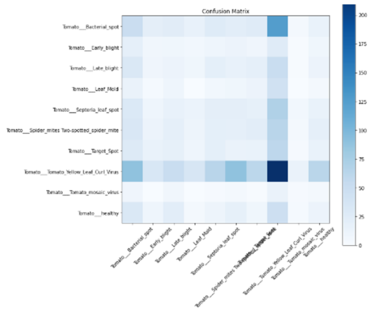
(b)



(c)

```
Classification Report
                                              precision    recall  f1-score   support

                   Tomato___Bacterial_spot       0.15      0.15      0.15       333
                     Tomato___Early_blight       0.07      0.08      0.07       106
                      Tomato___Late_blight       0.06      0.06      0.06       181
                        Tomato___Leaf_Mold       0.05      0.05      0.05       100
               Tomato___Septoria_leaf_spot       0.12      0.11      0.12       220
Tomato___Spider_mites Two-spotted_spider_mite    0.09      0.09      0.09       224
                      Tomato___Target_Spot       0.11      0.11      0.11       198
    Tomato___Tomato_Yellow_Leaf_Curl_Virus       0.29      0.29      0.29       719
              Tomato___Tomato_mosaic_virus       0.00      0.00      0.00        43
                          Tomato___healthy       0.10      0.10      0.10       169

                                  accuracy                           0.16      2293
                                 macro avg       0.10      0.10      0.10      2293
                              weighted avg       0.16      0.16      0.16      2293
```

(d)



(e)

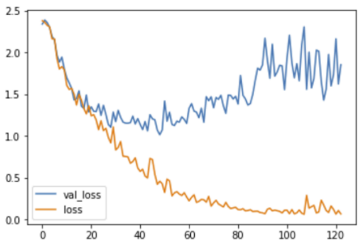
Fig.8.5. model c plant village (a) confusion matrix, (b) loss curve, (c) accuracy curve, (d)classification report, (e) heatmap

Real-world dataset

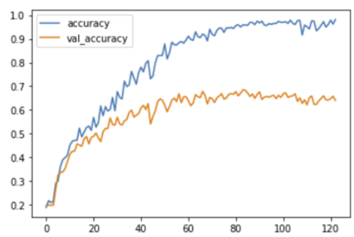
(c): The dataset has 12 classes in all. The model gave 72% accuracy on real-world dataset on experiment 1 model c.

```
[[11  0 11  7  1  5  3  4  7  6  4 20]
 [ 1  0  1  5  1  1  1  0  2  2  3  7]
 [ 5  2  4  5  0  2  3  6  5  3  4 26]
 [10  2  5  6  3  8  2  3  2  4  3 16]
 [ 3  0  3  2  2  2  0  0  1  2  0  3]
 [ 3  0  4  3  2  1  3  3  4  4  3  8]
 [ 5  0  5  2  1  4  5  2  2  1  1  8]
 [ 3  2  4  3  2  1  2  1  2  1  3 17]
 [ 6  0  2  4  1  2  2  4  4  0  5 18]
 [ 5  1  4  6  1  1  3  3  3  5  1 18]
 [ 6  1  1  3  1  0  0  4  5  4  2 10]
 [32  6 13 17  3 15 11 11 13 15 12 72]]
```

(a)



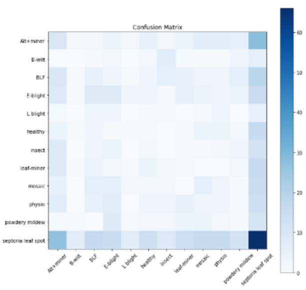
(b)



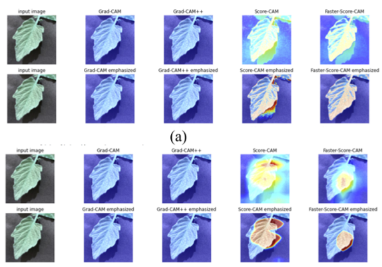
(c)

```
Classification Report
                   precision    recall  f1-score   support

       Alt+miner       0.12      0.14      0.13        79
          B-wilt       0.00      0.00      0.00        24
             BLF       0.07      0.06      0.07        65
        E-blight       0.10      0.09      0.09        64
        L blight       0.11      0.11      0.11        18
         healthy       0.02      0.03      0.03        38
          insect       0.14      0.14      0.14        36
     leaf-miner        0.02      0.02      0.02        41
          mosaic       0.08      0.08      0.08        48
          physio       0.11      0.10      0.10        51
  powdery mildew       0.05      0.05      0.05        37
septoria leaf spot     0.32      0.33      0.33       220

        accuracy                           0.16       721
       macro avg       0.10      0.10      0.10       721
    weighted avg       0.15      0.16      0.16       721
```

(d)



(e)



(f)

Fig.8.6. model c real-world (a) confusion matrix, (b) loss curve, (c) accuracy curve, (d)classification report, (e) heatmap, (f) visualisation of first and last layer

Experiment 2

a) The model was trained on 188 epochs using Grad-CAM. It gave an accuracy of 93% and heatmaps were generated.

Plant village dataset results

```
[[ 53  14  25  10  33  32  27 105   5  29]
 [ 16   5   8   5  16   7  10  27   2  10]
 [ 33   6  14  15  15  15  15  58   1   9]
 [ 21   4  13   3   6   6   6  30   0  11]
 [ 32  13  18   9  17  21  15  73   4  18]
 [ 35   9  14   9  18  27  24  71   2  15]
 [ 20   9  23   4  22  18  14  64   6  18]
 [109  32  57  30  61  93  54 228   9  46]
 [  7   1   5   2   4   5   2  14   0   3]
 [ 28   8  18   8  15  14  13  46   7  12]]
```

```
Classification Report
                                                    precision    recall  f1-score   support

                      Tomato___Bacterial_spot            0.15      0.16      0.15       333
                        Tomato___Early_blight            0.05      0.05      0.05       106
                         Tomato___Late_blight            0.07      0.08      0.07       181
                           Tomato___Leaf_Mold            0.03      0.03      0.03       100
                  Tomato___Septoria_leaf_spot            0.08      0.08      0.08       220
Tomato___Spider_mites Two-spotted_spider_mite            0.11      0.12      0.12       224
                         Tomato___Target_Spot            0.08      0.07      0.07       198
       Tomato___Tomato_Yellow_Leaf_Curl_Virus            0.32      0.32      0.32       719
                 Tomato___Tomato_mosaic_virus            0.00      0.00      0.00        43
                             Tomato___healthy            0.07      0.07      0.07       169

                                     accuracy                                0.16      2293
                                    macro avg            0.10      0.10      0.10      2293
                                 weighted avg            0.16      0.16      0.16      2293
```

Fig.8.7. Confusion matrix and classification report of model (a) on plant village dataset

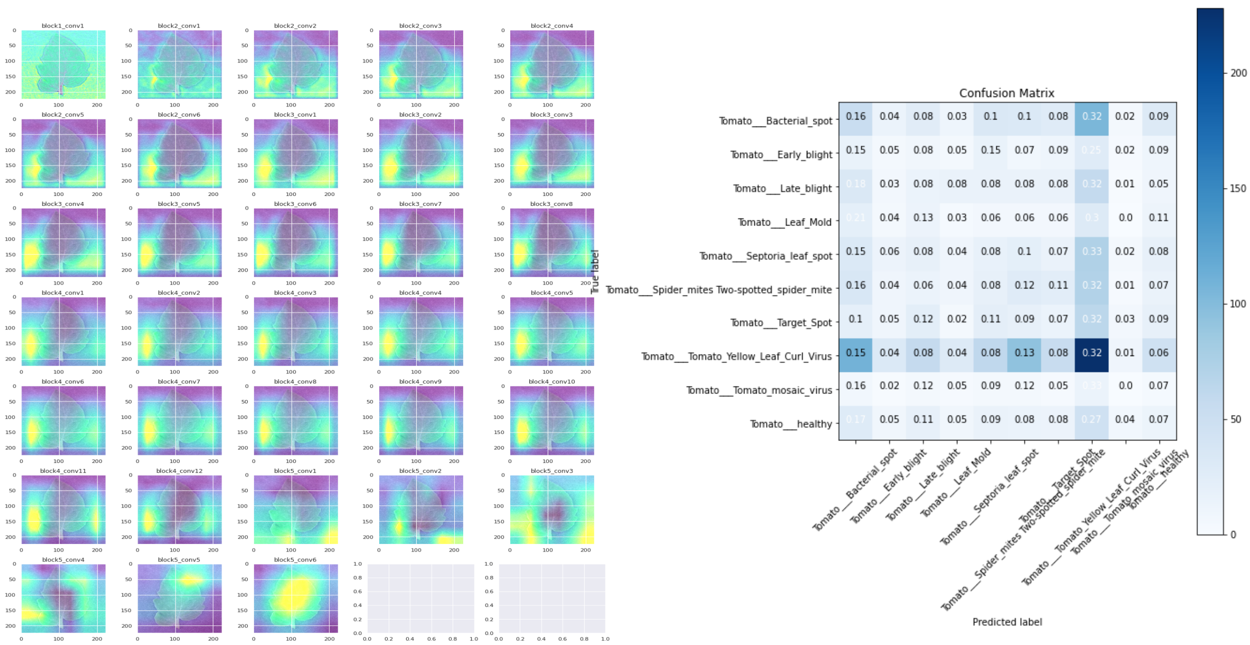


Fig.8.8 Model (a) visualisation on plant village dataset

From the visualisation we can conclude that similar blocks were obtained and hence the need to eliminate them was seen.

b) This improvised model gave better results than the previous model. The model was trained on 41 epochs and gave test accuracy of 94%.

```
[[ 41  13  26  13  30  37  32 101   9  31]
 [ 20   6   9   3   8   6  12  32   1   9]
 [ 25  14  18   4  19  20  13  57   3   8]
 [ 12   7  10   5  10  10  10  28   2   6]
 [ 36  15  14   9  18  15  19  71   8  15]
 [ 34  11  17  10  19  20  23  68   7  15]
 [ 21  16  12   8  16  15  22  66   2  20]
 [105  38  45  36  61  66  62 239  17  50]
 [  8   2   1   0   5   3   4  17   0   3]
 [ 28   7  10  15  21  13  13  53   0   9]]
```

```
Classification Report
                                               precision    recall  f1-score   support

                     Tomato___Bacterial_spot       0.12      0.12      0.12       333
                       Tomato___Early_blight       0.05      0.06      0.05       106
                        Tomato___Late_blight       0.11      0.10      0.10       181
                          Tomato___Leaf_Mold       0.05      0.05      0.05       100
                 Tomato___Septoria_leaf_spot       0.09      0.08      0.08       220
Tomato___Spider_mites Two-spotted_spider_mite       0.10      0.09      0.09       224
                        Tomato___Target_Spot       0.10      0.11      0.11       198
       Tomato___Tomato_Yellow_Leaf_Curl_Virus       0.33      0.33      0.33       719
                 Tomato___Tomato_mosaic_virus       0.00      0.00      0.00        43
                          Tomato___healthy       0.05      0.05      0.05       169

                                    accuracy                          0.16      2293
                                   macro avg       0.10      0.10      0.10      2293
                                weighted avg       0.16      0.16      0.16      2293
```

Fig.8.9. Confusion matrix and classification report of model (b) on plant village dataset

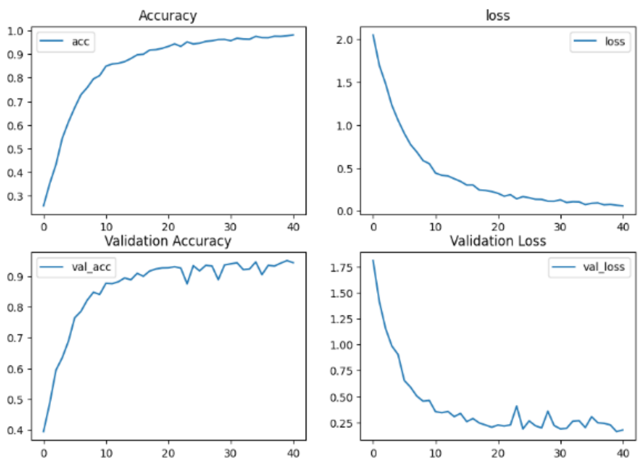


Fig.8.10. Model (b) accuracy curve and loss curve on plant village dataset

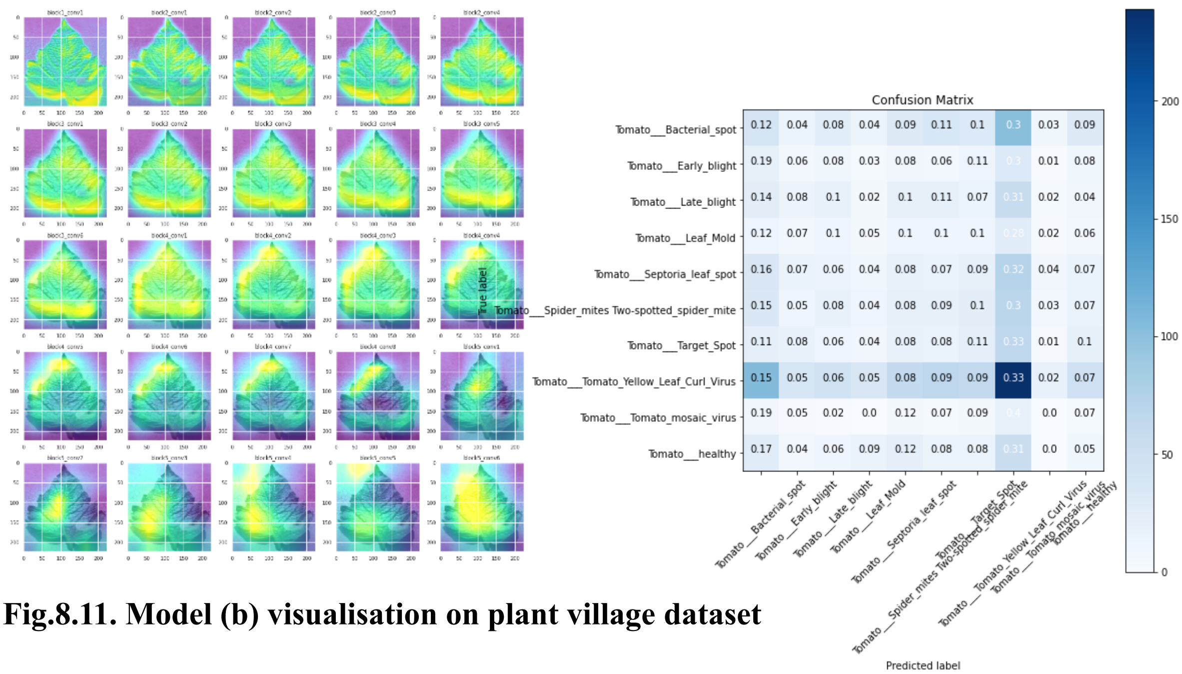


Fig.8.11. Model (b) visualisation on plant village dataset

On comparing the visualisation obtained from Fig. Model (a) visualisation and Fig. Model (b) visualisation, it is clearly seen that by eliminating the layers an improvement in the accuracy is seen as well as the architecture size has been reduced.

Real-world dataset

The dataset has 12 classes in all and gave an accuracy of 70% on experiment 2 model (a).

```
[[ 8  1  8  9  2  3  3  7  6  7  2 23]
 [ 1  1  4  2  1  2  1  0  1  0  1 10]
 [ 4  0  7  5  1  3  8  3  5  2  8 19]
 [ 8  0  5 10  1  2  7  4  4  3  3 17]
 [ 3  1  1  0  0  0  2  2  2  0  0  7]
 [ 3  3  5  2  1  1  4  2  6  2  2  7]
 [ 3  0  2  3  0  2  2  2  1  2  4 15]
 [ 5  1  2  2  3  3  2  3  2  4  4 10]
 [ 5  1  8  4  0  7  1  1  5  0  0 16]
 [ 7  2  5  4  2  6  1  3  3  1  4 13]
 [ 0  0  5  2  0  0  1  1  2  1  3 22]
 [20  7 22 33  6 12 16 15 12 10  4 63]]
```

```
Classification Report
                    precision    recall  f1-score   support

         Alt+miner       0.12      0.10      0.11        79
            B-wilt       0.06      0.04      0.05        24
               BLF       0.09      0.11      0.10        65
          E-blight       0.13      0.16      0.14        64
          L blight       0.00      0.00      0.00        18
           healthy       0.02      0.03      0.03        38
            insect       0.04      0.06      0.05        36
       leaf-miner        0.07      0.07      0.07        41
            mosaic       0.10      0.10      0.10        48
            physio       0.03      0.02      0.02        51
    powdery mildew       0.09      0.08      0.08        37
septoria leaf spot       0.28      0.29      0.29       220

          accuracy                           0.14       721
         macro avg       0.09      0.09      0.09       721
      weighted avg       0.14      0.14      0.14       721
```

Fig.8.12. Confusion matrix and Classification report of model (a) on real-world dataset

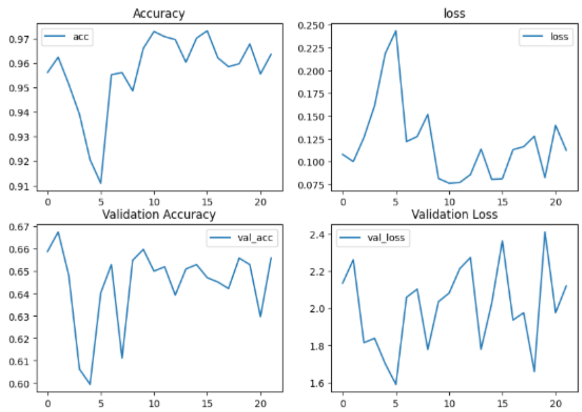


Fig. 8.13. Model (a) accuracy curve and loss curve on real-world dataset

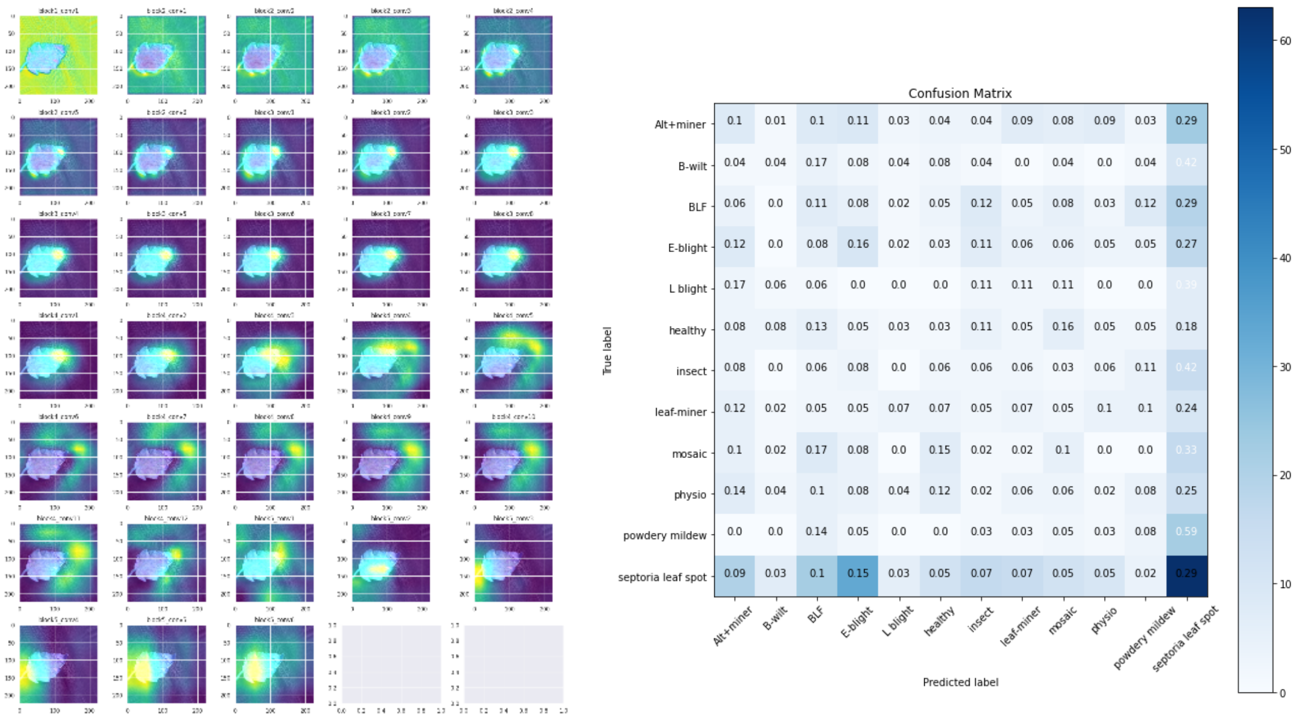


Fig.8.14. Model (a) visualisation on real-world dataset

The dataset has 12 classes in all and gave an accuracy of 71% on experiment 2 model (b).

```
[[13  2 10  5  2  1  2  8  4  7  1 24]
 [ 1  1  2  5  1  1  0  1  3  4  0  5]
 [ 8  1  8  8  0  5  2  1  1  5  3 23]
 [ 5  0  7  3  3 12  4  2  2  4  3 19]
 [ 1  0  4  0  0  1  0  0  1  0  2  9]
 [ 7  0  2  2  3  2  3  3  4  4  1  7]
 [ 4  1  5  2  2  0  2  4  3  3  1  9]
 [ 3  0  4  3  1  0  1  6  4  2  0 17]
 [ 7  0  6  3  2  1  1  3  5  5  3 12]
 [ 4  1  9  4  0  3  1  5  6  5  0 13]
 [ 1  0  6  3  2  2  0  1  3  3  2 14]
 [23  8 22 18  3 12 13 11 11 22  8 69]]
```

```
Classification Report
                     precision    recall  f1-score   support

          Alt+miner       0.17      0.16      0.17        79
             B-wilt       0.07      0.04      0.05        24
                BLF       0.09      0.12      0.11        65
           E-blight       0.05      0.05      0.05        64
           L blight       0.00      0.00      0.00        18
            healthy       0.05      0.05      0.05        38
             insect       0.07      0.06      0.06        36
         leaf-miner       0.13      0.15      0.14        41
             mosaic       0.11      0.10      0.11        48
             physio       0.08      0.10      0.09        51
     powdery mildew       0.08      0.05      0.07        37
septoria leaf spot        0.31      0.31      0.31       220

           accuracy                           0.16       721
          macro avg       0.10      0.10      0.10       721
       weighted avg       0.16      0.16      0.16       721
```

Fig. 8.15. Confusion matrix and Classification report of model (b) on real-world dataset

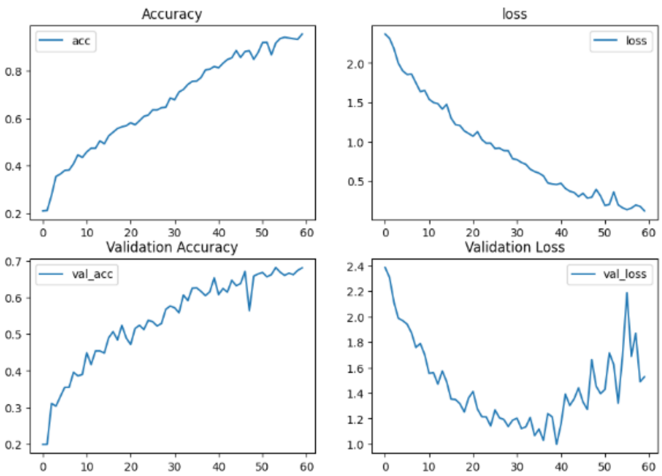


Fig.8.16. Model (b) accuracy curve and loss curve on real-world dataset

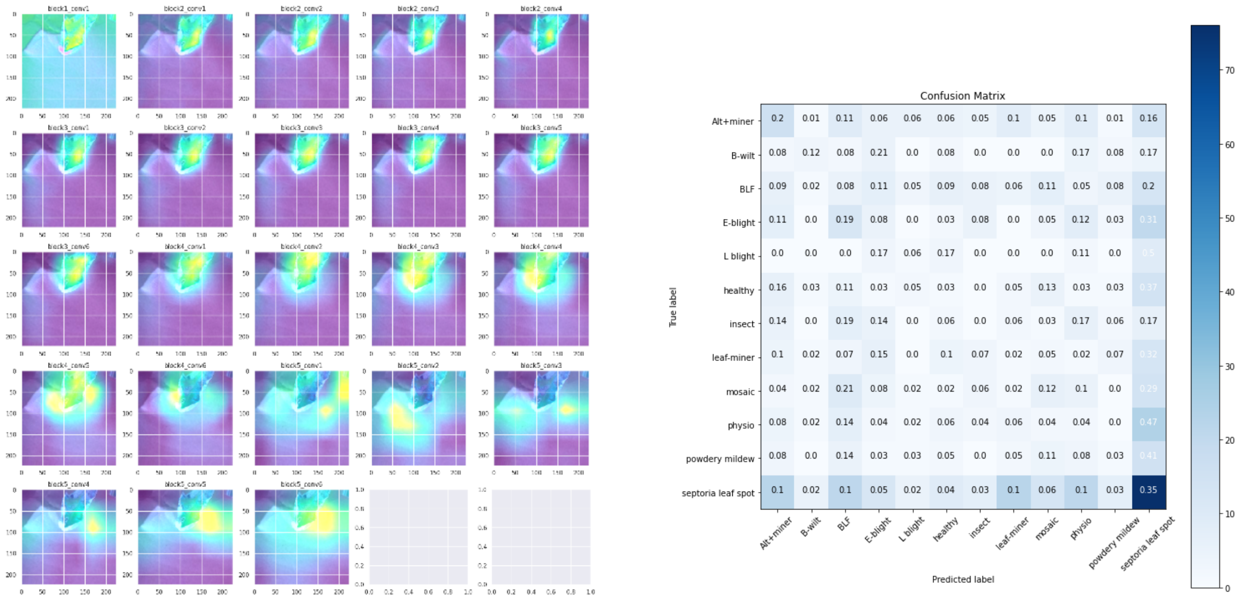


Fig. 8.17.Model (b) visualisation on real-world dataset

Hence from the above observations we can conclude that model (b) from experiment 2 gives satisfying results by using Grad-CAM as our visualizing technique and outperforms model (a) in every aspect.

Final Results

Table 8.2. Experiment results Plant village dataset

Sr. no	Architecture	Visualization Technique used	Training Accuracy	Validation Accuracy	Test Accuracy
1.	Resnet based proposed model-1	Grad-CAM	0.9854	0.9329	0.9315
2.	**Resnet based proposed model-2**	**Grad-CAM**	**0.9856**	**0.9329**	**0.9315**
3.	VGG16 transfer learning model-5	Score-CAM	0.9512	0.8611	0.8513
4.	VGG16 based proposed model-6	Score-CAM	0.9468	0.8904	0.8870
5.	**VGG16 based proposed model-7**	**Score-CAM**	**0.9863**	**0.9588**	**0.9533**

Table 8.3 Experiment results Real-world dataset

Sr. no	Architecture	Visualization Technique used	Training Accuracy	Validation Accuracy	Test Accuracy
1.	Resnet based proposed model-2	Grad-CAM	0.9608	0.6813	0.7171
2.	VGG16 based proposed model-7	Score-CAM	0.9814	0.6404	0.7212

Table 8.4Experiment results Internet downloaded dataset

Sr. no	Architecture	Visualization Technique used	Training Accuracy	Validation Accuracy	Test Accuracy
1.	Resnet based proposed model-2	Grad-CAM	0.2793	0.2727	0.2846
2.	VGG16 based proposed model-7	Score-CAM	1	0.6580	0.6516