Sunflower Disease detection using Ensemble Deep Learning Models

Saksham Kathuria (saksham.kathuria0211@gmail.com)
IIITP: Indian Institute of Information Technology Pune
Noamaan Abdul Azeem
IIITP: Indian Institute of Information Technology Pune
Sanjeev Sharma
IIITP: Indian Institute of Information Technology Pune
Nitesh K Bharadwaj
IIITP: Indian Institute of Information Technology Pune

Research Article

Keywords: Deep Learning , Convolution Neural Network , Disease Classification , Transfer Learning , Ensemble Learning

Posted Date: November 29th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2490004/v1

Sunflower Disease detection using Ensemble Deep Learning Models

Saksham Kathuria* · Noamaan Abdul Azeem · Sanjeev Sharma · Nitesh K Bharadwaj

Received: date / Accepted: date

Abstract Plant diseases, such as fungi and parasites, disrupt or alter the plant's essential functions. In order to prevent potential economic losses from these plant diseases, early diagnosis of these diseases is crucial. The use of Deep Learning models for the detection and classification of plant diseases has been found to dramatically increase the speed of diagnosis while at the same time minimizing the amount of error involved. Taking advantage of a variety of tools and techniques, including transfer learning and ensemble learning, we have experimented with different deep-learning models and pre-existing architectures to find out which combination would best fit the data we gathered from the Sun Flower Fruits and Leaves dataset. Moreover, our best model has the ability to classify and detect sunflower disease with an accuracy of 97.91%, which is a considerable improvement over state-of-the-art models currently used for the classification and detection of sunflower disease.

Keywords Deep Learning · Convolution Neural Network · Disease Classification · Transfer Learning · Ensemble Learning

1 Introduction

It is obvious that agriculture plays a key role in the economic development of the country, and more importantly, it plays a critical role in the sustainability of our

Saksham Kathuria
Indian Institute of Information Technology, Pune
E-mail: sakshamkathuria20@cse.iiitp.ac.in

Noamaan Abdul Azeem
Indian Institute of Information Technology, Pune
E-mail: noamaana20@cse.iiitp.ac.in

Sanjeev Sharma
Indian Institute of Information Technology, Pune
E-mail: drsanjeev.sharma22@gmail.com

Nitesh K Bharadwaj
Indian Institute of Information Technology, Pune
E-mail: nitesh.bharadwaj@iiitp.ac.in

human population on the planet. Due to the exponential technological advancements in the agricultural sector, crop production has increased to feed the subtly growing population as a result of the sector's exponential technological advancements. Millions and millions of tons of cash crops are grown around the world each year, but a significant percentage of them get diseased as a result of fungi, bacteria, or parasites attacked on them. Detecting plant diseases at an early stage, can minimize the impact of these diseases on crop productivity while also increasing the quantity and quality of the crop yields [24]. Traditional techniques of disease detection based on optical inspection have considerable margins for error due to the complexity of crop management and phytopathological issues encountered today, which can lead to misclassification, wrong conclusions, and crop failure as a result.[22]

In recent years, machine learning has enhanced computer vision algorithms to incorporate the strengths of human biological vision. They are capable of correctly identifying features from images even without the help of humans. Consequently, its use has been extended to a variety of categorization and identification tasks [23]. In this paper, we extend Deep Learning to the classification of sunflower leaf diseases, using newly proposed image processing methods. Among the most important oilseed crops, sunflowers are commonly grown in moderate climates. In any crop, it is the sensitive leaves where sickness emerges first and spreads to other parts [21]. There are a variety of illnesses that can have a significant impact on sunflower productivity if they appear early in the growing season, including parasitic sickness and downy mildew [18].

There are a number of notable contributions that have been made in this proposed paper, including:

- This study represents the state-of-the-art in the field of classification of sunflower disease into four different classifications, Downy Mildew, Gray Mold, Leaf Scars and Fresh Leaf.
- The use of deep learning models to classify plant diseases may have already been provided as a solution in a number of previous studies, but this study has used different methodologies in order to come up with a model that has significant advantages over other existing models.
- Use of ensembling learning in order to improve the quality of results.
- In this study, substantial advances have been made in the application of deep learning architecture to automate disease detection at an industrial scale through the use of deep learning.

The rest of the paper is divided into the following sections: The literature review of the relevant study is covered in Section 2. The study of materials and procedures is covered in Section 3. Section 4 of the report discusses the tests and findings. Finally, section 5 provides the work's conclusion.

2 Literature Review

The need for appropriate methods for the detection and diagnosis of plant diseases has existed for a very long time in the world of agriculture. A variety of

approaches, methods and Machine Learning or Deep Learning models have been proposed in order to detect symptoms in diseased plants as early as possible.
Trong-Yen Lee et al. [20] proposed a highly efficient Convolutional Neural Network (CNN) which consists of four layers in its architecture, namely a convolution layer, a pooling layer, an activation function layer, and a full connection layer, for the detection of disease in potato leaves. As a result of using the Adam optimizer and the Softmax layer in their model, they were able to detect diseases with 99.53 percent accuracy.
Dongfang Wang et al. [43] proposed a method that is capable of identifying and classifying distinct types of diseases, for the purpose of mapping them separately, using a convolutional neural network model with trilinear convolutions and bilinear pooling. When tested in a controlled laboratory environment, they were able to achieve 99.99% and 99.7% accuracy on the test set that they were given.
M. Nandhini et al. [27] proposed a novel approach in early disease prediction and classification by a a sequential model consisting of CNN to extract potential features from the images fed to it in a sequential manner, and a RNN to learn from the temporal features in a sequential manner. It was found that the newly developed model Gated-Recurrent Convolutional Neural Network (G-RecConNN) considerably reduced the processing time for the preprocessing of the data.
The advancements in computer vision-based methods for segmenting of images for data preprocessing have been helpful in extracting features of objects from the data. Amar Kumar Dey et al. [9] extracted the patches of rotten leaves, as well as features related to them, using segmentation techniques, from an observed data set. After that, they were able to use the data to train an image processing algorithm to identify the characteristics of color that characterized rotten leaves based on their color features.
Mohammed Ali Al-windi et al. [25] proposed a robust method for the detection of wheat stem rust by converting RGB images to Hue Saturation Values that can be used to easily and quickly eliminate the background of the image. Additionally, they implemented feature selection in the preprocessing step, which improved detection speed and accuracy.
Aafreen Kazi et al. [16] researched CNN models and other computer vision technologies that can be used in the food industry, such as background removal and picture enhancement. In order to classify three different fruit varieties and their relative freshness, they proposed multiple traditional convolutional neural network designs. It was found that residual networks perform exceptionally well on the fruits dataset, producing test accuracy scores over 99%.
Amreen Abbas et al. [1] proposed a method that uses Conditional Generative Adversarial Networks (C-GAN) to generate synthetic images of tomato plants that can be used to detect disease. In order to train and test the DesenNet121 Model, they used both synthetic images as well as real images from the PlantVillage dataset, to classify diseased tomato leaves into different categories using transfer learning. This model accomplished the accuracy of 99.51%, 98.65% and 99.11% when images were classified in 5 classes, 7 classes, and 10 classes, respectively.
Xue Zhao et al. [45], DTL-SE-ResNet50 model was developed to determine vegetable diseases. To get new weight for this newly formed model, SE-ResNet50, they first pre-trained the SENet model with ImageNet, then trained it with AI Challenger 2018. By using dual-transfer learning, this SE-ResNet50 was trained on their own database to form the DTL-SE-ResNet50 model for identifying vegetable dis-

eases. In comparison with other convolutional neural networks, this model achieved better results in accuracy, precision, recall, and F1-Score.
Ashraf Darwish et al. [8] built automatic classification model by ensembling two pre-trained CNN models, namely VGG16 and VGG19 to distinguish whether the maize plant leaves are healthy or infected. Exponential learning rate decay schema was used to train every single model in ensemble, while hyperparameters of each one of them was optimized by the orthogonal learning particle swarm optimization (OLPSO) algorithm. This solved the problem of manually identifying the optimal hyperparameters with specific architecture in CNN models.
Mobeen Ahmad et al. [2] proposed the stepwise transfer learning approach. Using PlantVillage and Pepper Disease datasets, they trained their system using class-balancing methods. Since a number of deep learning based models and architecture exists in this space of image processing and transfer learning, many researches have performed the comparative study on different models.
Parul Sharma et al. [34] compared F-CNN and S-CNN models and showed how S-CNN model trained using segmented images performs better when it comes to automated detection of diseases in plants.
Edna Chebet Too et al. [42] presented a comparative study of different deep learning architecture including VGG 16, Inception V4, ResNet and DenseNets, based on fine tuning. PlantVillage dataset was used to train the models on healthy and diseased leaves from 14 different plants. From the experiment, they evaluated DenseNet as the best state-of-the-art deep convolutional neural network for image-based plant disease classification, with highest level of performance and test accuracy of 99.75%.

3 Materials and methods

The proposed methodology involves first pre-processing of data set using different techniques to maximize its use and to further prevent overfitting in the CNN models during training. After that, the models are trained on the dataset using transfer learning and further test their performance against a variety of criteria, as shown in the Fig.1, depicting the flow graph of the proposed methodology.

3.1 Dataset

For Convolution Neural Networks to develop intelligence for picture classification, appropriate data must be evaluated. We gathered two thousand one hundred and fifteen (2135) images from the Sun Flower Fruits and Leaves dataset for Sunflower Disease Classification using Machine Learning and Deep Learning in order to train our model This dataset contains 467 images, including diseases affecting sunflowers, such as Downy Mildew, Leaf Scars, and Gray Mold, along with Fresh Leaves.

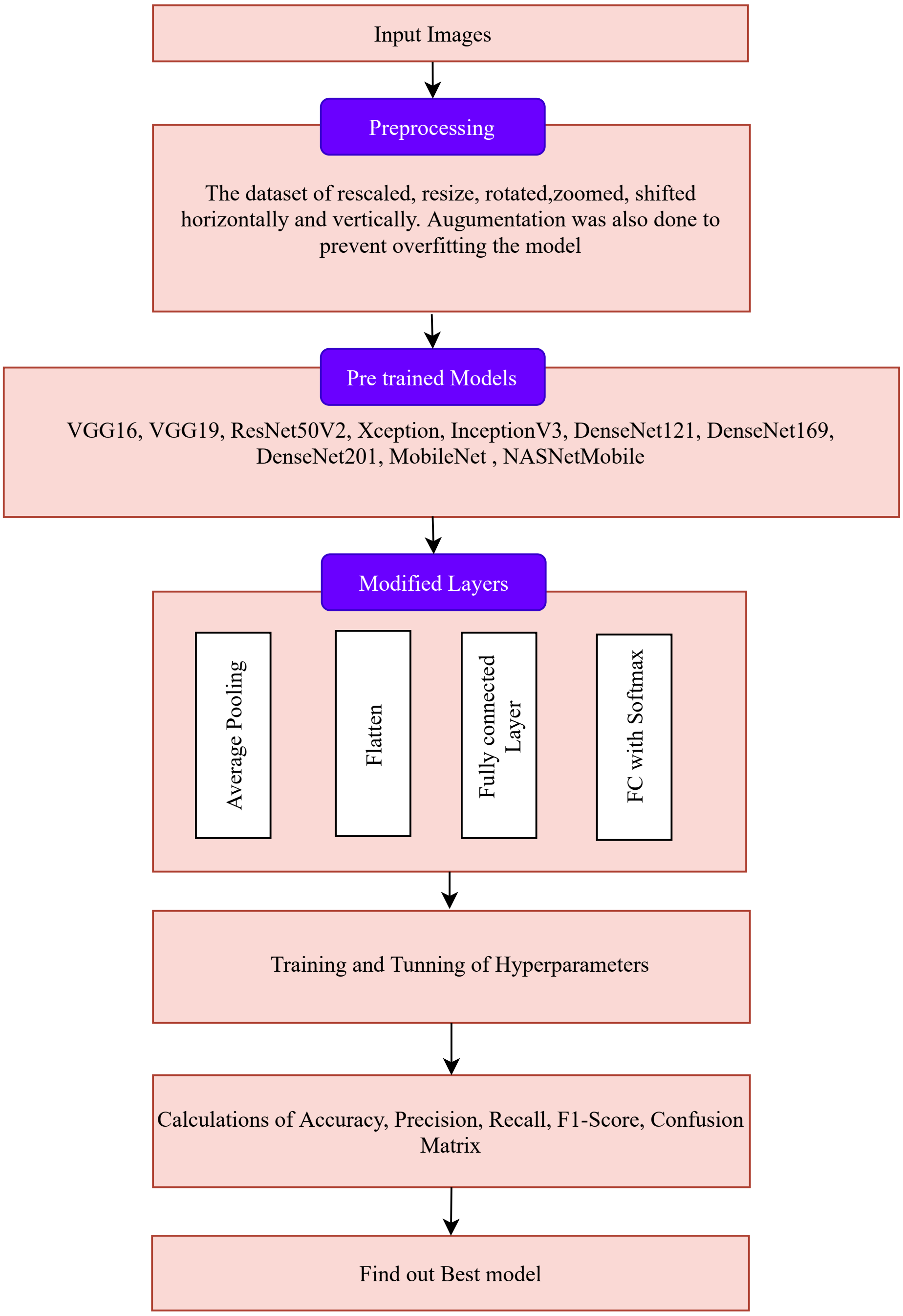


Fig. 1: Flow graph of the proposed methodology.

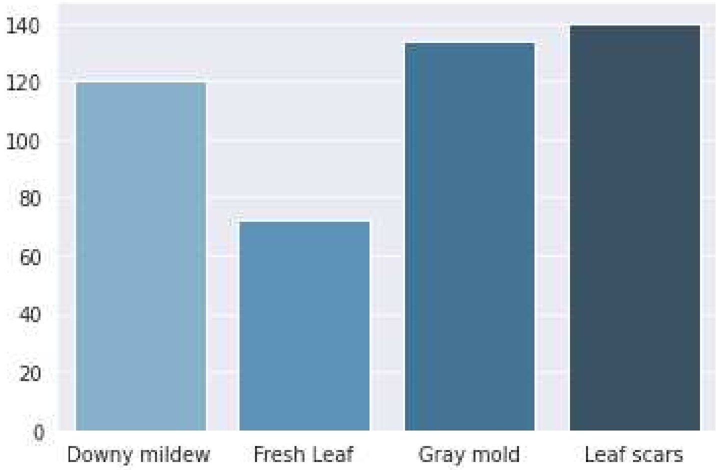


Fig. 2: Train - Data Distribution Graph

(b) Downy Mildew (d) Gray Mold (f) Leaf Scar (h) Fresh Leaf

Fig. 3: Images from different classes in the dataset

3.2 Data Preprocessing

Preprocessing an image is mainly used to increase the dataset size and the variety of images available to the model. Below are the techniques that are used for each epoch to modify an image from each class.

- Re-scaling of Image : RGB coefficients in the range of 0-255 original images are too high for our models to process, thus we scale with a 1./255 factor to target values between 0 and 1.
- Image Rotation : On an axis between 1° and 30°, pictures are rotated from right to left.
- Image Shifting : The image's content has been sheared, shifted vertically, and moved horizontally.
- Flipping of Image : By using a flip transformation, i.e., flipping pictures horizontally.

3.3 Transfer Learning

In order to train a deep learning model from scratch, a lot of resources are required [40]. To optimally arrange the randomly initialised weights of an untrained model, a substantial amount of data must be collected. Training data might be expensive or difficult to collect in some cases. This has led to pre-trained high performance models being increasingly used to extract weights and features[32]. By transferring knowledge from a similar domain, transfer learning is utilised to enhance a learner from certain domain[28]. The level of extraction varies based on the amount of data available and the similarity between the base and target domains. To achieve transfer learning, there are a variety of deep learning architectures available, ResNet, DenseNet, VGG etctera.

3.3.1 Visual Geometry Group (VGG) Models

Classical CNN model, VGG model is one of the benchmark for existing advance object-recognition models [36]. Instead of large number of hyper-parameters, utilisation of small filter 3 X 3 in VGG, allows VGG to have a large number of weight layers [44][4].

VGGNet is available in two types, VGG16 and VGG19, with 16 and 19 layers, respectively, in each. The VGG16 follows the configuration of 13 convolutional layers, 3 fully connected layers, a max-pooling layer that reduces volume size, and a fully connected layer with a softmax activation function as the final layer. VGG16 always employs the same padding and MaxPool layer of a 2x2 filter with stride 2 and concentrates on 3x3 filter convolution layers with stride 1. The model architecture of VGG19, in contrast, consists of 16 convolutional layers, 3 fully connected layers, 5 max-pooling layers for reducing the dimensions, and 1 softmax layer as an activation function discussed in [19]. The architecture of the VGG16 and VGG19 models is seen in Fig 4 and Fig 5, respectively.

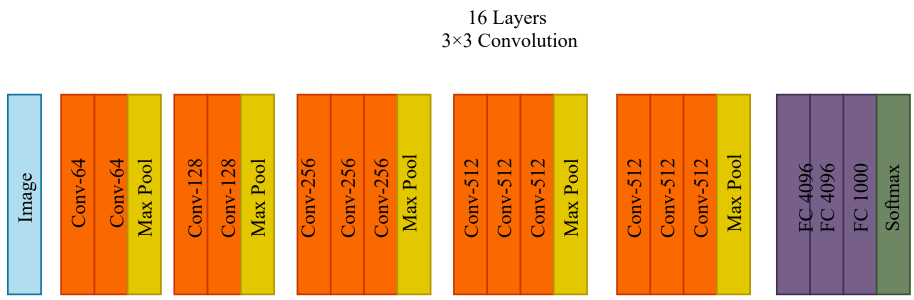


Fig. 4: VGG16 Model architecture [6]

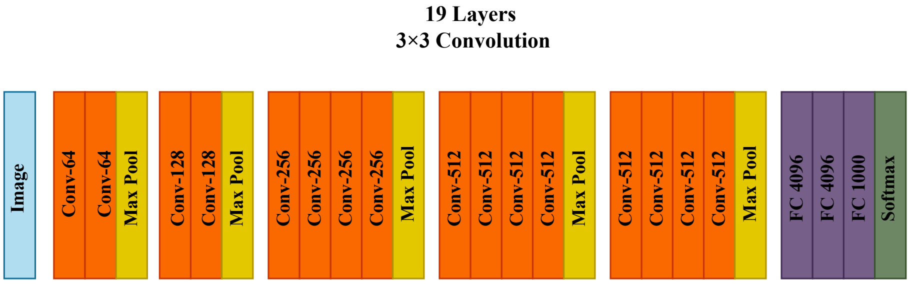


Fig. 5: VGG19 Model architecture [6]

3.3.2 InceptionV3

Inception model was introduced to address the issue of overfitting while increased the depth and width of the network without much unititlisation of computational resources[38]. It aligns the convolutional layers in parallel with different sized filter. Parallelisation reduced the depth of the inception model, while variety in convolutional filter sizes helped in extracting feature from image at varying scale[3]. Inception module also comprises concatenation layer, to combine multiple output from convo filters into single output for next inception module.

InceptionV3, upgraded version among inception models, employed auxiliary classifiers to regularise the gradient descent. Futhermore, implementation of factorisation in InceptionV3, significantly reduced the parameters and computational cost of the model[39]. Fig 6 shows the architecture of the Inception model.

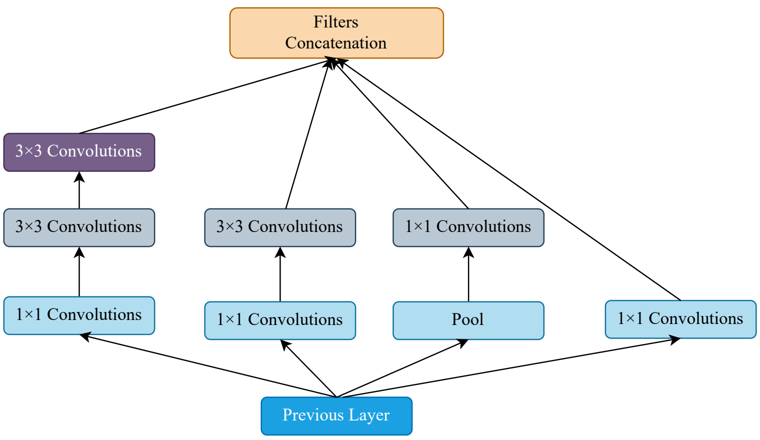


Fig. 6: Inception Module with the factorization of 5×5 convolutions into 3×3 ones [26]

3.3.3 Xception

Xception, extreme inception, is the first of its kind of convolution neural network, which differs from the standard convolution network in the way data computation is performed across the layers [7]. Xception architecture applies Depthwise Separable Convolution computation method rather than performing channelwise and spatial computation in one step. Depthwise convolution splits the input into different channels while using a single filter for each channel. Output from this convolution is linearly combined by Pointwise convolution, which forms the other half of the Depthwise Separable Convolution [10]. Fig 7 shows the architecture of the Xception model.

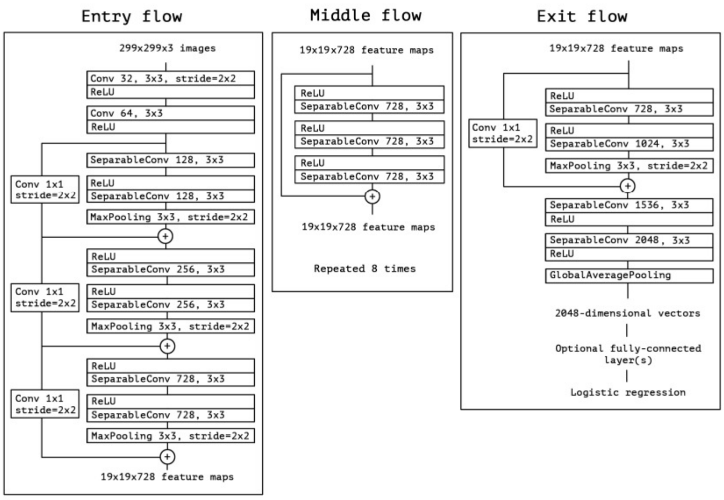


Fig. 7: Xception Model Architecture [7]

3.3.4 ResNet50V2

Residual Network (ResNet) is a prominent CNN-based architecture that was developed to provide a unique solution to the vanishing gradient problem, which was a fundamental obstacle for deep neural networks at the time [35]. With increasing depth in CNN architecture, error% increases, hence performance of the network degrades [41]. ResNet tackled this problem by using the concept of Residual blocks. These blocks contains skip connections which connects layer to each other by skipping few layers in between. Skip connections forms the fundamental art of residual blocks. Residual blocks make it simple for deeper layers to learn identity functions rather than unreferenced functions without compromising network performance [11]. Skip connections allow gradients to bypass those layers that have an effect on the network's performance. Fig. 8 shows the structure of skip connection in ResNet model.

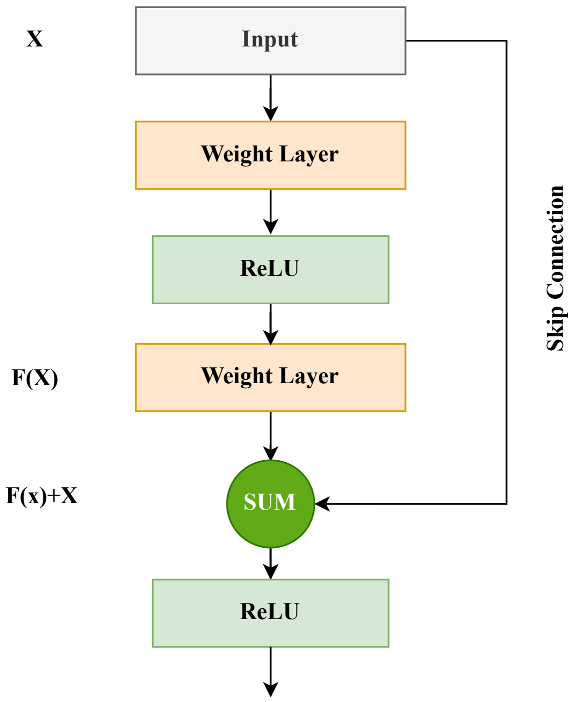


Fig. 8: A pictorial description of residual skip connections of the ResNet model[29]

3.3.5 DenseNet

DenseNet, short for Densely-Connected-Convolutional Network, was built on ResNet to develop to improve the knowledge transfer down the layers of network. CNN with larger number of layer and depth, faced the shortcoming of short memory, wherein the data, when feed forwarded from preceding layers to layers ahead gets vanished. This vanishing gradient in high-level neural network turned out to be great concern in efficient training and learning of the model. With Dense Blocks, DenseNet, accomplish Dense architecture, where all the layers are directly connected to each other in a feed-forward fashion [14]. Each layer takes in feature maps from all the previous layer, then concatenates them instead of additions and the pass on this collective knowledge to subsequent layers. Transition layers between dense blocks keep a note of keeping the dimensions of feature maps fixed through composite operations, including pooling layer, batch normalisation and activation funcitons [17].

Fig. 9 shows the layer architeture of densenet. Further, the proposed work uses DenseNet121, DenseNet169 and Denset201.

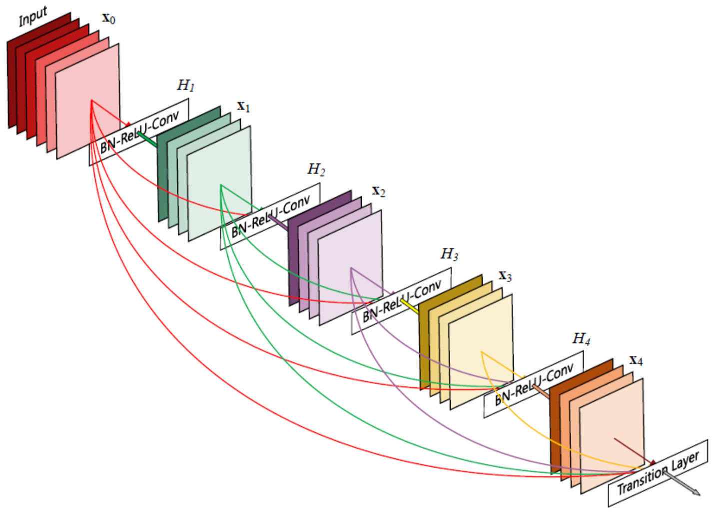


Fig. 9: A 5-layer dense block with a growth rate of k = 4. Each layer takes all preceding feature maps as input [15]

3.3.6 MobileNetV2

MobileNets are built on a simplified design that generates lightweight deep neural networks for mobile devices by using depth-wise separable convolutions [12]. Hyperparameter like width multiplayer and resolution wise multiplayer further reduces the computation cost of MobileNet. MobileNetV1 used dense blocks to significantly reduce the number of parameters generated while training of the network [30]. While MobileNetV2 utilised inverted residual framework, where skip connections are present between layers with reducing dimensions and nodes when compared to previous layers [33]. Fig. 10 shows the architecture of the Xception model.

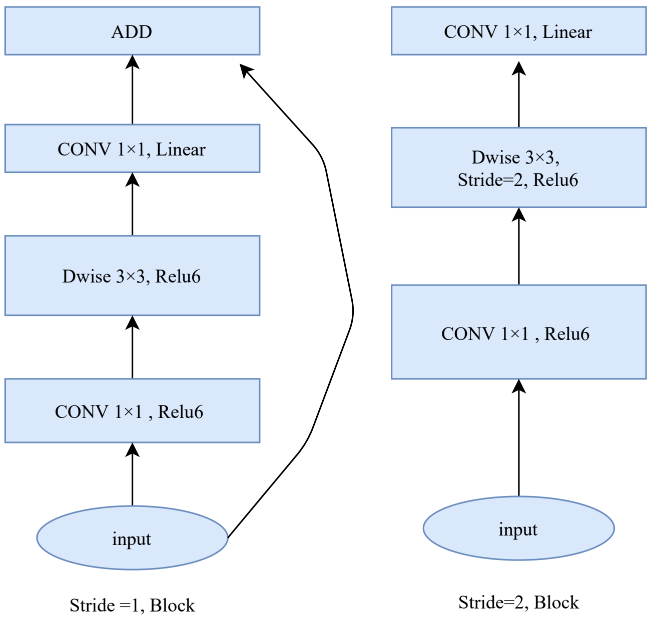


Fig. 10: MobileNet Model architecture [6]

3.4 Ensemble Approach

Ensembling approach is used for classification and regression tasks to produce novel models by fusing existing deep learning models [13]. These ensembled models outperform the different prediction models by overcoming low accuracy, high variance, while suppressing feature noises and biasness [5]. Among the several ensembling strategies are stacking, probability averaging, and majority voting [37].

Following separate training of multiple architectures, the architectures with the best validation accuracies and f1-score were chosen. These were then merged and ensembled to create a single best prediction model for training.

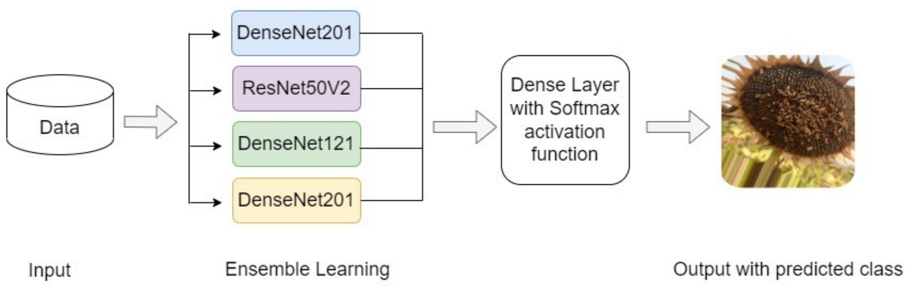


Fig. 11: Ensemble Learning

4 Experiments and Results

4.1 Hardware and software configuration

The operating system used is Windows 11 Home Single Language, with a 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz. With Python as the programming language, all models were trained on Intel(R) Iris(R) Xe Graphics and NVIDIA GeForce MX350 GPU, and TensorFlow with version 2.11.0 was required to get decent results.

4.2 Data Division Scheme

The datasets in this instance are divided into different categories such as the training datasets, validation datasets, and test datasets. The data used in the training process is 60 percent, while the data used in the validation and testing process is 20% each. Table 1 describes the division of images in different classes.

Table 1: Number of sample images for the experiments

Class Name	Total Images in a class	No. of Images in Training Set	No. of Images in Validation Set	No. of Images in Test Set
Disease Free	126	76	24	26
Phytopthora	114	68	22	24
Red rust	87	52	17	18
Scab	106	63	21	22
Styler and Root	94	55	19	20

4.3 Performance Evaluation Metrics

In order to determine the effectiveness and performance of the model based on the training data, a variety of assessment metrics are being used in the assessment process. First, a confusion matrix was used to represent how many predictions the model made based on the test dataset, which contained N classes, as a result of the model making predictions. Actual classes are on the x-axis and predicted classes are on the y-axis of the N*N confusion matrix. Using this matrix and following formulas other performances metrics were calculated:

$$Accuracy = \frac{NumberofCorrectPrediction}{TotalnumerofPredictions}$$

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + FalsePositive + FalseNegative + TrueNegative}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$F1\text{-}Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

4.4 Hyperparameter Optimization

For each model, these parameters were appropriately optimised. With Adam serving as an optimizer for its smooth properties, the learning rate employed was 0.00001. The Softmax function served as the last layer's activation function. The batch size used was 12 and 3, and these parameters were run using the transfer and ensemble learning approach over the course of 30 and 80 epochs, respectively, to improve accuracy.

4.5 Results

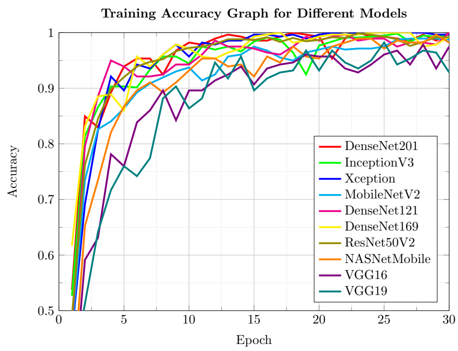
The outcomes of all models are presented in this section. We compare the findings for two data division schemes: 60/20/20 and 80/10/10. The percentage of data utilised for training, validation, and testing is represented by the first, second, and third components in this equation, respectively.

4.5.1 Results for 60/20/20 dataset division Scheme

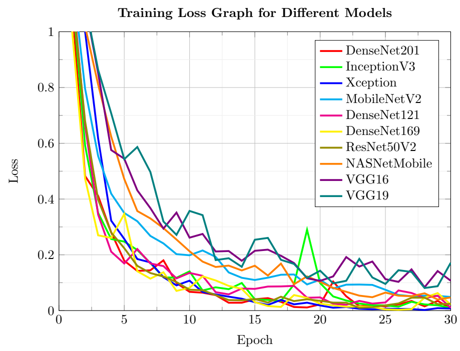
We experimented with ten different deep learning architectures. Table 2 describes the results of each model and from Fig 13 to Fig 22 depicts the confusion matrix of respective individual models on different classes. Fig. 12(b), 12(d), 12(f), 12(h) shows the graphs for individual models of training accuracy, training loss, validation accuracy and validation loss respectively.

Table 2: Individual model results comparison table for 60/20/20

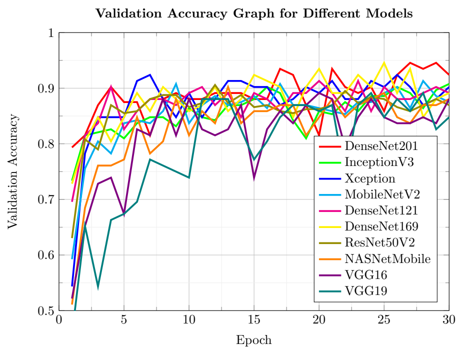
Model	Test Accuracy	Precision (MA)	Recall (MA)	F1-Score (MA)	Precision (WA)	Recall (WA)	F1-Score (WA)
NASNetMobile	86.32	0.88	0.87	0.87	0.87	0.86	0.86
MobileNet	85.26	0.86	0.86	0.86	0.86	0.85	0.85
InceptionV3	88.42	0.89	0.89	0.89	0.89	0.88	0.88
Resnet50V2	89.47	0.90	0.90	0.90	0.90	0.89	0.89
VGG19	83.16	0.85	0.84	0.83	0.84	0.83	0.83
VGG16	84.21	0.86	0.85	0.85	0.84	0.84	0.84
Xception	84.21	0.86	0.85	0.85	0.85	0.84	0.84
DenseNet121	88.42	0.89	0.89	0.89	0.89	0.88	0.88
DenseNet201	89.47	0.91	0.90	0.90	0.90	0.89	0.89
DenesNet169	87.37	0.89	0.88	0.88	0.88	0.87	0.87



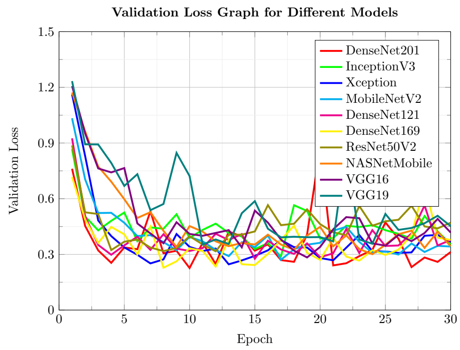
(b) Training Accuracy



(d) Training Loss



(f) Validation Accuracy



(h) Validation Loss

Fig. 12: Graphs for comparing the accuracy and loss of several individual models during training and validation for 60/20/20

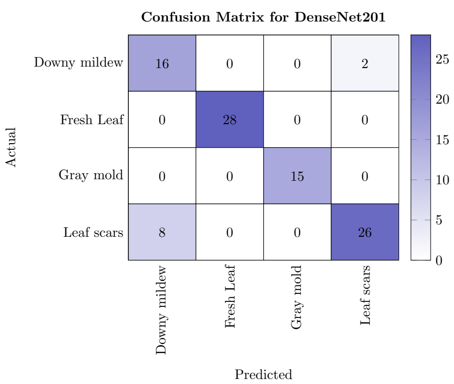


Fig. 13: Confusion Matrix for DenseNet201

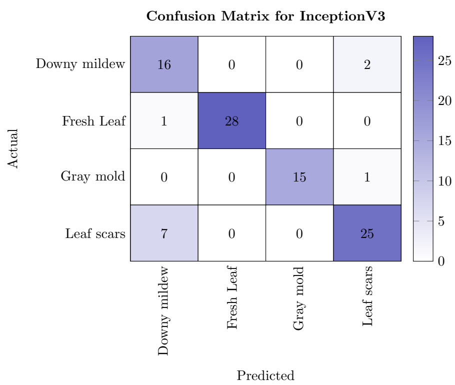


Fig. 14: Confusion Matrix for InceptionV3

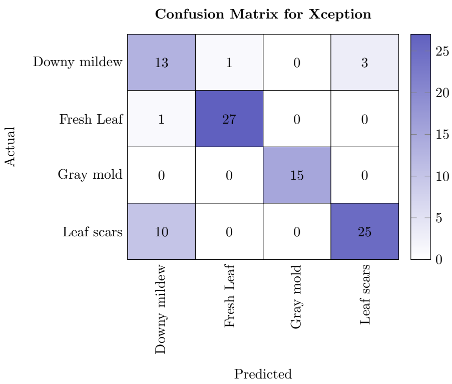


Fig. 15: Confusion Matrix for Xception

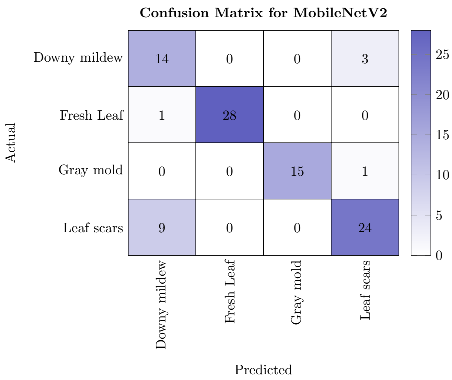


Fig. 16: Confusion Matrix for MobileNetV2

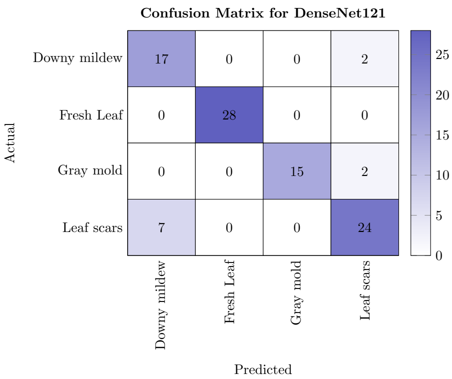


Fig. 17: Confusion Matrix for DenseNet121

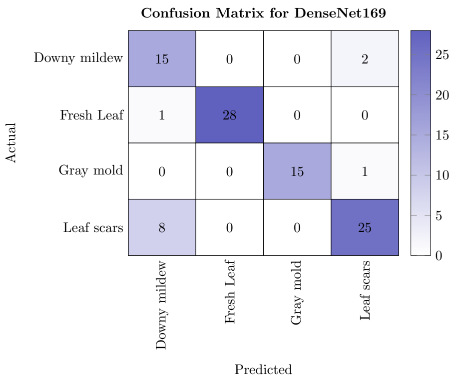


Fig. 18: Confusion Matrix for DenseNet169

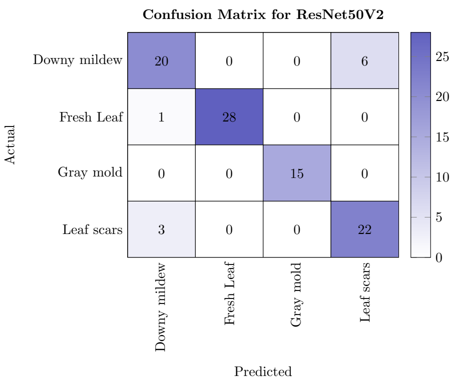


Fig. 19: Confusion Matrix for ResNet50V2

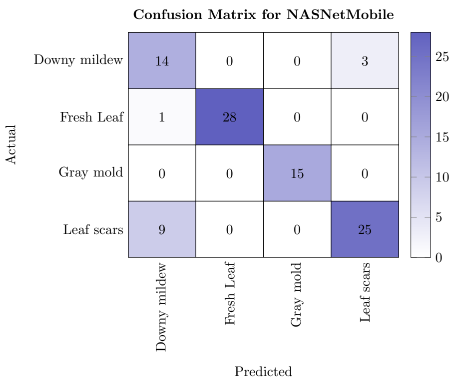


Fig. 20: Confusion Matrix for NASNetMobile

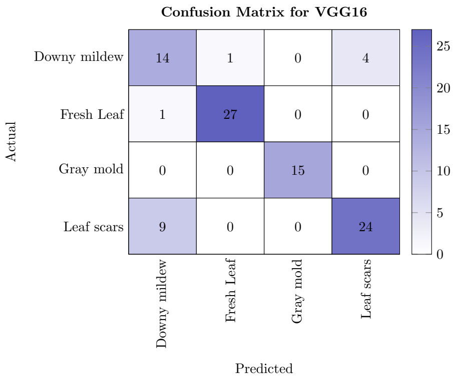


Fig. 21: Confusion Matrix for VGG16

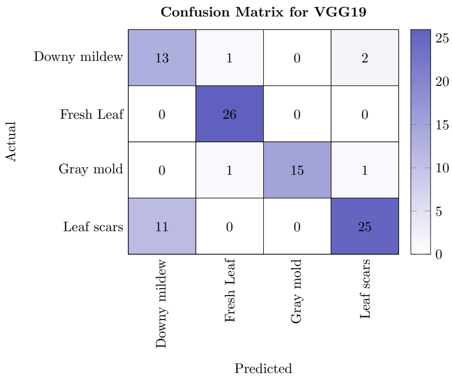


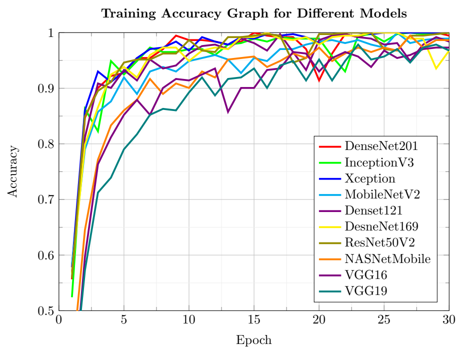
Fig. 22: Confusion Matrix for VGG19

4.5.2 Results for 80/10/10 dataset division Scheme

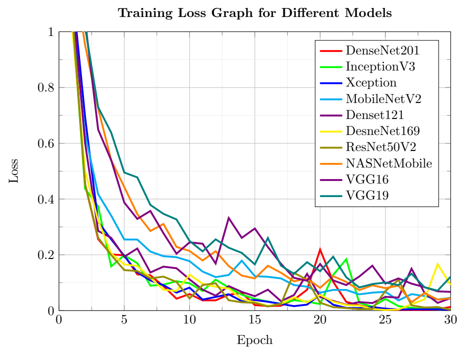
We experimented with ten different deep learning architectures. Table 3 describes the results of each model and from Fig 24 to Fig 33 depicts the confusion matrix of respective individual models on different classes Fig. 23(b), 23(d), 23(f), 23(h) shows the graphs for individual models of training accuracy, training loss, validation accuracy and validation loss respectively.

Table 3: Individual models result comparison table for 80/10/10

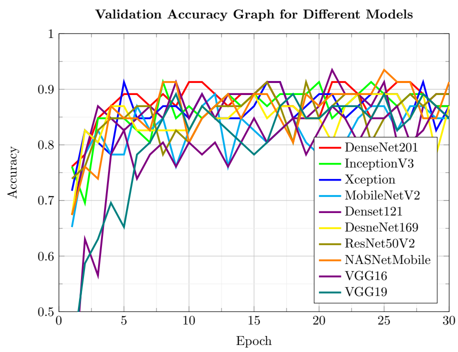
Model	Test Accuracy	Precision (Avg Macro)	Recall (Avg Macro)	F1-Score (Avg Macro)	Precision (WA Avg)	Recall (WA Avg)	F1-Score (WA Avg)
NASNetMobile	89.58	0.91	0.90	0.90	0.90	0.90	0.89
MobileNet	89.58	0.91	0.89	0.90	0.90	0.90	0.89
InceptionV3	87.50	0.89	0.88	0.88	0.88	0.88	0.87
Resnet50V2	89.58	0.91	0.90	0.90	0.90	0.90	0.89
VGG19	87.50	0.88	0.88	0.87	0.88	0.88	0.87
VGG16	89.58	0.91	0.90	0.90	0.90	0.90	0.89
Xception	89.58	0.91	0.90	0.90	0.90	0.90	0.89
DenseNet121	93.75	0.94	0.94	0.94	0.94	0.94	0.94
DenseNet201	93.75	0.94	0.94	0.94	0.94	0.94	0.94
DenseNet169	91.67	0.93	0.93	0.92	0.93	0.92	0.92



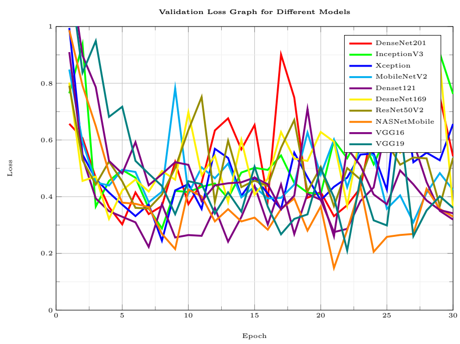
(b) Training Accuracy



(d) Training Loss



(f) Validation Accuracy



(h) Validation Loss

Fig. 23: Graphs for comparing the accuracy and loss of several individual models during training and validation for 80/10/10

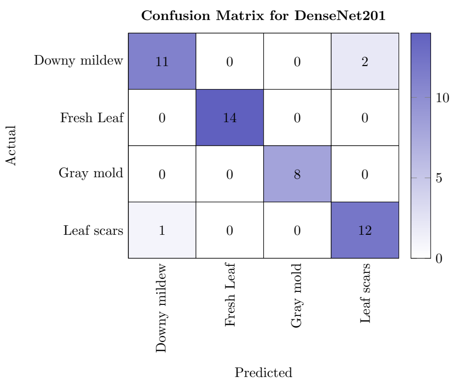


Fig. 24: Confusion Matrix for DenseNet201

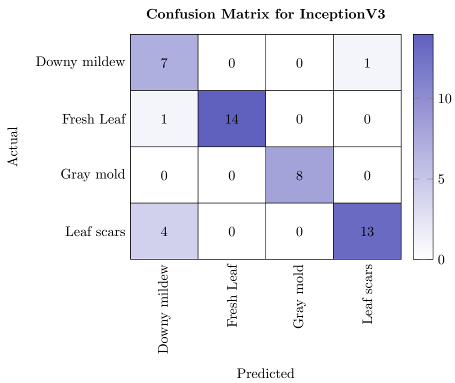


Fig. 25: Confusion Matrix for InceptionV3

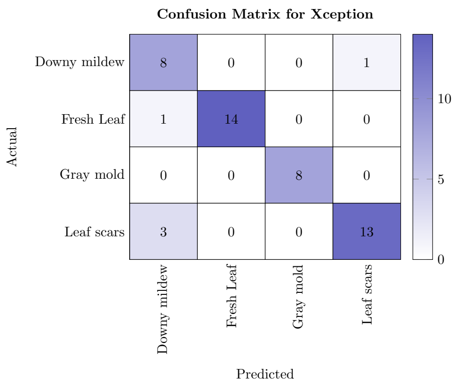


Fig. 26: Confusion Matrix for Xception

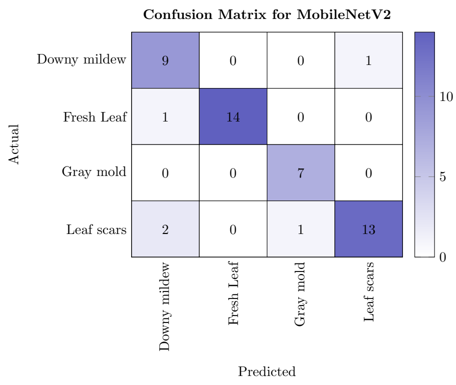


Fig. 27: Confusion Matrix for MobileNetV2

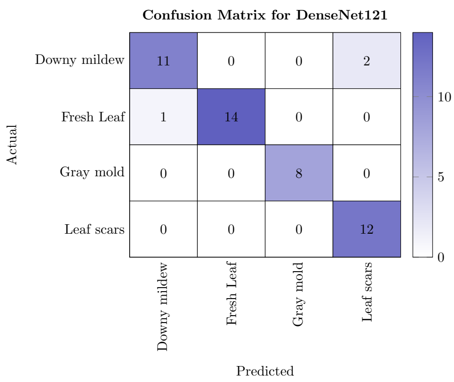


Fig. 28: Confusion Matrix for DenseNet121

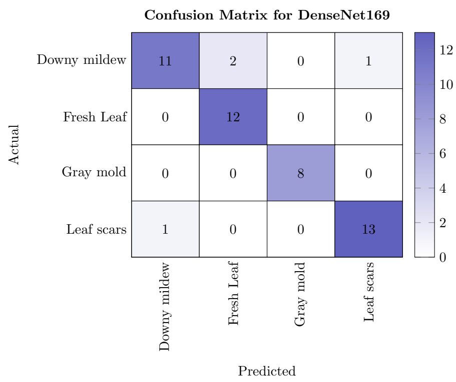


Fig. 29: Confusion Matrix for DenseNet169

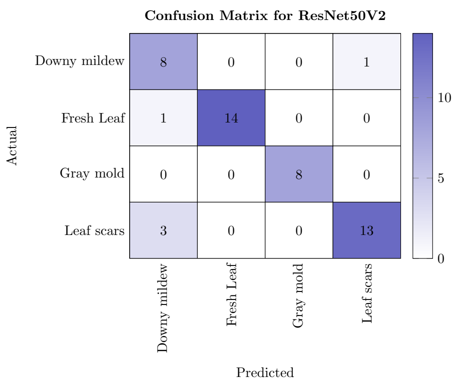


Fig. 30: Confusion Matrix for ResNet50V2

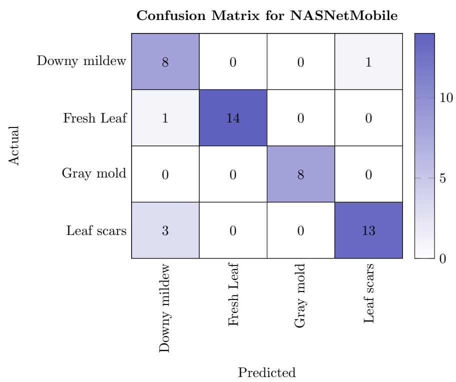


Fig. 31: Confusion Matrix for NASNetMobile

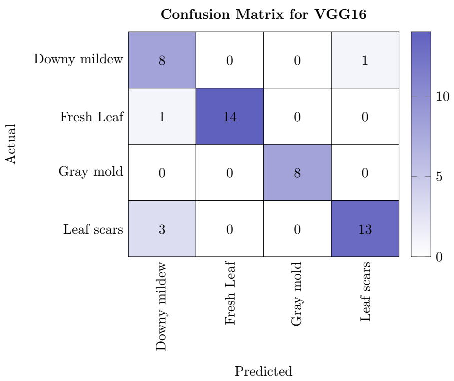


Fig. 32: Confusion Matrix for VGG16

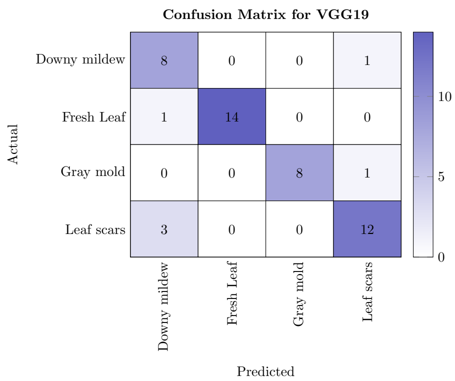


Fig. 33: Confusion Matrix for VGG19

4.5.3 Results of the Ensemble model

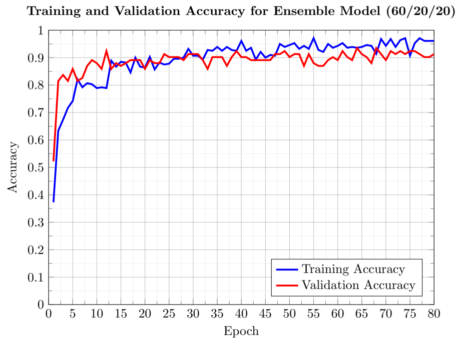
Over both dataset partitioning strategies, the outcomes of several architectures that were fine-tuned via transfer learning were compared. Ensemble learning was then used to train the top-performing models, DenseNet201, ResNet50V2, DenseNet121, and DenseNet169. After being trained on datasets split in 60/20/20 and 80/10/10 manner, this hybrid model was able to achieve accuracy levels of 91.98% and 97.91% respectively. Table 4 and 5 are the classification reports of the trained model over corresponding dataset.

Table 4: Ensemble learning parameters calculations for 60/20/20

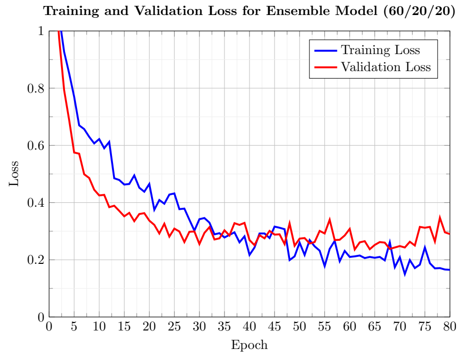
	Precision	Recall	F1-Score	Support
Downy mildew	0.863	0.791	0.826	24.000
Fresh Leaf	1.00	1.00	1.000	28.000
Gray mold	1.000	1.000	1.000	15.000
Leaf scars	0.833	0.892	0.862	28.000
accuracy	0.915	0.915	0.915	0.915
macro avg	0.924	0.921	0.922	95.000
weighted avg	0.916	0.915	0.915	95.000

Table 5: Ensemble learning parameters calculations for 80/10/10

	Precision	Recall	F1-Score	Support
Downy mildew	0.923	1.000	0.960	12.000
Fresh Leaf	1.000	1.000	1.000	14.000
Gray mold	1.000	1.000	1.000	8.000
Leaf scars	1.000	0.928	0.962	14.000
accuracy	0.979	0.979	0.979	0.979
macro avg	0.980	0.982	0.980	48.000
weighted avg	0.980	0.979	0.979	48.000

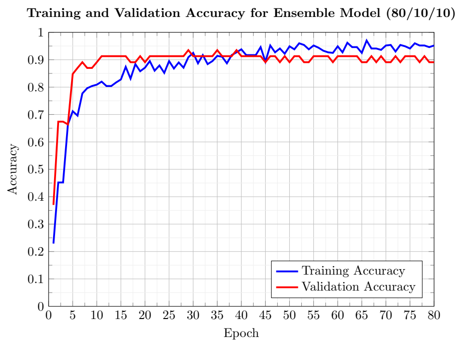


(b) Downy Mildew

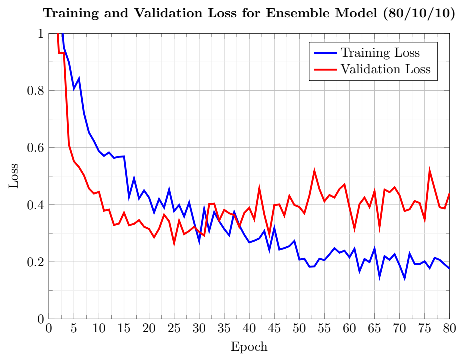


(d) Gray Mold

Fig. 34: Images from different classes in the dataset



(b) Downy Mildew



(d) Gray Mold

Fig. 35: Images from different classes in the dataset

4.6 Comparative study

This section shows the comparative study of the proposed methods with the existing work. Every paper and research done on our same dataset is compared and mentioned below. Our proposed best model obtained 97.91% validation accuracy over 80/10/10 dataset splitting technique, exceeding previous studies that applied Random Forest Classifier and Logistic Regression Classifier, which produced 90.68% and 88.85% accuracy over 80/20 dataset splitting technique [31], respectively. Table 6 shows the comparative study in a table format for simple comprehension, including Precision and Recall / Sensitivity as performance metrics.

Table 6: Comparative Study Table

	Classes	Precision	Recall/Sensitivity
Random Forest Classifier [31]	Downy mildew	73.17%	78.26%
	Gray Mold	73.68%	70.70%
	Leaf Scar	77.78%	78.87%
	Fresh Leaf	90.07%	79.87%
Logistic Regression Classifier [31]	Downy mildew	66.67%	69.57%
	Gray Mold	60.75%	65.66%
	Leaf Scar	70.29%	68.31%
	Fresh Leaf	85.94%	69.18%
Ensemble 60/20/20 Approach	Downy mildew	86.36%	79.16%
	Gray Mold	100%	100%
	Leaf Scar	83.33%	89.28%
	Fresh Leaf	100%	100%
Ensemble 80/10/10 Approach	Downy mildew	92.30%	100%
	Gray Mold	100%	100%
	Leaf Scar	100%	92.85%
	Fresh Leaf	100%	100%

4.7 Discussion

The major objective of this study is to develop models that perform better than existing state-of-the-art models in categorising images of sunflower disease over the choosen dataset. Transfer learning and ensemble learning were implemented to train each deep learning model that was used in this research.

Firstly, various experimental studies were carried out on original and augmented dataset employing transfer learning using 10 different architectures. Performance comparision of these architecture were given in Tables 2 and 3 over 2 different dataset splitting schemes, 60/20/20 and 80/10/10, respectively. The best performing models, models DenseNet201, ResNet50V2, DenseNet121 and DenseNet169 were then choosen to be collectively trained using ensemble learning. Table shows that this hybrid model had the maximum accuracy of 97.91%, which is much higher than the other state-of-the-art models, as shown in 4.6.

5 Conclusions and Future Scope

Recent developments in Deep Learning and Computer Vision have demonstrated considerable advancements in a broad range of application areas, including pattern recognition and picture processing. In this study, we first presented a thorough review of recent developments in image classification, and then we assessed a number of deep learning-based approaches and models. We used a total of 2135 images, including photos taken at the BARI sunflower garden and the remaining augmented photos. To curate our models, two major methodologies were put out, one of which aimed for creating models employing transfer learning from 10 distinct deep learning architectures that had already been developed. The best performing individual models were then pooled using the ensemble learning approach to collectively predict the results for the classification of sunflower leaf images over two different dataset division schemes of 60/20/20 and 80/10/10 representing training, validation, and testing datasets, respectively. The proposed models were compared using different performance evaluation metrics such as accuracy, precision, recall, confusion matrix and F1-Score. Our highest performing model achieved 97.91% utilising the ensemble learning approach on an 80/10/10 dataset partition scheme. It is considerably outperforming in respect to existing equivalent state-of-the-art methods and models.

As potential future work, we intend to employ big data to analyse these photos and expand the variety of plants and disease classes. The disease severity of the sick leaf image class may also be determined by designing a novel, effective optimised model.

6 Compliance with Ethical Standards

Conficts of interest The authors declare that they have no confict of interest.
Funding No funding was received to assist with the preparation of this manuscript.
Data Availability The paper uses the publicly available dataset for Sun Flower Fruits and Leaves. The dataset is openly avvailable at https://data.mendeley.com/datasets/b83hmrzth8/1
Human Participants This article does not contain any studies involving human participants performed by any of the authors.

References

1. Abbas, A., Jain, S., Gour, M., Vankudothu, S.: Tomato plant disease detection using transfer learning with c-gan synthetic images. Computers and Electronics in Agriculture **187**, 106279 (2021). DOI https://doi.org/10.1016/j.compag.2021.106279. URL https://www.sciencedirect.com/science/article/pii/S0168169921002969
2. Ahmad, M., Abdullah, M., Moon, H., Han, D.: Plant disease detection in imbalanced datasets using efficient convolutional neural networks with stepwise transfer learning. IEEE Access **9**, 140565–140580 (2021)
3. Alake, R.: Deep learning: Understanding the inception module. https://towardsdatascience.com/deep-learning-understand-the-inception-module-56146866e652
4. Ümit Atila, Uçar, M., Akyol, K., Uçar, E.: Plant leaf disease classification using efficient-net deep learning model. Ecological Informatics **61**, 101182 (2021). DOI https://doi.org/10.1016/j.ecoinf.2020.101182. URL https://www.sciencedirect.com/science/article/pii/S1574954120301321
5. Ayuya, C.: Ensemble learning on bias and variance. https://www.section.io/engineering-education/ensemble-bias-var/
6. Bansal, M., Kumar, M., Sachdeva, M., Mittal, A.: Transfer learning for image classification using vgg19: Caltech-101 image data set. Journal of Ambient Intelligence and Humanized Computing pp. 1–12 (2021)
7. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1800–1807 (2017). DOI 10.1109/CVPR.2017.195
8. Darwish, A., Ezzat, D., Hassanien, A.E.: An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis. Swarm and Evolutionary Computation **52**, 100616 (2020). DOI https://doi.org/10.1016/j.swevo.2019.100616. URL https://www.sciencedirect.com/science/article/pii/S2210650219305462

9. Dey, A.K., Sharma, M., Meshram, M.: Image processing based leaf rot disease, detection of betel vine (piper betlel.). Procedia Computer Science **85**, 748–754 (2016). DOI https://doi.org/10.1016/j.procs.2016.05.262. URL https://www.sciencedirect.com/science/article/pii/S1877050916306123. International Conference on Computational Modelling and Security (CMS 2016)
10. Guo, Y., Li, Y., Feris, R.S., Wang, L., Simunic, T.: Depthwise convolution is all you need for learning multiple visual domains. In: AAAI Conference on Artificial Intelligence (2019)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016). DOI 10.1109/CVPR.2016.90
12. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
13. Huang, F., Xie, G., Xiao, R.: Research on ensemble learning. In: 2009 International Conference on Artificial Intelligence and Computational Intelligence, vol. 3, pp. 249–252 (2009). DOI 10.1109/AICI.2009.235
14. Huang, G., Liu, Z., Maaten, L.V.D., Weinberger, K.Q.: Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2261–2269. IEEE Computer Society, Los Alamitos, CA, USA (2017). DOI 10.1109/CVPR.2017.243. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.243
15. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700–4708 (2017)
16. Kazi, A., Panda, S.: Determining the freshness of fruits in the food industry by image classification using transfer learning. Multimedia Tools and Applications **81** (2022). DOI 10.1007/s11042-022-12150-5
17. Khanna, M.: Paper review: Densenet -densely connected convolutional networks. https://towardsdatascience.com/paper-review-densenet-densely-connected-convolutional-networks-acf9065dfefb
18. Kofoet, A., Fischer, K.: Evaluation of plant resistance improvers to control peronospora destructor, p. parasitica, bremia lactucae and pseudoperonospora cubensis. Journal of Plant Diseases and Protection **114**, 54–61 (2007). DOI 10.1007/BF03356204
19. Kumar, K.: Transfer learning with vgg16 and vgg19, the simpler way! https://koushik1102.medium.com/transfer-learning-with-vgg16-and-vgg19-the-simpler-way-ad4eec1e2997
20. Lee, T., Lin, I., Yu, J., Yang, J., Chang, Y.: High efficiency disease detection for potato leaf with convolutional neural network (2021). DOI https://doi.org/10.1007/s42979-021-00691-9
21. Li, L., Zhang, S., Wang, B.: Plant disease detection and classification by deep learning—a review. IEEE Access **9**, 56683–56698 (2021). DOI 10.1109/ACCESS.2021.3069646
22. M. Akila, P.D.: Detection and classification of plant leaf diseases by using deep learning algorithm. International Journal of Engineering Research Technology (IJERT) (2018). URL https://www.ijert.org/research/detection-and-classification-of-plant-leaf-diseases-by-using-deep-learning-algorithm-IJERTCONV6IS07082.pdf
23. Mahadevkar, S.V., Khemani, B., Patil, S., Kotecha, K., Vora, D.R., Abraham, A., Gabralla, L.A.: A review on machine learning styles in computer vision—techniques and future directions. IEEE Access **10**, 107293–107329 (2022). DOI 10.1109/ACCESS.2022.3209825
24. Math, R., Dharwadkar, N.: Early detection and identification of grape diseases using convolutional neural networks. Journal of Plant Diseases and Protection **129** (2022). DOI 10.1007/s41348-022-00589-5
25. Mohammed Ali Al-windi, B.K., Abbas, A.H., Shakir Mahmood, M.: Wheat stem rust leaf disease detection using image processing. Materials Today: Proceedings (2021). DOI https://doi.org/10.1016/j.matpr.2021.07.251. URL https://www.sciencedirect.com/science/article/pii/S2214785321051105
26. Moses, K., Miglani, A., Kankar, P.K., et al.: Deep cnn-based damage classification of milled rice grains using a high-magnification image dataset. Computers and Electronics in Agriculture **195**, 106811 (2022)
27. Nandhini, M., Kala, K., Thangadarshini, M., Madhusudhana Verma, S.: Deep learning model of sequential image classifier for crop disease detection in plantain tree cultivation. Computers and Electronics in Agriculture **197**, 106915 (2022). DOI https:

//doi.org/10.1016/j.compag.2022.106915. URL https://www.sciencedirect.com/science/article/pii/S0168169922002320
28. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering **22**(10), 1345–1359 (2010). DOI 10.1109/TKDE.2009.191
29. Paul, A., Pramanik, R., Malakar, S., Sarkar, R.: An ensemble of deep transfer learning models for handwritten music symbol recognition. Neural Computing and Applications **34**(13), 10409–10427 (2022)
30. Pujara, A.: Image classification with mobilenet. https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470
31. Rajbongshi, A., Biswas, A.A., Biswas, J., Shakil, R., Akter, B., Barman, M.R.: Sunflower diseases recognition using computer vision-based approach. In: 2021 IEEE 9th Region 10 Humanitarian Technology Conference (R10-HTC), pp. 1–5 (2021). DOI 10.1109/R10-HTC53172.2021.9641588
32. Ribani, R., Marengoni, M.: A survey of transfer learning for convolutional neural networks. In: 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T), pp. 47–57 (2019). DOI 10.1109/SIBGRAPI-T.2019.00010
33. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018). DOI 10.1109/CVPR.2018.00474
34. Sharma, P., Berwal, Y.P.S., Ghai, W.: Performance analysis of deep learning cnn models for disease detection in plants using image segmentation. Information Processing in Agriculture **7**(4), 566–574 (2020). DOI https://doi.org/10.1016/j.inpa.2019.11.001. URL https://www.sciencedirect.com/science/article/pii/S2214317319301957
35. Shi, G.: Implementing a resnet model from scratch. https://towardsdatascience.com/implementing-a-resnet-model-from-scratch-971be7193718
36. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2015)
37. Singh, A.: A comprehensive guide to ensemble learning (with python codes). https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/
38. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9 (2015). DOI 10.1109/CVPR.2015.7298594
39. T, A.N.: Inception v3 model architecture. https://iq.opengenus.org/inception-v3-model-architecture
40. .T, R., Anandamurugan, S., Kaliappan, V.: Automated tomato leaf disease classification using transfer learning-based deep convolution neural network. Journal of Plant Diseases and Protection **128** (2020). DOI 10.1007/s41348-020-00403-0
41. Team, G.L.: Introduction to resnet or residual network. https://www.mygreatlearning.com/blog/resnet/
42. Too, E.C., Yujian, L., Njuki, S., Yingchun, L.: A comparative study of fine-tuning deep learning models for plant disease identification. Computers and Electronics in Agriculture **161**, 272–279 (2019). DOI https://doi.org/10.1016/j.compag.2018.03.032. URL https://www.sciencedirect.com/science/article/pii/S0168169917313303. BigData and DSS in Agriculture
43. Wang, D., Wang, J., Li, W., Guan, P.: T-cnn: Trilinear convolutional neural networks model for visual detection of plant diseases. Computers and Electronics in Agriculture **190**, 106468 (2021). DOI https://doi.org/10.1016/j.compag.2021.106468. URL https://www.sciencedirect.com/science/article/pii/S0168169921004853
44. Wei, J.: Vgg neural networks: The next step after alexnet. https://towardsdatascience.com/vgg-neural-networks-the-next-step-after-alexnet-3f91fa9ffe2c
45. Zhao, X., Li, K., Li, Y., Ma, J., Zhang, L.: Identification method of vegetable diseases based on transfer learning and attention mechanism. Computers and Electronics in Agriculture **193**, 106703 (2022). DOI https://doi.org/10.1016/j.compag.2022.106703. URL https://www.sciencedirect.com/science/article/pii/S0168169922000205