ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2022.0371

AN IMPROVED CNN MODEL FOR CLASSIFICATION OF APPLE LEAF DISEASE AND VISUALIZATION USING WEIGHTED GRADIENT CLASS ACTIVATION MAP

Dharmendra Kumar Mahato[1], Amit Pundir[2] and Geetika Jain Saxena[3]
[1]*Department of Electronic Science, Babasaheb Bhimrao Ambedkar Bihar University, India*
[2,3]*Department of Electronics, Maharaja Agrasen College, India*

Abstract

Convolutional Neural Network (CNN), a particular type of forwarding feed network composed of convolutional, pooling, and fully connected layers, has become the dominant and most widely used deep learning architecture. Significantly enhanced effectiveness of ConvNets has made CNNs the go-to architecture model for almost every image processing-based application. CNNs automatically and adaptively learn spatial hierarchies of features with high accuracy, precision, and efficiency. This paper proposes three CNN models with 5, 6, and 7 layers with two types of classification layers at the top of the model, resulting in six kinds of models. Each model is trained on apple leaf diseases obtained with augmentation deployed on the PlantVillage dataset containing images of healthy and three types of leaf diseases. The trained models are compared on training time, testing accuracy, testing time. The best performing model (6-layer based model with fully connected layer as a classifier (6FC) in our case) yields 99.14% accuracy. This best-performing model is also compared with the state-of-art models such as VGG-16, InceptionV3, and MobileNetV2, trained using the transfer learning approach. After model comparison, we found our best model (6FC) outperformed the other models based on evaluated performance metrics with improvements as 3.94% gain in accuracy, 25.97% reduced parameters, and less training time (0.51hr) and testing time (20.5 sec) compared to VGG-16. Comparing precision, recall, and f1-score values are also found high (between 0.98 to 1) with our proposed model. The weighted gradient class activation map (Grad-CAM) technique generates a visualization of class predictions on the test dataset. The Grad-Cam visualization of results validates the prediction score attained by the proposed model.

Keywords:

Convolutional Neural Network, Grad-CAM, Deep Learning, Data Augmentation, Transfer Learning

1. INTRODUCTION

In recent years a surge of deep learning models in the field of computer vision is being witnessed, considerably due to their significant capabilities, accuracies, and other improved performance parameters in a variety of visual processing and understanding applications such as object detection and recognition, human activity recognition, image segmentation, and activity classification. The deep learning approaches explore the unknown structures in the input data and discover with high confidence good representations, often at multiple levels. These methods learn pattern hierarchies in which higher-level features are described in terms of the composition of lower-level features. One of the core research areas in image processing-based learning is artificial neural networks, i.e., convolutional neural networks (CNN).

With CNN in deep learning, it can analyze large dataset sizes to conclude based on the patterns (features) learned. Thus, everything is done without any human intervention while learning features from images. Mainly it is used for two purposes: 1. Object detection in the image (presence or absence of object) and 2. Localization of objects detected in the image. Earlier, these were done with the help of hand-crafted feature extraction with pattern recognition algorithms, but now these tasks are done readily with various proposed models based on CNNs. Multiple factors affect deep learning methods, but a large amount of required data is a must; otherwise, overfitting (responsible for low performance) is the biggest issue. However, the data augmentation technique when data size is small can handle this.

However, even with an impressive performance in image classification tasks, it is not easy to interpret the CNN model decisions. In this way, we can assume neural networks as black boxes [1]. To understand these model predictions, we need to dive into the field of explainable CNN. There are various methods [2]-[6] that help us to understand the insight into the decision making of such CNN models. These methods are a visual interpretation of model predictions. The oldest and frequently used method, called saliency map, is used for model interpretation in deep learning. The saliency map of an input image provided to the network specifies parts or regions of the image. It contributes most to the activity of a specific layer in the network or as a whole network decision. There are mainly three approaches to get the saliency map of an input image. The first approach is using deconvolutional networks [7]. In this method, a deconvolutional network is used that reconstructs the input from the activation of that layer. Along with deconvolution (transformed version of convolutional filter) operation, un-pooling (inverse of pooling) and ReLU (inverse of itself) are used. Although pooling operation is non-invertible, a module called switch is used in the deconvolutional network to recover maxima positions in the forward pass. The second method [5] is the most straightforward approach of getting a saliency map. In this method, the backpropagation algorithm computes the gradients of logits w.r.t. to the networks input. In addition, this backpropagation method can highlight pixels of the input image based on the gradient they receive, which shows their contribution to the final score. The author of the paper [6] combined these approaches and proposed the third method called a guided backpropagation algorithm. Although saliency maps are used to interpret CNNs, a couple of papers have shown that saliency maps are not always reliable [4]. B. Zhou et al. (2016) introduced another approach, class activation map (CAM), which explains CNNs [3]. The papers authors replaced the stack of fully connected layers at the end of the CNN model with a layer named Global Average Pooling (GAP). GAP averages the activations of each feature map along with depth and concatenates these averaged values. It results in output as a vector which is fed to the final softmax loss layer. Using this architecture, we can highlight the critical regions of the image by projecting back the weights of the output on the convolutional feature maps using heatmap visualization. A more versatile version of CAM is Grad-CAM method by Selvaraju et

al. [2], producing visual explanations for any arbitrary CNN, even if the network contains fully connected layers as a classification layer. The approach to obtain Grad-CAM of an image is quite a similar approach as for obtaining saliency maps. In this method, the gradients of any target concept score (logits for any class of interest such as cat or dog) flow into the final convolutional layer. Then the importance score based on the gradients is computed, and a coarse localization map highlighting the critical regions in the image is produced for predicting that concept.

This paper proposes using the CNN model to classify plant disease associated with apple leaves. The features learned from the image of the apple leaf are used for enhanced classification accuracy. We propose three models for feature learning (by varying depth of model) and two types of layers: fully connected (FC) dense layer and GAP layer. A total of six architectures models results for feature learning and classification tasks. Performance evaluation of all the proposed models is done based on overall accuracy, confusion matrix, and ROC curves. The best-proposed model is compared with state-of-art models such as VGG-16, InceptionV3 and MobileNetV2, trained, tested, and evaluated on the same dataset. The proposed model was found to be performing better than the three standard models on the selected dataset. The visualization of the classification process in the proposed CNN model is done by generating the heat map using the Grad-CAM technique. These activation maps highlight the regions and provide visualization of the area in the leaf image having a class of disease based on features learned. These regions are similar to the actual visual characteristic in the leaf of a particular disease and help in the evaluation of weakly supervised object localization as proposed in the papers [8]-[10].

The paper is organized as follows: Section 2 deals with literature survey reporting related work. Then, in section 3, the methodology of network architecture design and Grad-CAM visualization technique is presented. In section 4, the experimental results of the training, testing, and validation of the models are given. Finally, in section 5, the papers conclusion detailing the results establishes the proposed model as the best-performing model for the selected dataset.

2. RELATED WORK

As per the literature survey, CNN based model is the most used method for image-based learning among various deep learning architectures. This is because it can analyze high-dimensional, unstructured data such as image, text, and audio. However, classical Machine Learning (ML) is challenging to handle, i.e., non-deep-learning or hand-crafted (non-ML) algorithms.

Much research is going into plant/crop disease detection using deep learning convolutional neural networks (DLCNN). The work mainly detects diseases using images of leaves in various conditions (laboratory and actual field of plant/crop). Mohanty et al. [11] analyses two CNN-based models (AlexNet [12] and GoogLeNet [13]) on the ability to detect 26 diseases, 14 crop species, 38 class labels in a dataset of 54,306 images. Three types of images (Color, Grayscale, Leaf Segmented), two approaches (transfer and training from scratch), and various train-validate data split ratio options were adopted while training these models. With this approach, they achieved the best accuracy of 99.35% using GoogLeNet transfer learning with color images. Liu et al. (2017) proposed a CNN model to identify the four common types of apple leaf diseases with a dataset containing 13,689 images [14]. Their experimental results show model achieving an overall accuracy of 97.62%. Compared with other standard models such as AlexNet, GoogLeNet, ResNet-20 [15], and VGG-16, their model achieved better accuracy with reduced parameter requirements. In a paper [16], five standard CNN models are presented for plant disease detection and diagnosis using deep learning methodologies. Models were trained on a database of 87,848 images containing 25 different plants in a set of 58 distinct classes. The best performance of 99.53% success rate using the VGG model was achieved. However, the total training time for that model, on a single GPU was about 5.5 days. In another paper [17], a method based on region-of-interest-aware (ROI) deep convolutional neural networks (DCNN) is proposed to recognize apple leaf diseases on a dataset containing three classes of apple leaf with two diseases and one healthy class. The proposed ROI-aware DCNN architecture consists of two subnetworks, i.e., ROI subnetwork and VGG-subnetwork. One predicts the ROI feature map for dividing the input images into the background, leaf area, and spot area, and the second (VGG-subnetwork) classifies the leaf diseases. The ROI-aware DCNN achieved better recognition accuracy (84.3%) than state-of-the-art methods such as the multiscale-based deep feature extraction and pooling (MDFEP) method, fisher vector encoding (FVE) with scale-invariant feature transform (SIFT), and DCNN-based bilinear model. In a paper [18], apple disease dataset of 8400 leaf images of five infected and healthy. Using 70% and 30% train validation split on a prepared dataset with a modification and training on a deep learning model named ResNet-34 achieved 97.18 % accuracy for automatic classification of apple diseases. Y. Guo et al. (2020) proposed a mathematical model of plant disease detection and recognition using a deep learning approach [19]. A region proposal network (RPN) with Chan–Vese (CV) algorithm was proposed to recognize and localize the leaves in the complex background. The segmented leaves were fed into the transfer learning model and trained by the dataset of diseased leaves with a simple background. Total 4714 images of four classes, including one healthy and three disease classes, were used for the experiment. The accuracy achieved was 83.57% compared to the traditional ResNet-101 model (42.5%). A DCNN based early diagnosis method for apple tree leaf diseases was proposed by X. Chao et al. [20]. They have used a dataset of five common diseases and healthy leaves, which contain images in both laboratories and cultivation field conditions. The DCNN model proposed is a combination of DenseNet and Xception, using global average pooling. They extracted features by the proposed DCNN model then used a support vector machine to classify the apple leaf diseases and achieved an overall accuracy of 98.82%, which is higher than some standard models.

Traditional and deep learning approaches and challenges are discussed to solve plant disease and pest disease problems in a paper by Liu & Wang (2021) [21]. Even with excellent performance, it is not easy to analyze the reason behind their work. Several approaches for understanding and visualizing CNN have been developed in the literature. The reported works suggested methods with which one can understand the outcomes from the CNN model. B. Zhou et al. (2014) suggested work to perform object localization without using any bounding box

annotations required by many other object detection algorithms. Instead, they used Class Activation Mapping (CAM), which provides them the highlighted discriminative object part upon which the CNN models prediction depends [22]. Jia and Shen (2017) applied a CAM-based approach in two-stage with only one network. A model is trained with an image in the first stage, and then image cropping at the maximum activation map area is performed in the input image. The cropped image is fed again to re-train the same model and conclude the final result in the second stage [23]. Similarly, Charuchinda et al. (2019) used CAM for land cover mapping where the high values indicated a high probability of the presence of a particular class [24]. There is no need for manual labelling with their approach, even though getting land cover mapping but with low accuracy. Sun et al. [25] proposed a deep learning approach with CAM for fault region diagnosis in the image to characterize the status of the machines. They can localize the fault in the machine image with the help of CAM. The work proposed a novel industrial application for automatic machine condition monitoring systems. Jiang et al. [26] proposed a Single Shot Detection (SSD) approach to locate an object in the image. Post classification, the real-time object detection of five types of apple leaf disease localization in images is done. The classification step obtained and reported feature activation results in different disease spots from the images background (leaf area).

The activation mapping approach was applied in the present work at the image classification training step, and then predictions are made by the trained model with the activation map obtained for the proposed CNN models using Grad-CAM.

3. METHODOLOGY

In this paper, we have proposed CNN models with different numbers of layers for classification purposes. After model training and evaluation based on several parameters, the best model out of the proposed models was identified. Finally, this model was compared with various pre-trained models such as VGG-16, InceptionV3, and MobileNetV2. The results confirmed the better performance achieved by the proposed model among all the models. The prediction performance is visualized with the activation heat maps over the input image, and all the results support the prediction results.

3.1 DATASET PREPARATION

The dataset for the present work is taken from the PlantVillage dataset repository [11]. The dataset contains four classes of apple leaf diseases such as Apple Scab, Apple Cedar Rust, Apple Black Rot, and healthy leaves. The total number of images, 2536 images, are taken, and the distribution of images among classes is given in Table.1. As the total number of images is insufficient for CNN model training and the models may result in overfitting, manual augmentation of images to generate new images from the available dataset was done. The augmentation process generated 12 new images of each image using left-right flipping, brightness control, random rotation, horizontal flip, and noise addition such as gaussian, local var, Poisson, salt, pepper, salt, pepper, and speckle noise. Data augmentation, therefore, resulted in 32,968 images, which were further split into train, validation, and test set as 70%, 15%, and 15%, respectively. As the data remained imbalanced, the number of images per class varied, class imbalance function in Keras package was applied to avoid class-wise biasing.

Table.1. Dataset distribution

Apple Leaf Disease Class	No. of Images (original dataset)	No of Images (after augmentation)
Apple Scab	504	6552
Apple Cedar Rust	220	2860
Apple Black Rot	496	6448
Apple Healthy	1316	17108
Total	2536	32968

3.2 CNN BASED MODELS

3.2.1 Proposed CNN Models:

The proposed models have 5, 6, and 7 features extracting convolutional layers but different classification layers, one model having fully connected layers (FC) and other global average pooling (GAP) layers. The two architectures with three different feature extracting layers result in six different models proposed in this study. These models are named based on the number of layers and classifiers used. The GAP is a pooling operation designed to replace FC layers in classical CNNs models. The network generates one feature map at the last convolutional layer for each corresponding category of the classification task. The GAP layer takes an average of the feature map, and the resulting vector is fed directly into the softmax layer for classification probability. As there is no parameter to optimize the GAP layer, overfitting is avoided at this layer. Furthermore, GAP sums the spatial information and is more robust to spatial translations of the input image.

The proposed models are deep learning convolutional neural networks; VGG-16 inspires the architecture with few modifications in the layers, the modification involves keeping a minimum number of layers and hence a reduced number of learnable parameters. The proposed model architecture with feature extracting and classification layers is shown in Fig.1 and Fig.2. Similarly, 6- and 7-layer models were designed containing 6 and 7 feature extraction layers respectively. Hence these models had GAP and FC.

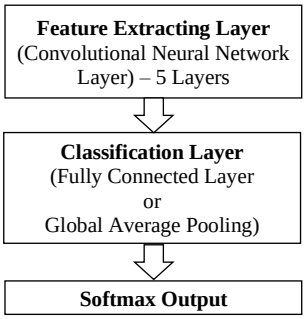


Fig.1. 5 Layer-based proposed CNN model

The FC layers contain two layers (dense 1 and dense 2) as a classifier with two dropouts post flattening after the 5th convolutional layer. The fully connected layers are prone to overfitting, thus hampering the generalization ability of the overall model. The dropout layer [27] acts as a regularizer that

randomly sets a few of the activations to the fully connected layer layers to zero during model training. It improves the generalization ability and mainly prevents overfitting problems [12]. Max-pooling layers after each convolutional filter reduce the feature maps front dimensions, which ensures a limited weight and better computational efficiency for the model.

Model: "Sequential"		
Layer (type)	**Output Shape**	**Total Parameters**
Conv 1	(224, 224, 64)	1792
Conv 2	(224, 224, 64)	36928
Max_pooling 1	(112, 112, 64)	0
Conv 3	112, 112, 128)	73856
Max_pooling 2	(56, 56, 128)	0
Conv 4	(56, 56, 256)	295168
Max_pooling 3	(28, 28, 256)	0
Conv 5	(28, 28, 256)	590080
Max_pooling 4	14, 14, 256)	0
Flatten	(1,50176)	0
Dense 1	(1, 1024)	51381248
Dropout 1	(1, 1024)	0
Dense 2	(1, 1024)	1049600
Dropout 2	(1, 1024)	0
Dense 3	(1, 4)	4100
Total parameters: 53,432,772 Trainable parameters: 53,432,772 Non-trainable parameters: 0		

(a)

Model: "Sequential"		
Layer (type)	**Output Shape**	**Total Parameters**
Conv 1	(224, 224, 64)	1792
Conv 2	(224, 224, 64)	36928
Max_pooling 1	(112, 112, 64)	0
Conv 3	(112, 112, 128)	73856
Max_pooling 2	(56, 56, 128)	0
Conv 4	(56, 56, 256)	295168
Max_pooling 3	(28, 28, 256)	0
Conv 5	(28, 28, 256)	590080
Max_pooling 4	(14, 14, 256)	0
Global_average_pooling	(1, 256)	0
Dense 1	(1, 4)	1028
Total parameters: 998,852 Trainable parameters: 998,852 Non-trainable parameters: 0		

(b)

Fig.2. Model architecture (a) 5FC and (b) 5GAP

The input image size is 224×224×3, and the kernel size is 3×3 at all layers in all the proposed models. The parameter in each type of model is shown in Table.2, indicating the highest parameters (53,432,772) for 5FC. The actual apple leaf images in the selected dataset are of size 256×256×3, rescaled to model input of size 224×224×3 before feeding to the network in each model.

3.2.2 VGG-16 Model:

VGG-16 is a CNN model that achieved 92.7% top-5 test accuracy in ImageNet data of 1000 classes. This model introduced by Simonyan and Zisserman [28] is an improvement over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another.

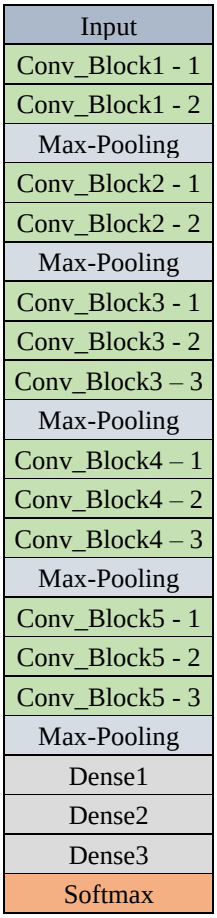


Fig.3. VGG-16 model

The input is of fixed size 224×224 RGB image at the top of a stack of convolutional (Conv) layers, as shown in Fig.3. A tiny receptive field filter of size 3×3 is used in each layer of the architecture. In addition, five max-pooling layers are used for spatial pooling using a 2×2-pixel window with stride 2. There were three fully connected (FC) dense layers at the end of the architecture. The first two with 4096 channels each, and the third was 1000 for classification for 1000 classes with a softmax layer, as shown in Fig.3.

3.2.3 InceptionV3 Model:

The Inception deep convolutional architecture was first introduced as GoogLeNet in 2014 [13], and afterward, various versions of the architecture were developed. Variation includes batch normalization [29] named InceptionV3 and later factorization [30], referred to as InceptionV3. The idea of factorization of convolutions is to reduce the number of parameters without decreasing the network efficiency. It is achieved by replacing a 5×5 filter (25 parameters) with two 3×3 filters having 18 parameters (3×3+3×3). It results in a 28%

reduction in a parameter named inception module A as shown in Fig.4.

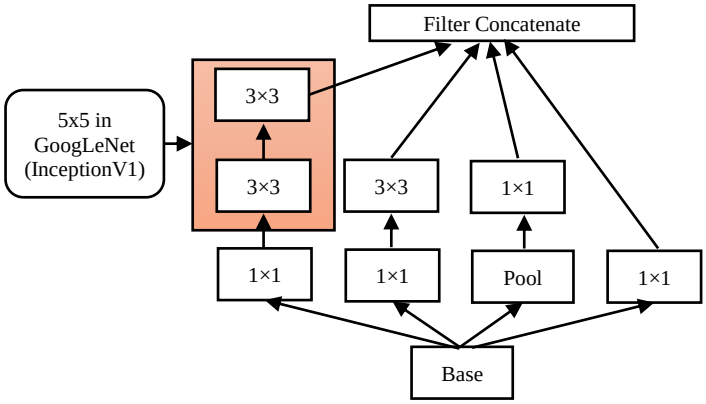


Fig.4. Inception module using factorization

Improvement in the inception layer, such as factorization into asymmetric convolutions, resulted in inception module B, and high dimensional representation using inception module C was introduced. It ensured less parameter requirement, and consequently, the network ability to go deeper. In InceptionV3, an auxiliary classifier with batch normalization is used as a regularizer.

3.2.4 MobileNetV2:

The MobileNet is the first mobile computer vision model to be used in mobile applications in which two-step separable convolutions (depthwise and pointwise) are used [31]. It significantly reduces the number of parameters compared to the other network with regular convolutions with the same depth in the architecture resulting in lightweight deep neural networks. MobileNetV2 [32] module has inverted residual structure improvement in the network. Non-linearity in thin layers is handled in this version yielding state-of-the-art performances for object detection and semantic segmentation. The model feature is shown in Fig.5.

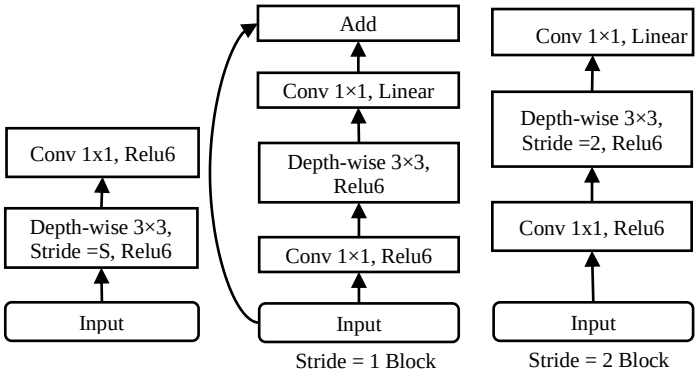


Fig.5. MobileNet Models: (a) MobileNetV1 and (b) MobileNetV2

3.3 TRANSFER LEARNING

The proposed CNN model are trained from scratch on the dataset of apple leaf disease. As the deep learning models (VGG-16, InceptionV3) need extensive computational resources, pre-trained models on vast collections of datasets such as ImageNet were fine-tuned on the dataset with few modifications in the last stages of the network. Using this transfer learning approach [33]-[37] performance of standard models (VGG-16, InceptionV3, and MobileNetV2) were compared with the proposed models.

3.4 GRAD-CAM

After models training, weights with each model learned, the model evaluation is done. Finally, class activation maps using Grad-CAM were generated for the proposed model for understanding regions of interest of the image upon which models ability to predict depends.

To obtain the class-discriminative localization map defined by $L^c_{Grad-CAM} \in \mathbb{R}^{u \times v}$ where width u and height v for given class c, we need to calculate the gradient of the score (y_c, before softmax layer) for class c feature map activation (A^k) of a convolutional layer i.e. $\partial y^c / \partial A^k_j$. The neuron significant weights, α^c_k obtained over the width and height (indexed by i and j respectively) during back propagation gradients concerning activations as:

$$\alpha^c_k = \frac{1}{Z} \sum_i \sum_j \frac{y^c}{A^k_{ij}} \quad (1)$$

The alpha α^c_k value for class c and feature map k is weighted with corresponding feature map and hence calculate a weighted sum of feature map as the final Grad-CAM heatmap using equation;

$$L^c_{Grad-CAM} = \mathrm{Re}LU\left(\sum_k \alpha^c_k A^k \right) \quad (2)$$

ReLU activation function operation is applied to emphasize only the positive values. In this way, a Grad-CAM heatmap of size $u \times v$, which is the size of the final feature map, is obtained. As the size was smaller than the input image, an up-sampling of this heatmap was performed to match the size of the original image for final visualization [2].

4. RESULTS

4.1 EXPERIMENTAL SETUP

A fixed partitioning scheme has been chosen to train and evaluate the proposed models model performance and plot associated activation curves. The manually augmented apple leaf image dataset had 70%, 15%, and 15% training, validation, and test splits for all the models. The Windows 10 based Anaconda3 environment with python3 language was chosen for the investigation. The system used had Intel i5, 9th generation 2.4 GHz, and 8GB RAM, GPU-Nvidia GTX 1050 Ti, 4GB NVRAM with CUDA packages. All the proposed models were trained from scratch using stochastic gradient descent (SGD) with a learning rate of 0.001, momentum of 0.9, batch size 16, and 30 epochs. The standard models were trained using the transfer learning approach, and in the case of InceptionV3, the RMSprop optimizer was used to train the model. Library packages such as NumPy, Matplotlib, Sklearn, Keras, Tensorflow, etc., were used to model definition, training, evaluation, and plotting model performance. The Keras package was used to obtain and plot the activation and heat maps within various layers of the CNN model [38].

4.2 EXPERIMENTAL RESULTS

All the models were trained and tested on augmented images, and the model evaluation comparison was made in two steps. We studied the proposed models (5, 6, and 7 layer-based models with fully connected or GAP layers) for the performance analysis in the first step. In the second step, the best-identified model was compared with standard pre-trained models such as VGG-16, InceptionV3, and MobileNetV2 on the same dataset. The evaluation parameters, such as accuracies, training time, and test time obtained for all the models, are shown in Table.2. It shows the 6FC model attaining the highest training and testing accuracy (Fig.6) of 99.14% amongst all the proposed models. Although the training and testing time of 6FC model is relatively high, evaluation on other parameters such as training curve, confusion matrices, precision, recall, and F-1 score outperforms all the standard architectures considered in this study. An improvement of 3.76% accuracy is observed in the 6FC over the pre-trained VGG-16 model.

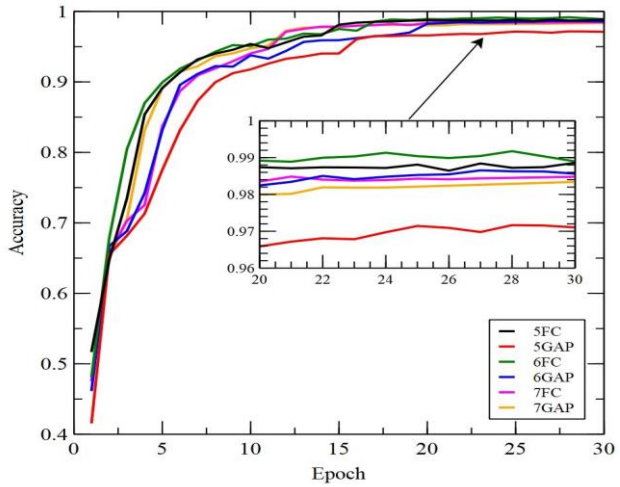


Fig.6. Training curves of all the models: training accuracy vs. epoch

Table.2. Experimental results and comparison for all models

Model	Total Parameters	Training Time (Hrs.)	Test Accuracy (%)	Testing time (s)
5FC	53,432,772	2.99	98.94	17.14
5GAP	998,852	2.47	97.69	16.22
6FC	30,692,548	3.37	**99.14**	18.67
6GAP	3,949,764	3.26	99.02	18.36
7FC	10,311,108	2.64	98.88	17.75
7GAP	4,539,844	2.28	98.63	17.75
VGG-16	41,459,524	3.88	95.38	39.17
InceptionV3	21,810,980	2.19	94.10	43.45
MobileNetV2	2,257,984	0.79	92.59	9.49

Table.3. Confusion matrices, precision, recall, and f1-score of the best model and pre-trained models

6FC Model

Predicted class \ *Actual Class*				
	956	0	2	17
	2	931	0	3
	2	0	401	0
	10	1	5	2566

Class	Precision	Recall	F1-Score
0	0.99	0.98	0.98
1	1	0.99	1
2	0.98	1	0.99
3	0.99	0.99	0.99

VGG-16 Model

Predicted class \ *Actual Class*				
	852	19	52	52
	1	922	4	9
	0	0	397	6
	26	9	48	2499

Class	Precision	Recall	F1-Score
0	0.97	0.87	0.92
1	0.97	0.99	0.98
2	0.79	0.99	0.88
3	0.97	0.97	0.97

InceptionV3 Model

Predicted class \ *Actual Class*				
	822	19	57	77
	1	926	2	7
	1	1	399	2
	60	53	9	2460

Class	Precision	Recall	F1-Score
0	0.93	0.84	0.88
1	0.93	0.99	0.96
2	0.85	0.99	0.92
3	0.97	0.95	0.96

MobileNetV2 Model

Predicted class \ *Actual Class*				
	781	40	44	110
	6	905	14	11
	4	6	383	10
	44	66	8	2464

Class	Precision	Recall	F1-Score
0	0.94	0.8	0.86
1	0.89	0.97	0.93
2	0.85	0.95	0.9
3	0.95	0.95	0.95

Also, the proposed model took 18.67 sec time to process and detect class disease in the 4896 total testing images, which is an improvement over VGG-16.

The comparison of 6FC with VGG-16, InceptionV3, and MobileNetV2 architectures, in terms of training and validation accuracy, is shown in Fig.7. It is evident from the figure that the proposed 6FC model is the best in terms of training and testing accuracy convergence. The confusion matrix of performance metr

-ics of the proposed and the conventional architectures, showing the class-wise prediction, are reported in Table.3. The diagonal elements show the exactness between the actual class and the predicted class. For the 6FC model, the class-wise prediction, precision, recall, and F1 score are better than other architectures. In addition to the highest test accuracy of 99.14%, precision, recall, and F1 score ranges between 0.98 to 1 for all class predictions indicating better performance than the other standard models. Regarding the time taken to detect class disease, the 6FC model took 18.67 seconds than the other models, except MobileNetV2, to process 4896 test images. MobileNetV2 was the fastest model, which took only 9.49 seconds for the same test sample size, but its accuracy was the lowest.

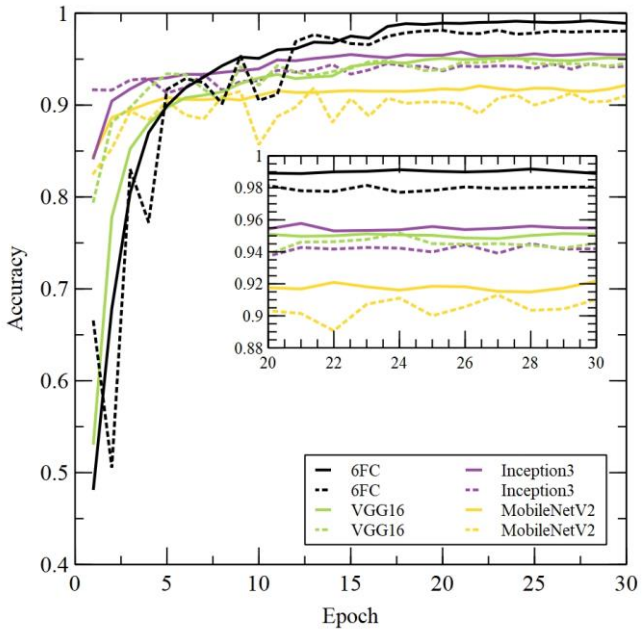


Fig.7. Training and validation accuracy of the 6FC and pre-trained models

Receiver Operating Characteristics (ROC) curves are plotted in Fig.8 for multi-classification problems using the one vs. all techniques in which each label is considered at a time, and all the others can be grouped as one label. Class-wise performance of the models with micro-averaged ROC curves is visualized. 6FC is observed to have the best class separation amongst the models by having the largest area under the curve.

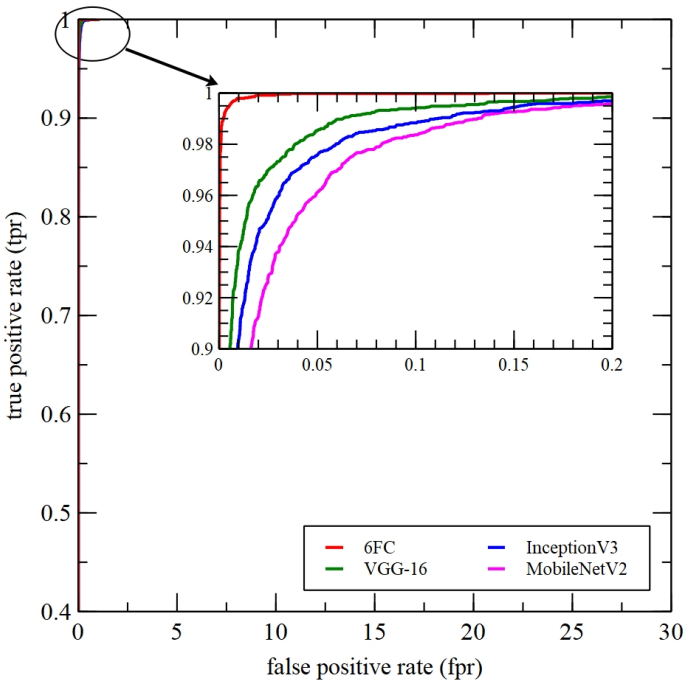
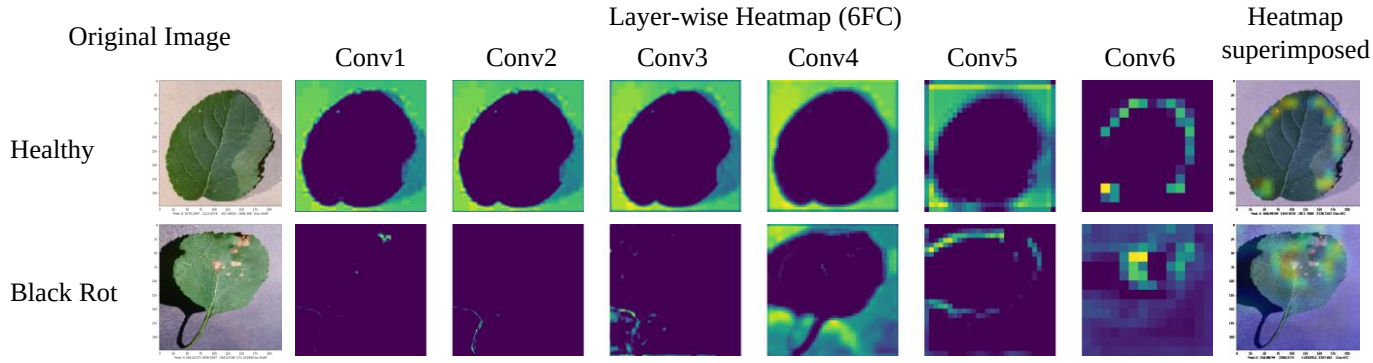


Fig.8. Micro-Averaged ROC curves of the 6FC and pre-trained models

To further validate and visualize the results of the 6FC model, visualization techniques were used to understand better the prediction mechanism of CNN models for a particular class in the image. One such technique for image-based CNN visualization is Grad-CAM [2]. One test image of each class belonging to apple leaf disease and a healthy leaf was fed to the 6FC model, and the corresponding heat map generated at each layer is shown in Fig.9. Infected regions were highlighted in a leaf image at various convolution layers. When the heatmap of the last layer was superimposed on the original image, class- wise features of a particular class were clearly highlighted at the last layer. Using Grad-CAM, we can substantiate that the proposed model focuses on learning the features in the image for a particular class with adequate accuracy and precision. The regions highlighted at the end of the network indicate the region of interest and are specific to the class type. This method can also be helpful in object detection problems for highlighting diseased regions without the need for the annotation of images.



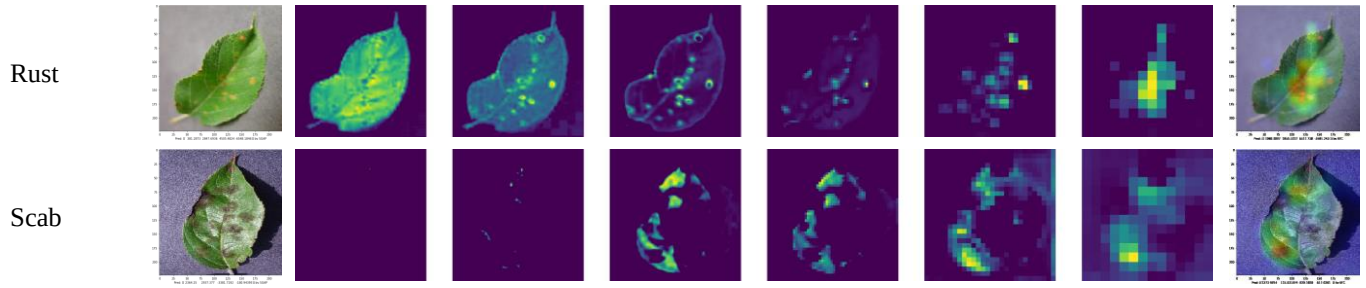


Fig.9. 6FC model activation map plot at each layer and the last layer heatmap superimposed on the input image. First column: input images, row-wise: corresponding heatmap at each layer in each class, last column: corresponding heatmap of the last layer superimposed on the input image

5. CONCLUSION

The classification of apple leaf disease is performed with the standard VGG-16, InceptionV3, and MobileNetV2 models and the proposed models with two variants, i.e., fully connected and global average pooling layers. The proposed 6-layer model (6FC) with a fully connected layer as a classifying layer performed the best in terms of overall accuracy (99.14%), F1 score, precision, and ROC. In all performance metrics, the results are better than attained in the state-of-the-art standard models.

Grad-CAM visualization of activation map w.r.t. the class over the input image with and without the disease were obtained to support the prediction score by the proposed model. The activation maps for different classes indicate the region in the input image to predict that class quite accurately. The results further validate and confirm the accuracy and efficiency of the proposed model in identifying and classifying the disease correctly.

REFERENCES

[1] V. Buhrmester, D. Munch and M. Arens, "Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 232-239, 2019.

[2] R.R Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, "Gradcam: Visual Explanations from Deep Networks via Gradient-Based Localization", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 618-626, 2017.

[3] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva and A. Torralba, "Learning Deep Features for Discriminative Localization", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921-2929, 2016.

[4] P.J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K.T Schutt, S. Dahne, D. Erhan and B. Kim, "The (Un)reliability of Saliency Methods", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 415-449, 2017.

[5] K. Simonyan, A. Vedaldi and A. Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2221-2228, 2013.

[6] J.T. Springenberg, A. Dosovitskiy, T. Brox and M. Riedmiller, "Striving for Simplicity: The All Convolutional Net", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1189-1197, 2015.

[7] M.D. Zeiler, and R. Fergus, "Visualizing and Understanding Convolutional Networks", *Proceedings of IEEE Conference on Computer Vision*, pp. 818-833, 2014.

[8] R.G. Cinbis, J. Verbeek and C. Schmid, "Weakly Supervised Object Localization with Multi-Fold Multiple Instance Learning", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 1, pp. 189-203, 2017.

[9] M. Oquab, L. Bottou, I. Laptev and J. Sivic, "Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1717-1724, 2014.

[10] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is Object Localization for Free? Weakly-Supervised Learning with Convolutional Neural Networks", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 685-694, 2015.

[11] S.P. Mohanty, D.P. Hughes and M. Salathe, "Using Deep Learning for Image-Based Plant Disease Detection", *Frontiers in Plant Science*, Vol. 7, pp. 1419-1427, 2016.

[12] A. Krizhevsky, I. Sutskever and G. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks", *Advances in Neural Information Processing Systems*, Vol. 25, pp. 1106-1114, 2012.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going Deeper with Convolutions", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-14, 2014.

[14] B. Liu, Y. Zhang, D. He and X. Li, "Identification of Apple Leaf Diseases Based on Deep Convolutional Neural Networks", *Symmetry*, Vol. 10, No. 1, pp.1-18, 2017.

[15] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 330-345, 2015.

[16] K.P. Ferentinos, "Deep Learning Models for Plant Disease Detection and Diagnosis", *Computers and Electronics in Agriculture*, Vol. 145, pp. 311-318. 2018.

[17] H. Yu and C. Son, "Apple Leaf Disease Identification through Region-of-Interest-Aware Deep Convolutional

Neural Network", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-19, 2019.

[18] A.I. Khan, S.M.K. Quadri and S. Banday, "Deep Learning for Apple Diseases: Classification and Identification", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-13, 2020.

[19] Y. Guo, J. Zhang, C. Yin, X. Hu, Y. Zou, Z. Xue and W. Wang, "Plant Disease Identification Based on Deep Learning Algorithm in Smart Farming", *Discrete Dynamics in Nature and Society*, Vol. 2020, pp. 1-11, 2020.

[20] X. Chao, G. Sun, H. Zhao, M. Li and D. He, "Identification of Apple Tree Leaf Diseases Based on Deep Learning Models", *Symmetry*, Vol. 12, No. 7, pp. 1-17, 2020.

[21] J. Liu and X. Wang, "Plant Diseases and Pests Detection based on Deep Learning: A Review", *Plant Methods*, Vol. 17, No. 22, pp. 1-18, 2021.

[22] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva and A. Torralba, "Object Detectors Emerge in Deep Scene CNNs", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-12, 2014.

[23] X. Jia and L. Shen, "Skin Lesion Classification using Class Activation Map", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 441-448, 2017.

[24] P. Charuchinda, T. Kasetkasem, I. Kumazawa and T. Chanwimaluang, "On the Use of Class Activation Map for Land Cover Mapping", Proceedings of IEEE International Conference on Electrical Engineering Electronics, Computer, Telecommunications and Information Technology, pp. 653-656, 2019.

[25] K.H. Sun, H. Huh, B.A. Tama, S.Y. Lee, J.H. Jung and S. Lee, "Vision-Based Fault Diagnostics using Explainable Deep Learning with Class Activation Maps", *IEEE Access*, Vol. 8, pp. 129169-129179, 2020.

[26] P. Jiang, Y. Chen, B. Liu, D. He and C. Liang, "Real-Time Detection of Apple Leaf Diseases using Deep Learning Approach Based on Improved Convolutional Neural Networks", *IEEE Access*, Vol. 7, pp. 59069-59080, 2019.

[27] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R.R. Salakhutdinov, "Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 666-678, 2012.

[28] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 441-449, 2014.

[29] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 881-889, 2015.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 771-776, 2015.

[31] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", *Proceedings of IEEE Conference on Computer Vision*, pp. 1-12, 2017.

[32] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 45-58, 2018.

[33] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang and C. Liu, "A Survey on Deep Transfer Learning", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 89-98, 2018. K

[34] . Weiss, T.M. Khoshgoftaar and D. Wang, "A Survey of Transfer Learning", *Journal of Big Data*, Vol. 3, No. 9, pp. 1-13, 2016.

[35] Y. Nagaraju, Venkatesh, S. Swetha and S. Stalin, "Apple and Grape Leaf Diseases Classification using Transfer Learning via Fine-tuned Classifier", *Proceedings of IEEE International Conference on Machine Learning and Applied Network Technologies*, pp. 1-6, 2020.

[36] N.K. Hebbar and A.S. Kunte, "Transfer Learning Approach for Splicing and Copy-Move Image Tampering Detection", *ICTACT Journal on Image and Video Processing*, Vol. 11, No. 4, pp. 2447-2452, 2021.

[37] V.C. Burkapalli and P.C. Patil, "Transfer Learning: Inception-V3 Based Custom Classification Approach for Food Images", *ICTACT Journal on Image and Video Processing*, Vol. 11, No. 1, pp. 2261-2267, 2020.

[38] Keras.io, "Grad-CAM Class Activation Visualization, Code Example by Fchollet". Available at: https://keras.io/examples/vision/grad_cam/#lets-try-another-image, Accessed at 2020.