IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 May 2023, accepted 31 May 2023, date of publication 9 June 2023, date of current version 30 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3284680

RESEARCH ARTICLE

Pathogen-Based Classification of Plant Diseases: A Deep Transfer Learning Approach for Intelligent Support Systems

K. P. ASHA RANI AND S. GOWRISHANKAR
Department of Computer Science and Engineering, Dr. Ambedkar Institute of Technology, Bengaluru, Karnataka 560056, India
Visvesvaraya Technological University (VTU), Belagavi, Karnataka 590018, India

Corresponding author: K. P. Asha Rani (asharanikp.cs@drait.edu.in)

ABSTRACT The national economy's key pillar, agriculture has a significant influence on society. Plant health monitoring and disease detection are essential for sustainable agriculture. To protect plants against pathogen damage, farmers must be able to detect an infection prior to its obviousness. Effective plant disease detection technique can greatly lessen the use of toxic chemicals thereby aiding a better environment. For diseases to be managed effectively, plant pathogens must be accurately detected. The pathogens that cause plant diseases include bacteria, fungi, viruses, oomycetes, nematodes, phytoplasmas, protozoa, and parasitic plants. In this paper pathogen-based plant disease detection is done. An automated plant disease detection and its classification are done along with identifying the pathogen responsible for it using keras transfer learning models. This is done by considering Agri-ImageNet dataset as well as images of leaves, bulb, and flowers of sunflower and cauliflower captured in a natural realistic environment. This dataset overcomes the drawback of PlantVillage dataset in which images are captured in homogeneous backgrounds and controlled settings. These problems can be solved by reusing knowledge representations through deep transfer learning. Main objective of this paper is to explore and analyze all the deep transfer learning models, to identify which model is best suited for plant disease dataset. This work has been carried out by using 38 deep transfer learning models to obtain best classification accuracy. EfficientNetV2B2 and EfficientNetV2B3 models' give highest accuracy in comparison with all other deep transfer learning models for sunflower, cauliflower and Agri-ImageNet datasets. Classification report is generated from the best deep transfer learning model.

INDEX TERMS Plant disease detection, pathogens, transfer learning, deep learning, deep transfer learning, pretrained models, Xception, VGG_16, VGG_19, ResNet models, InceptionV3, InceptionResNetV2, MobileNet models, DenseNet121 models, NasNet models, EfficientNet models, EfficientNetV2 models, ConvNeXt models.

I. INTRODUCTION

Traditional methods of identifying plant disease include visually examining the plant, effects of pathogens can only be seen after the plant has undergone a significant amount of damage. There exists a lot of potential for automated plant disease detection. The main causes of plant diseases are inadequate nutrition, microbial attack, rodent infestation, adverse environmental conditions [1].

The associate editor coordinating the review of this manuscript and approving it for publication was Seifedine Kadry.

Plants are more susceptible to diseases due to a huge quantity of pathogens in their environment [2]. Any physiological or structural anomaly that is brought on by a living organism is considered a plant disease. Plant diseases are caused by plant pathogens or environmental factors. Infection by pathogens in plants is one of the main causes of decreased agricultural yields on a global scale. Plants are attacked by various pathogen groups individually, or by more than one pathogen, resulting in a more severe illness. Plant diseases are a hazard to food security because they can damage crops, reducing food supply and increasing food costs.

In order to achieve and sustain food security and sources of income for a rising world, it is more important to protect plants from diseases [3].

A. PLANT PATHOLOGY

Plant pathology deals with the study of plant diseases. When plants are exposed to unfavorable climatic conditions and parasitic microbes that cause disease, their chances of survival are attempted to be increased through the study of plant diseases [4]. Along with some parasitic plants and algae, the bulk of plant pathogens are microbes like bacteria, viruses, nematodes, fungi, and protozoa [5]. Through the release of enzymes, poisons, and other compounds, they cause damage to plants, eat nutrients, and spread disease. Plant diseases result in financial losses, failed crops, famine, as well as the extinction of entire plant species due to food poisoning. By absorbing resources from the host cells for their own growth, multiplication, and dissemination, plant pathogens penetrate and wreak disease in plant tissues [6]. Once an infection has penetrated a plant, there is a disruption in the structural stability and functioning of its cells and tissues by the production of enzymes, poisons, growth regulators, and other active compounds [7]. Pathogen indications include dead tissue patches on leaves, stems, fruit, and roots are treated as blights, which cause leaves and young shoots to suddenly die. Depending on the sort of disease that is causing them, the severity of the symptoms might range from minor to the death of the entire plant [8]. On the basis of various number of factors, plant diseases are categorized as shown in Figure 1.

B. ARTIFICIAL INTELLIGENCE (AI), MACHINE LEARNING (ML), DEEP LEARNING (DL), TRANSFER LEARNING (TL), AND DEEP TRANSFER LEARNING (DTL) IN PLANT DISEASE DETECTION

Farmers overuse herbicides, insecticides, and pesticides to cure plant diseases because they lack appropriate knowledge of the diseases that harm crops. Hence, it is imperative that diseases be diagnosed using the ever-evolving technology, so that the proper quantity and kind of chemicals may be administered [9]. The issues faced by different industries, including the agricultural sector, such as crop harvesting, irrigation, soil content sensitivity, crop monitoring, weed, harvest, and establishment, are managed by Artificial Intelligence (AI)-based technology, which also helps to increase efficiency across all sectors. In farms, AI technology aids in the diagnosis of pests, illnesses, and malnutrition. AI sensors can also detect and identify weeds. The methods used to classify diseases, segment afflicted areas, and identify diseases are employed [10]. Deep Learning has significantly increased the identification accuracy of image classification and object detection systems in recent years. The problem can be effectively and practically solved by AI, which also gave rise to Machine Learning (ML), Deep Learning (DL), Transfer Learning (TL), and Deep Transfer Learning (DTL) as shown in Figure 2.

Artificial Intelligence (AI) is a broad field that enables machines to think independently of human instruction. Though often used in conjunction with AI, ML is a subset of AI. ML refers to systems that continuously learn without human involvement. ML is used in DL, which is used with big datasets [11]. History of DL can be categorized into 3 developmental stages. The first version of neural network (1943–1969) is a linear model that was developed in 1943 and can only handle linear classification issues. The multilayer perceptron was compatible with the second generation of Neural Network Back Propagation (BP) (1986–1998). The second rise in neural networks was brought on by this technique. Nevertheless, a gradient vanishing issue with the BP method was identified in 1991. Third-generation neural networks' is DL (2006-present). In 2012, AlexNet, a Deep Learning model, gained significant recognition when it won the "The ImageNet Large Scale Visual Recognition Challenge (ILSVRC)" [12]. Transfer learning, an advanced DL technique, has demonstrated its efficacy for tasks relating to identification and classification. Deep Transfer Learning (DTL) is a technique used in Deep Learning, a subset of machine learning that involves reusing knowledge learned from one data source or task to train a model for a different data source or task. Deep Transfer Learning (DTL) aims to reduce this dependency on task-specific data by leveraging knowledge learned from one source data or task to train a model for a different task or data source. By reusing learned information from a pre-trained model or adapting representations from one domain to another, DTL can potentially mitigate the need for extensive task-specific labeled data for every new task or data source [13]. Deep Learning techniques are often utilized in pattern recognition and machine vision tasks. In order to identify plant diseases, researchers presented several Deep Learning models [14]. DTL can be used for plant disease detection with few numbers of parameters to achieve high accuracy [15]. On the whole Transfer Learning in Intelligent Support Systems can lead to several advantages, including reduced training time, improved performance with limited labeled data, and the ability to benefit from existing knowledge and expertise.

The key contributions of the paper are as follows:

1. Comparison and evaluation of 38 transfer learning model techniques by using sunflower and cauliflower dataset to analyze the efficacy for detecting plant diseases and the pathogens responsible.
2. Identification of best transfer learning model which can successfully accomplish the goal of classifying plant diseases using pre-trained ImageNet models.
3. Proposed system also predicts pathogen type for a specific disease which helps in taking precautionary measures accordingly.
4. The visualization of Feature maps and comparison between EfficientNetV2B2 and InceptionResNetV2

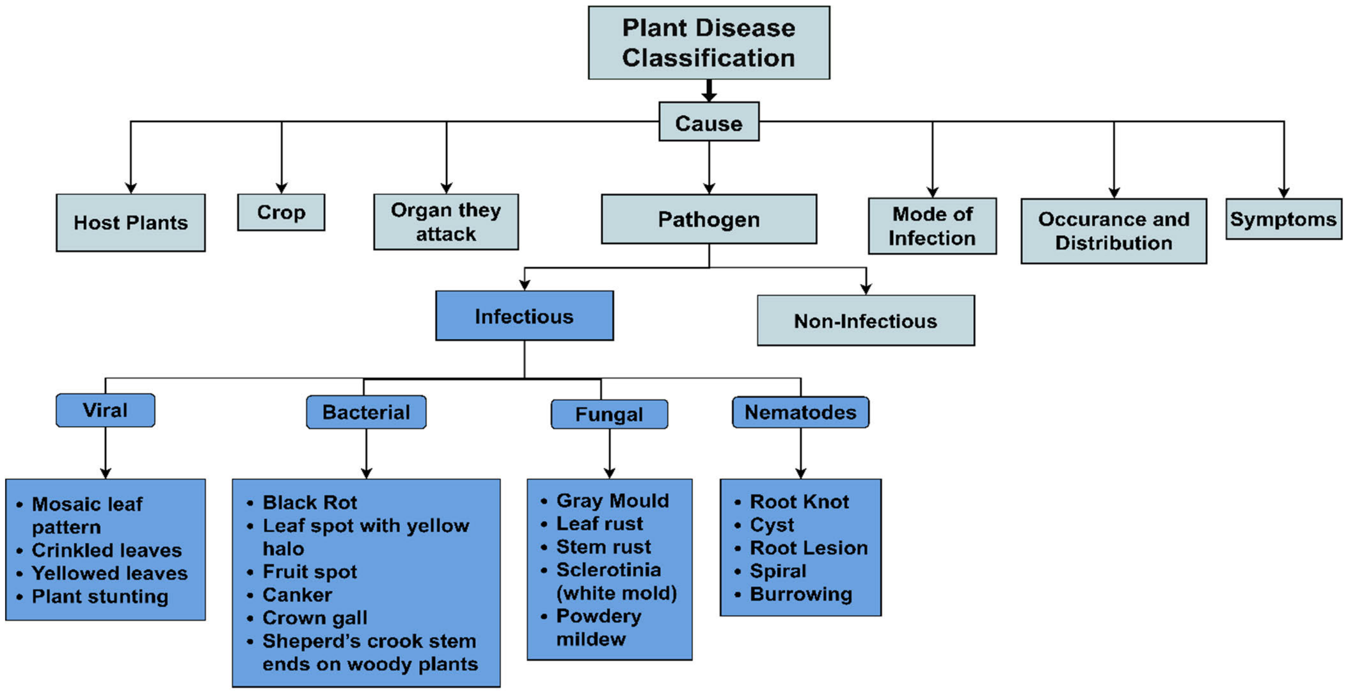


FIGURE 1. Classification of plant diseases.

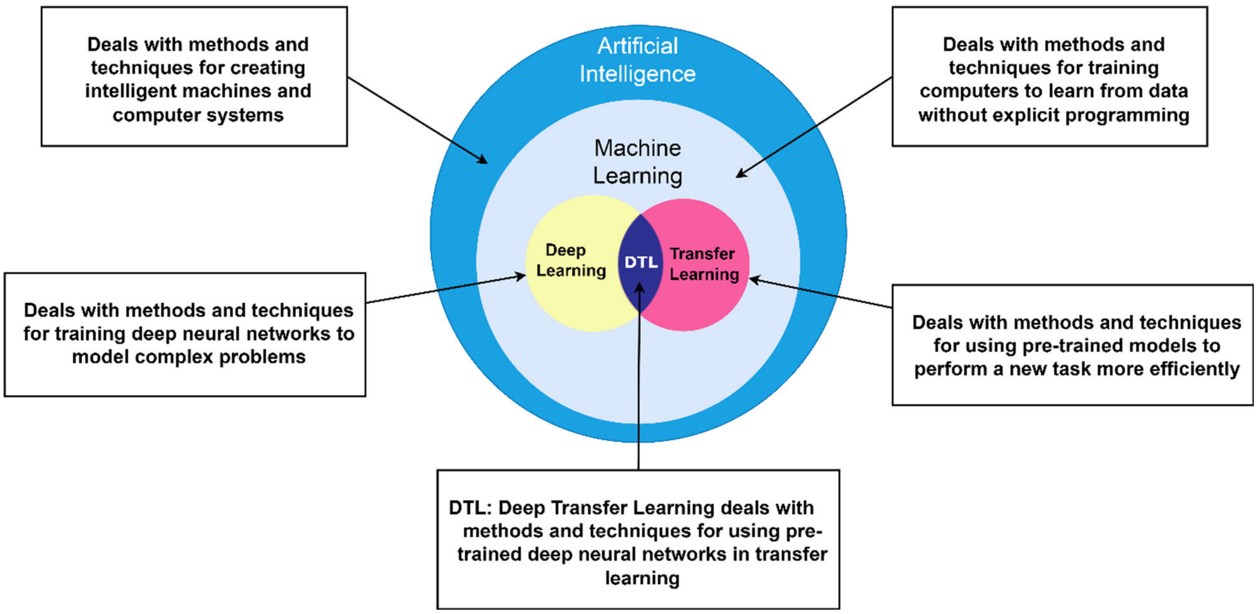


FIGURE 2. Visual representation of relationship between artificial intelligence, machine learning, deep learning, transfer learning, and deep transfer learning.

demonstrates that the optimal baseline model has been selected, providing outstanding results.

The paper is organized into several sections. The Introduction section provides an overview of the research topic and the purpose of the study. Following this is the related work section, which examines papers related to pathogen identification, plant disease detection, machine learning, Deep Learning, and transfer learning models. In the materials and methods section, the implementation specifics of the study are covered, including an introduction to deep transfer learning models and their architectures. The results and discussion section presents the findings of the experiment and provides an opportunity to discuss them.

II. LITERATURE SURVEY

In recent years researchers have proposed various techniques for automated disease detection in plants. These methods operate on distinct datasets. Some of them are studied and reviewed as discussed below:

Malik et al. [16] suggests a deep CNN hybrid approach for identification of sunflower leaf diseases, which makes use of Deep Learning techniques. Paper discusses four sunflower diseases, namely, Alternaria leaf blight, downy mildew, Phoma blight, and Verticillium wilt. Two models VGG-16 and MobileNet are combined, and one of the ensemble learning techniques, stacking, is used to learn the new model combination. With the same dataset, the proposed approach exceeds the competition, with an accuracy of 89.2%.

Liu et al. [17] extracts 19 feature values of color feature and texture feature from the diseased areas, and random forest algorithm is constructed to identify the diseased areas. A 95% overall recognition rate is possible.

A traditional Deep Learning model is proposed by Sara et al. [18], offers a dataset of sunflower leaves and flowers that will help researchers in developing effective algorithms for the detection of diseases. This process crosses over five major steps such as (a) Image Augmentation, (b) Resizing, (c) Splitting Images, (d) Model Generation, and (e) Performance Evaluation.

InceptionV3 is a Deep Learning model used for very accurate automated cassava disease identification, according to Ramcharan et al. [19]. With this technique, models can be easily trained on a desktop and deployed on mobile devices without the cumbersome and labor-intensive phase of feature extraction from images. As compared to the original dataset, it was anticipated that the larger image leaflet dataset would perform better for all illness groups. According to the results, the dataset size has no effect on improving prediction accuracy.

Huang et al. [20] incorporated cascade recovery of spike lesions at different scales into their training model, which was based on the GoogLeNet model. The Softmax classifier gave them a 92.0% accuracy rate. With the help of the feature map, Liu et al. improved the original image. Then, they used the Region Proposal Network (RPN) to down sample the suggested region and perform boundary regression and classification [21]. The rate of grape disease detection was 87.2%. In order to increase the network branch, Liu et al. combined the sparse self-encoder and CNN. They then used different sampling input from the encoder training to retrieve the low-dimensional features that represented the CNN's initial weight, which not only fixed the problem of its own small sample size but also spread up the network's convergence.

The paper [22] proposes a method for identifying diseases in apple leaves using Genetic Algorithm (GA) and Correlation-based Feature Selection (CFS) methods. The proposed method involves preprocessing of the images of apple leaves and extracting features using CFS. The GA is then used to select the most relevant features from the extracted features. The selected features are used to train a Support Vector Machine (SVM) classifier to identify the diseases in apple leaves. The proposed method is compared with other existing methods, and the results show that the proposed method performs better in terms of accuracy and efficiency. The proposed method can potentially be used for the early detection and prevention of diseases in apple trees.

Liu et al. [23] innovative machine learning approach was used to identify apple diseases. HIS(Hue, Saturation, and Intensity), YUV(Brightness (Y) + Color Difference (U,V)), and grey colour spaces were used by them to threshold out backdrop. A region-growing technique is used to calculate the shape, colour, and texture properties of each infected region. SVM is then used to classify the most prominent features, which were chosen using the Genetic Algorithm (GA) and Correlation-based Feature Selection (CFS) technique.

Xue et al. [24] segregate the samples and identify the levels of rotting in freshly cut cauliflowers using machine vision technology. PLS-DA(Partial Least Squares Discriminant Analysis) and ELM (Extreme Learning Machine) discriminant models had identification accuracy for decaying samples of 95% and 90.9%, respectively. Additionally, the rotting grades were divided based on the size of the rotten areas. The region growth algorithm was used to identify the contours and feature areas of the rotten cauliflower samples. The findings demonstrate the capability of machine vision technology to separate coherent fresh-cut cauliflower samples and to identify fresh and rotting cauliflower samples both qualitatively and quantitatively.

Several image classes are arranged by ImageNet into a semantic hierarchy that is well populated. A clean dataset for every level of the WordNet hierarchy is provided by Deng et al. [25]. On average, precision of 99.7% is attained. Their goal for ImageNet is to have nearly 50 million full-resolution, clean, and diversified images distributed over 50K synsets.

Krizhevsky et al. [26] Neural Network architecture contains 60 million parameters. They achieved 67.4% and 40.9% top-1 and top-5 error rates on ImageNet dataset. The best published results on this dataset are 78.1% and 60.9%

III. METHODOLOGY

A. DATASET DESCRIPTION

Existing Deep Transfer Learning pretrained model's performance evaluation is done using sunflower dataset, an open access standard dataset of plant disease classification. Sequentially, each image maintains a fixed width and height of 512 × 512 pixels. Gray mold, downy mildew, and leaf scars are three fungi that are highly common in sunflower plant [27]. These three fungi-related diseases are the main focus of the current dataset. Description of the dataset is as shown in Figure 3.

Diseases that affect cauliflower include bacterial spot rot, blackleg, black rot, clubroot, downy and powdery mildew, sclerotinia stem rot, white rust, cauliflower mosaic, ringspot, and more [28]. Additionally, it has been noted that a number of diseases, including downy mildew, black rot, and bacterial spot rot, regularly damage cauliflower bulb and leaves. In this

Sunflower Dataset				
Disease Name	Gray Mold	Leaf Scars	Downy Mildew	Disease-Free
Number of Original images	72	120	141	134
Number of Augmented Images	398	470	509	491

FIGURE 3. Sunflower dataset description.

Cauliflower Dataset					
Disease Name	Downy Mildew	Black Rot	Bacterial Spot Rot	Healthy Leaf	Healthy Bulb
Number of Original images	177	100	173	18	188
Number of Augmented Images	500	500	500	1000	1000

FIGURE 4. Cauliflower dataset description.

instance, the proposed dataset serves as a cutting-edge model for creating algorithms for the early detection of several cauliflower diseases in farming. The detailed information regarding the dataset is shown in Figure 4.

Agri-ImageNet is a subset created from the larger ImageNet dataset, which is a widely used benchmark dataset for object recognition tasks in computer vision. Agri-ImageNet is specifically designed for agricultural image recognition and classification, containing a curated collection of images related to various agricultural crops and produce.

Agri-ImageNet dataset includes classes such as zucchini, strawberry, spaghetti squash, rapeseed, pineapple, orange, mushroom, lemon, jackfruit, head cabbage, fig, daisy, custard apple, cucumber, coral fungus, cauliflower, cardoon, butternut squash, broccoli, bell pepper, banana, artichoke, agaric, acorn squash, and many others. These classes represent different types of crops, fruits, vegetables, and other agricultural produce.

Agri-ImageNet is developed as a subset of ImageNet by curating and categorizing images from the original ImageNet dataset that are relevant to agriculture. This subset is tailored specifically for agricultural image recognition tasks, providing a focused and specialized dataset for training and evaluating agricultural image classification models. Agri-ImageNet aims to facilitate research and development of computer vision models for agricultural applications, such as crop monitoring, disease detection, and yield estimation.

B. DEEP TRANSFER LEARNING

Deep Transfer Learning refers to a Machine Learning technique in which a pre-trained model is used as the starting point for training a new model on a different but related task [13]. The idea behind transfer learning is that the knowledge gained by a model in solving one task can be leveraged to improve performance on a different task.

In Deep Transfer Learning, the pre-trained model is usually a neural network that has been trained on a large dataset for a particular task, such as image classification or Natural Language Processing [29]. The pre-trained model is then used as a feature extractor, and the final layers of the model are replaced or fine-tuned to suit the new task.

For example, a pre-trained neural network that has been trained on a large dataset of images can be used as a feature extractor for a new image classification task. The pre-trained model is used to extract features from the images, and these features are then fed into a new neural network that is trained specifically for the new task.

Deep Transfer Learning has proven to be an effective technique for improving the performance of machine learning models, especially when the amount of training data for the new task is limited [30]. It has been widely used in many applications, including computer vision, natural language processing, and speech recognition.

Currently available keras deep transfer learning models are Xception, VGG_16, VGG_19, ResNet50, ResNet50V2, ResNet101, Resnet101V2, ResNet152, MobileNet, MobileNetV2, InceptionV3, InceptionResNetV2, DenseNet121, Densenet169, DenseNet201, NasNetMobile, NasNetLarge, EfficientNet models, EfficientNetV2 models, ConvNeXtTiny, ConvNeXtSmall, ConvNeXtBase, ConvNeXtLarge, ConvNeXtXlarge.

The idea behind deep transfer learning for image classification is that if a model is trained on a huge enough dataset with adequate generality, it will be able to represent the visual world as a whole. By training a big model on a large dataset, learnt feature maps can be used instead of having to start from scratch. Tabular review of different keras deep transfer learning models are presented in Table 1.

To utilize a pre-trained model for a new task, its architecture is used to create a base model while considering the pre-trained weights [42]. It is essential to freeze the pre-trained layers to prevent their weights from being re-initialized, as this would be equivalent to training the model from scratch. Next, new trainable layers are added to transfer the previous dataset's old features into predictions [43]. Since the pre-trained model is loaded without the final output layer, a new output layer, i.e., a dense layer at the top, must be trained with units equal to the number of outputs predicted by the model. This enables the model to make predictions when given new datasets as input. The model's performance can be improved through fine-tuning [44]. Unfreezing the underlying model allows for fine-tuning, after which the model can be retrained on the entire dataset at a very slow learning rate

TABLE 1. Deep transfer learning models description.

Sl. No.	Transfer Learning Model	Highlights
1	Xception	Xception is a Convolutional Neural Network (CNN) that uses depthwise separable convolutions to reduce the number of parameters and computation required to train the network. In traditional convolutional layers, each filter performs both the spatial convolution and the depthwise convolution. In contrast, depthwise separable convolutions separate the two operations, performing the spatial convolution and depthwise convolution independently, which can lead to significant computational savings [31].
2	VGG_16	VGG16 and VGG19 architectures are primarily composed of convolutional layers, followed by fully connected layers. They use a series of 3x3 convolutional filters with a stride of 1, and a max-pooling operation with a stride of 2 to reduce the spatial size of the feature maps [32].
3	VGG_19	
4	ResNet50	It is known for its use of residual connections, which enable the network to learn very deep representations without suffering from the vanishing gradient problem.
5	ResNet50V2	
6	ResNet101	ResNet50, ResNet101, and ResNet152 are variations of the original ResNet architecture, with 50, 101, and 152 layers, respectively. ResNet50V2, ResNet101V2, and ResNet152V2 are updated versions of these models that use a modified residual block design, to improve performance [33].
7	ResNet101V2	
8	ResNet152	
9	ResNet152V2	
10	InceptionV3	InceptionV3 uses a combination of convolutional layers, max-pooling layers, and inception modules that contain multiple parallel convolutional layers with different filter sizes. This allows the network to capture information at different spatial scales, leading to better performance on image classification tasks [34].
11	InceptionResNetV2	InceptionResNetV2 uses a combination of convolutional layers, max-pooling layers, and inception modules with residual connections. The residual connections enable the network to learn very deep representations without suffering from the vanishing gradient problem [35].
12	MobileNet	MobileNet uses depthwise separable convolutions to reduce the number of parameters and computation required to train the network. This makes it an efficient and lightweight architecture that can run on resource-constrained devices.
13	MobileNetV2	MobileNetV2 further improves on MobileNet with a new block design that includes linear bottlenecks and shortcut connections between the bottlenecks. This leads to improved accuracy while maintaining the efficiency of the original MobileNet architecture [36].
14	DenseNet121	DenseNet models are widely used as pre-trained models for transfer learning in various computer vision applications, such as object detection and segmentation. The use of dense connectivity helps to alleviate the vanishing gradient problem and allows the network to learn very deep representations effectively [37].
15	Densenet169	
16	DenseNet201	
17	NasNetMobile	NasNetMobile is a lightweight architecture that is designed for mobile and embedded devices, while NasNetLarge is a larger architecture that is intended for use in more powerful computing environments. Both architectures use a combination of convolutional layers, batch normalization, and separable depthwise convolutions to reduce the number of parameters and computation required to train the network. They have achieved state-of-the-art performance on several benchmarks and have been widely used in various computer vision applications, such as image classification and object detection [38].
18	NasNetLarge	
19	EfficientNetB0	EfficientNet is a family of deep neural network architectures that was proposed by researchers at Google in 2019. These models are designed to be both accurate and efficient, making them suitable for use in resource-constrained environments. The EfficientNet family includes several variations, from EfficientNetB0 to EfficientNetB7, with increasing depth, width, and resolution. The models use a combination of depthwise and pointwise
20	EfficientNetB1	
21	EfficientNetB2	

TABLE 1. ***(Continued.)*** **Deep transfer learning models description.**

22	EfficientNetB3	convolutions, along with a compound scaling method that optimizes the balance between accuracy and computational efficiency. EfficientNet models have achieved state-of-the-art performance on various image classification benchmarks, including ImageNet. They have also been used in various computer vision applications, such as object detection and segmentation, and have shown to require fewer parameters and computation compared to other deep neural network architectures with similar performance [39].
23	EfficientNetB4	
24	EfficientNetB5	
25	EfficientNetB6	
26	EfficientNetB7	
27	EfficientNetV2B0	EfficientNetV2 is the second generation of the EfficientNet family, proposed by Google in 2021. These models continue to focus on achieving a balance between accuracy and computational efficiency, but with additional improvements in architecture design and training techniques. EfficientNetV2 includes several variations, from EfficientNetV2B0 to EfficientNetV2L, with varying depth, width, and resolution. The models use a combination of efficient building blocks, such as Squeeze-and-Excitation (SE) and Swish activation, and improved training techniques, such as improved data augmentation and stochastic depth [40].
28	EfficientNetV2B1	
29	EfficientNetV2B2	
30	EfficientNetV2B3	
31	EfficientNetV2S	
32	EfficientNetV2M	
33	EfficientNetV2L	
34	ConvNeXtTiny	ConvNeXt is a deep neural network architecture proposed by Facebook AI Research in 2018. It is designed to improve the performance of Convolutional Neural Networks (CNN) by using a combination of group convolutions and concatenation. ConvNeXtTiny, ConvNeXtSmall, ConvNeXtBase, ConvNeXtLarge, and ConvNeXtXlarge are variations of the original ConvNeXt architecture with increasing depth and width. These models have achieved state-of-the-art performance on various image classification benchmarks, including ImageNet. The use of group convolutions and concatenation helps to reduce the number of parameters and computation required to train the network, making it more efficient and scalable. ConvNeXt has been widely used in various computer vision applications, such as object detection and segmentation [41].
35	ConvNeXtSmall	
36	ConvNeXtBase	
37	ConvNeXtLarge	
38	ConvNeXtXlarge	

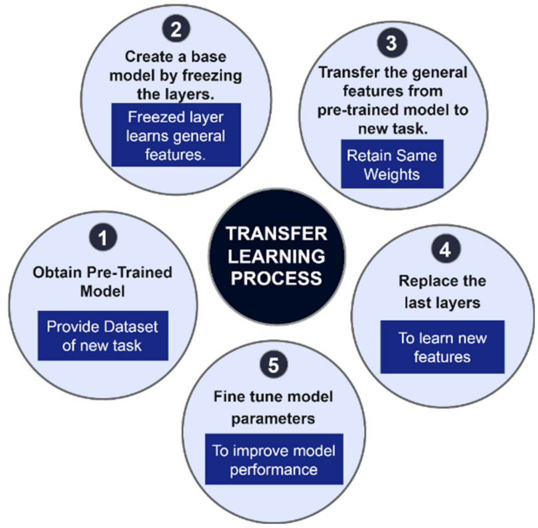


FIGURE 5. **Transfer learning process.**

to prevent overfitting and improve the model's performance on the new dataset. The learning rate must be low because the model is larger than the dataset [45]. The condensed overview of this process is shown in Figure 5.

Transfer learning has proven to be an effective approach for various computer vision tasks, including object detection, image classification, and segmentation. One of the significant advantages of transfer learning is that it reduces the amount of labeled training data required for a new task. This is particularly useful when labeled data is scarce or expensive to obtain. By leveraging pre-existing models trained on large datasets, transfer learning enables us to transfer the knowledge and patterns learned from one domain to another.

C. BLOCK DIAGRAM OF PLANT DISEASE DETECTION USING TRANSFER LEARNING

Transfer learning has gained popularity in the field of machine learning as a technique to solve related but different problems by leveraging pre-existing models trained on large datasets [46]. This approach can be applied in various ways, with one common method being to use a pre-trained model as a fixed feature extractor. In this method, the pre-trained layers' weights are frozen, and only new layers on top of it are trained for a specific task [47]. Another approach is fine-tuning, where the pre-trained model's weights are modified to improve its performance on a specific task, as shown in Figure 6.

In the context of plant disease detection, transfer learning has been successfully used to detect various diseases caused by pathogens such as fungi, bacteria, and viruses. We have achieved high accuracy in classifying plant disease images by using pre-trained models such as VGG, Inception, and ResNet [48].

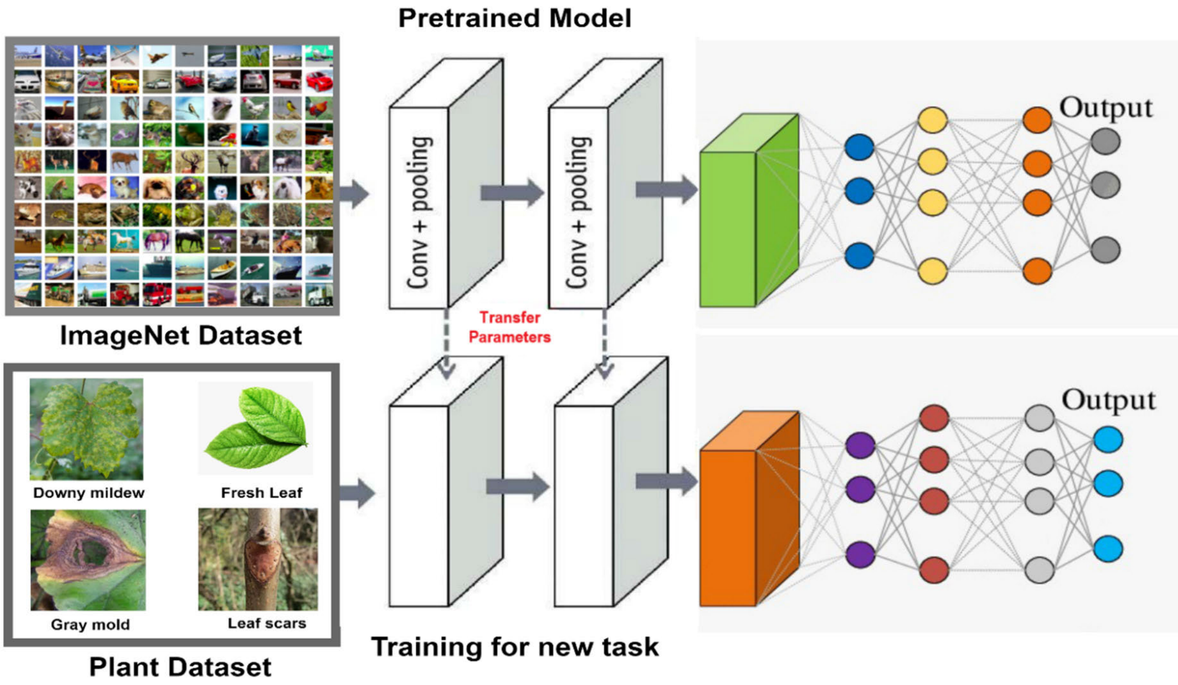


FIGURE 6. Transfer learning for plant disease detection.

To detect the pathogen responsible for a specific plant disease, researchers have utilized 38 transfer learning models on sunflower and cauliflower datasets. The proposed approach's data flow diagram along with derivation of Agri-ImageNet dataset from ImageNet dataset are shown in Figure 7-9.

IV. PERFORMANCE EVALUATION AND RESULTS

A. PERFORMANCE EVALUATION METRICS

Evaluating the performance of machine learning models is one of the key steps in creating effective ML models. Various metrics are used to evaluate model performance and quality, and these metrics help in understanding how well the model performed on a particular data. This allows us to improve model performance by optimizing hyper parameters. ML models are intended to generalize well to unseen and new data. Performance metrics help determine how well the model generalizes on new datasets. Performance metrics considered for Plant Disease Detection are highlighted in Table 2. From 38 models, a list of 10 models were selected based on their varied metric behavior. The efficiency of the proposed pathogen-based plant disease detection approach is evaluated as shown in Table 3 and Table 4 with respect to sunflower and cauliflower datasets.

The results presented in Table 3 and 4 suggest that the majority of the Deep Transfer Learning models tested were not successful in accurately distinguishing between images of downy mildew and leaf scars, as well as bacterial soft rot and downy mildew leaf.

This highlights the challenges involved in using Transfer Learning algorithms for the classification of plant diseases, which can be attributed to the similarity in visual appearance of some conditions.

B. TRANSFER LEARNING MODELS PERFORMANCE EVALUATION VIA CLASSIFICATION REPORT

To identify the optimal model, three separate datasets were evaluated. The chosen model must possess both simplicity and accuracy. By taking into account various performance factors such as training time, complexity, and F1 Score, a productive outcome can be achieved.

Our initial task is to identify the models that perform well in detecting plant diseases. Since all of the models were trained on the ImageNet dataset, which comprises 1000 classes, we take into account a subset of that dataset called "Agri-ImageNet," which includes all of the classes pertaining to plants. The 38 models are trained using the Agri-ImageNet dataset, and their accuracies are recorded as shown in TABLE. 5.

Why only accuracy?

The dataset used in this study is comprehensive and well-categorized, allowing for accurate evaluation of the models. This simplicity in evaluation enables clear decision-making in advancing the models. The 38 transfer learning models were trained with the same weights obtained from pretraining on the ImageNet dataset, and their accuracies were evaluated on three different datasets. The results are presented in Table 5 for easy comparison and analysis.

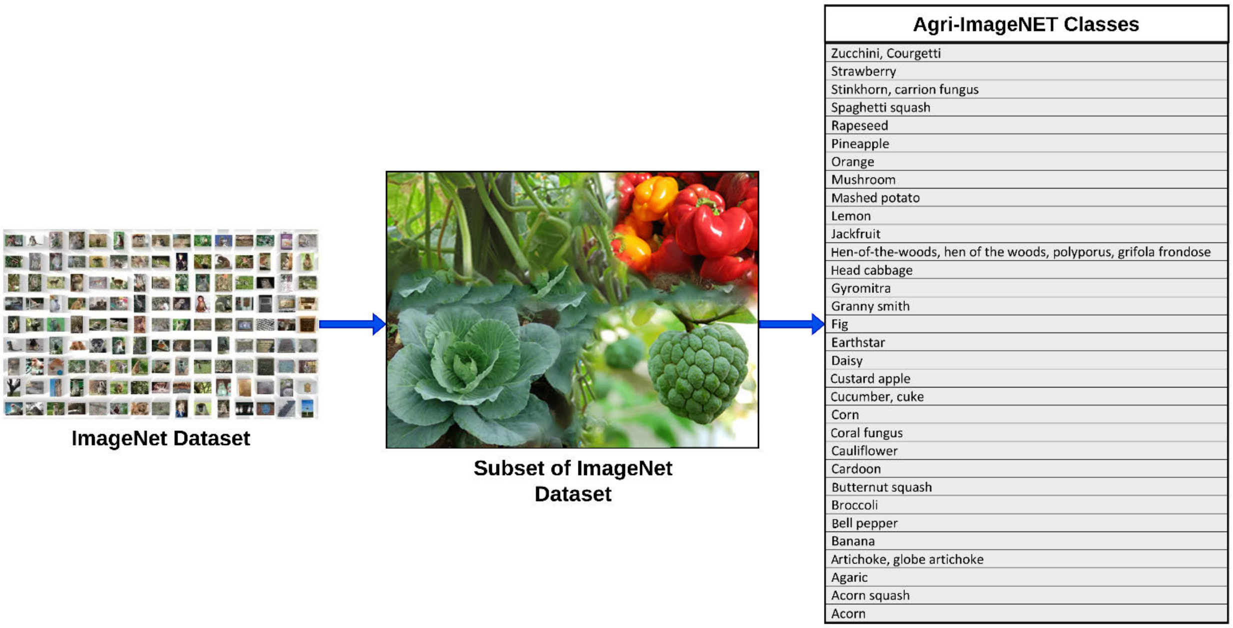
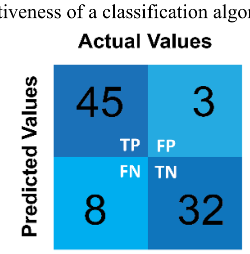


FIGURE 7. Extraction of Agri-ImageNet classes from ImageNet dataset.

TABLE 2. Classification metrics for models evaluation.

Sl. No.	Metrics	Description	Formula	Sunflower Dataset
1.	**Confusion Matrix**	It is a method for analyzing the effectiveness of a classification algorithm. 	**TP**-True Positive: The model predicted Yes, and the real or actual value also indicated Yes. **FN**-False Negative: The model predicted No, but the actual result was Yes. **FP**-False Positive: The model predicted Yes, but the actual result was No. **TN**-True Negative: The model predicted No, and the real or actual value also indicated No.	**Example** **TP**- True Positive: Classifying all 45 samples of Downey Mildew as Yes.. **FN**-False Negative: Misclassifying 15 samples of Downey Mildew as No instead of Yes. **FP**-False Positive: Misclassifying 8 samples of Downey Mildew as Yes instead of No. **TN**-True Negative: Classifying 32 samples other than Downey Mildew correctly as No.
2.	**Precision**	It represents the ratio of correctly predicted observations to all positively predicted observations.	$Precision = \frac{True\ Positive}{True\ Positive + False\ Position}$	
3.	**Recall**	It measures how many correctly anticipated positive observations there were compared to all of the actual class observations.	$Recall = \frac{True\ Postive}{True\ Positive + False\ Negative}$	
4.	**F1 Score**	It is the average of Precision and Recall weighted together.	$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$	
5.	**Accuracy**	The ratio of correctly anticipated observations to all observations is what determines this.	$Accuracy = \frac{True_{positive} + True_{negative}}{True_{positive} + True_{negative} + False_{positive} + False_{negative}}$	

For a broader classification, let us choose models with accuracy higher than 90% as shown in TABLE 6.

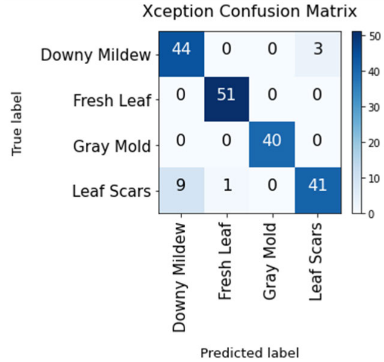
The use of transfer learning has shown great potential in improving the accuracy of plant disease detection and classification.

In Table 6, 10 transfer learning models are selected as good options for detecting or classifying plant data. To further investigate their performance, the study chose two specific datasets: sunflower and cauliflower.

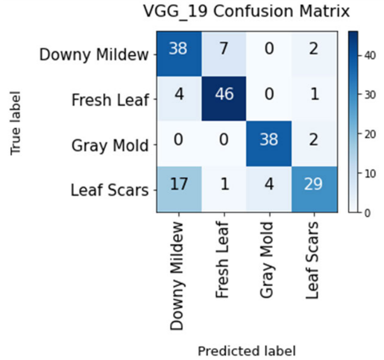
Table 7 presents the results for these two datasets, showing the accuracy achieved by each of the 10 models for each dataset. It is interesting to note that some models performed better on one dataset than the other.

TABLE 3. Classification report of sunflower dataset.

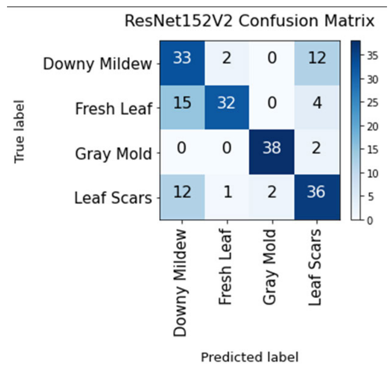
The confusion matrix represents the performance of a classifier on a sunflower dataset with 4 classes: "Downy Mildew-47 Test Images", "Fresh Leaf-51 Test images", "Gray Mold-40 Test Images", and "Leaf scars-51 Test Images".



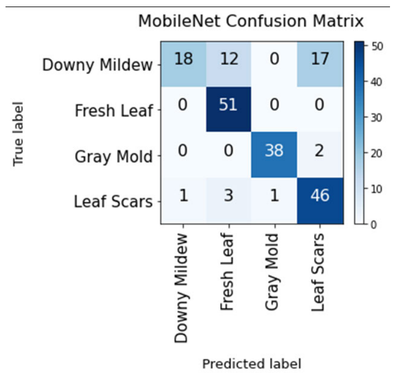
Xception
A scrutiny of the test images revealed that out of 47 Downy Mildew images, 3 were inaccurately identified as Leaf Scar. Additionally, of the 51 Leaf Scar images, 9 were mistakenly labeled as Downy Mildew and 1 as Fresh Leaf. These findings suggest that the majority of misclassifications are taking place between ***Downy Mildew and Leaf Scar images.***



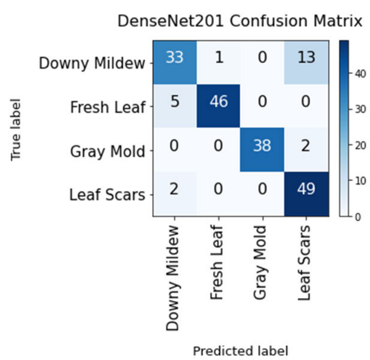
VGG19
An examination of the test images shows that 14.89% (7 images) of the Downy Mildew images were misclassified as Fresh Leaf and 4.26% (2 images) were misclassified as Leaf Scar. Out of 51 Fresh Leaf images, 7.84% (4 images) were misclassified as Downy Mildew and 1.96% (1 image) was misclassified as Leaf Scar. And out of 51 Leaf Scar images, 33.33% (17 images) were misclassified as Downy Mildew, 1.96% (1 image) is misclassified as Fresh Leaf, and 7.84% (4 images) are misclassified as Gray Mold. These findings indicate that a significant portion of misclassifications occur between ***Downy Mildew and Leaf Scar images.***



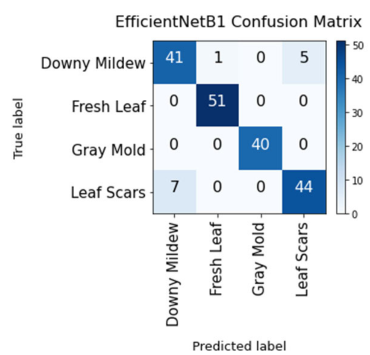
ResNet152V2
Misclassification between ***Downy Mildew and Fresh Leaf images.***



MobileNet
Misclassification between ***Downy Mildew and Leaf Scar images.***

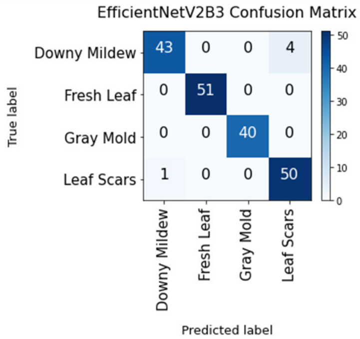


DenseNet201
Misclassification between ***Downy Mildew and Leaf Scar images.***

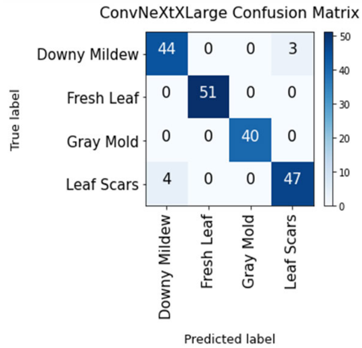


EfficientNetB1
Misclassification between ***Downy Mildew and Leaf Scar images.***

TABLE 3. ***(Continued.)*** **Classification report of sunflower dataset.**



EfficientNetV2B3
Misclassification between ***Downy Mildew and Leaf Scar images.***



ConvNeXtXlarge
Misclassification between ***Downy Mildew and Leaf Scar images.***

For example, the ConvNeXtSmall and ConvNeXtBase models achieved accuracy above 90% for the sunflower dataset but below 90% for the cauliflower dataset. This highlights the importance of selecting the appropriate model for a specific dataset.

Similarly, the EfficientNetV2B1 model achieved accuracy above 90% for the cauliflower dataset but below 90% for the sunflower dataset. This demonstrates that the performance of a model depends not only on the dataset but also on the specific layers and parameters used in the model.

Overall, these results suggest that selecting the appropriate transfer learning model is crucial for achieving high accuracy in plant disease detection or classification. It is important to evaluate the performance of multiple models on a specific dataset to identify the best model for the task at hand.

To identify the superior model, various aspects of the models' performance must be analyzed, including the number of parameters trained (measured in millions), the number of layers, accuracy, and F1 score. Since number of parameters and number of layers are interdependent, for convenience let us only consider number of parameter (in million)

Let 'F' be F1 Score, 'A' be accuracy and 'P' be number of parameters trained (in million), and N is derived as shown in equation (a)

$$N = \frac{F * A}{P} \tag{a}$$

(Greater the value of N the better is the Model)

1. *F1 Score and Accuracy*: The F1 Score and accuracy are commonly used performance metrics for evaluating the effectiveness of a model in making correct predictions. The F1 Score considers both precision and recall, providing a balanced measure of the model's ability to correctly classify positive and negative samples. Accuracy, on the other hand, measures the overall correctness of the model's predictions. By multiplying F1 Score and accuracy together, the formula takes into account both the precision, recall, and overall correctness of the model's predictions.
2. *Number of Parameters:* The number of parameters in a machine learning model represents its complexity. Models with more parameters are generally more complex and have a higher capacity to learn complex patterns from the data. However, a higher number of parameters also increases the risk of overfitting, where the model may memorize the training data instead of learning generalized patterns. The formula divides the F1 Score and accuracy by the number of parameters, P, which takes into account the complexity of the model.
3. *Importance of N:* The formula N = (F * A) / P provides an overall measure of the model's performance, taking into account both the accuracy of the model's predictions and its complexity (number of parameters). A higher value of N indicates a better-performing model that achieves a good balance between accuracy and model complexity. In other words, a higher N value indicates that the model achieves good performance (high F1 Score and accuracy) relative to its complexity (number of parameters). This formula provides a way to evaluate and compare models based on their performance and complexity, which can be useful in model selection and optimization tasks.

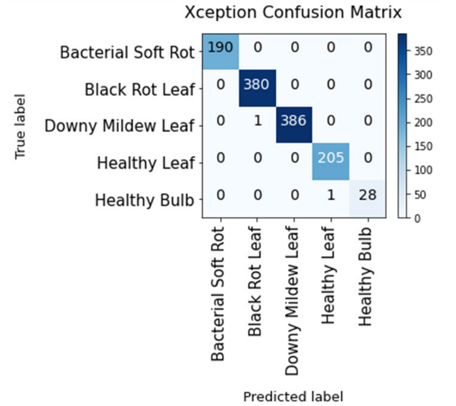
After a thorough evaluation of various models, it was determined that the EfficientNetV2B2 outperforms its competitors in terms of accuracy, F1 score, and inverse parameters' product, as depicted in Table 8. Although EfficientNetV2B1 boasted a high score in these areas, the EfficientNetV2B2 was found to have a slight advantage in terms of accuracy. Hence, EfficientNetV2B2 emerged as the top pick for detecting plant diseases in the Sunflower dataset.

C. TRANSFER LEARNING MODELS PERFORMANCE EVALUATION VIA FEATURE MAPS

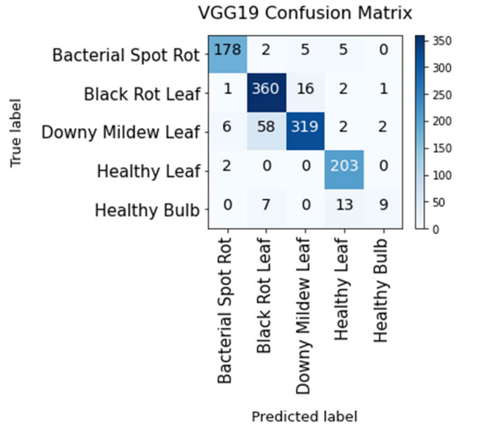
To attain a good model for Plant disease detection, we consider many criteria. This includes F1 score and Accuracy. Thus, EfficientNetV2B2 has been chosen for this purpose. Table 7 describes accuracies of 2 datasets considered for

TABLE 4. Classification report of cauliflower dataset.

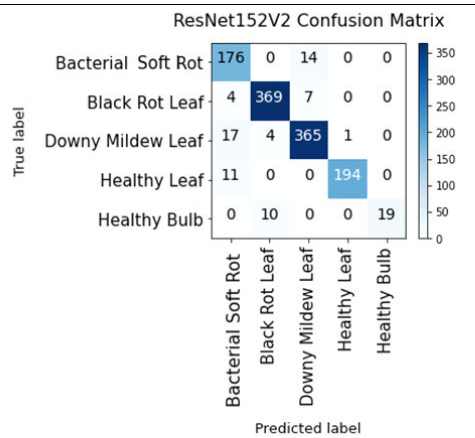
The confusion matrix represents the performance of a classifier on the Cauliflower dataset with 5 classes: "Bacterial spot rot", "Black Rot leaf", "Downy Mildew leaf", "Healthy leaf", and "Healthy Bulb". Each row of the matrix corresponds to the actual class and each column corresponds to the predicted class. The values in the matrix are the number of instances that were actually in the row class and predicted to be in the column class.



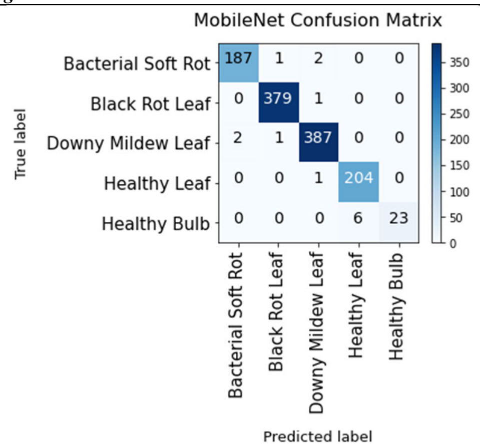
Xception
Misclassification between ***Black Rot & Downey Mildew Leaf images.***



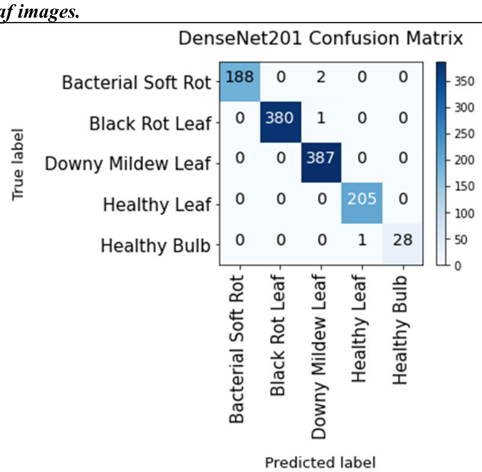
VGG19
Major misclassification between ***Healthy Leaf & Healthy Bulb images***.



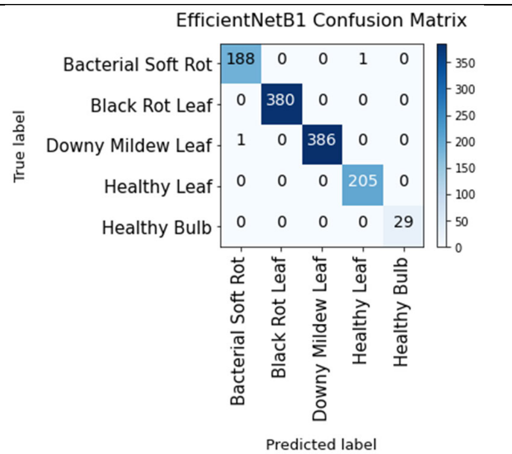
ResNet152V2
Misclassification between ***Bacterial Soft Rot & Downey Mildew Leaf images.***



MobileNet
Misclassification between ***Healthy Leaf & Healthy Bulb images.***

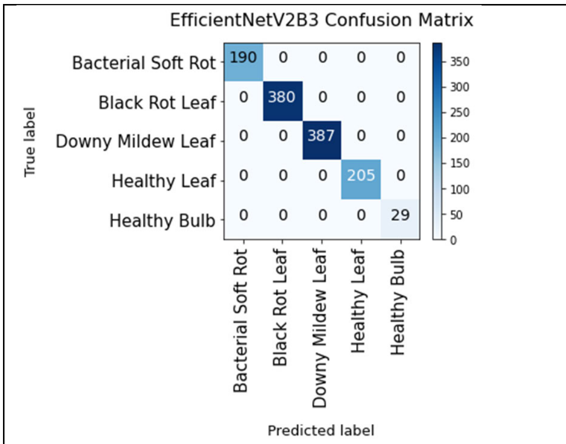
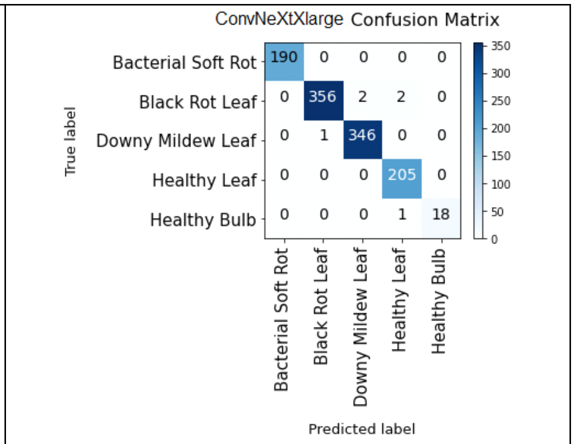


DenseNet201
Misclassification between ***Bacterial Soft Rot & Downey Mildew Leaf images.***



EfficientNetB1
Misclassification between ***Bacterial Soft Rot & Downey Mildew Leaf images.***

TABLE 4. ***(Continued.)*** **Classification report of cauliflower dataset.**

	
EfficientNetV2B3 No misclassification at all.	***ConvNeXtXlarge*** Slight misclassification between ***Black Rot Leaf & Downey Mildew Leaf images.***

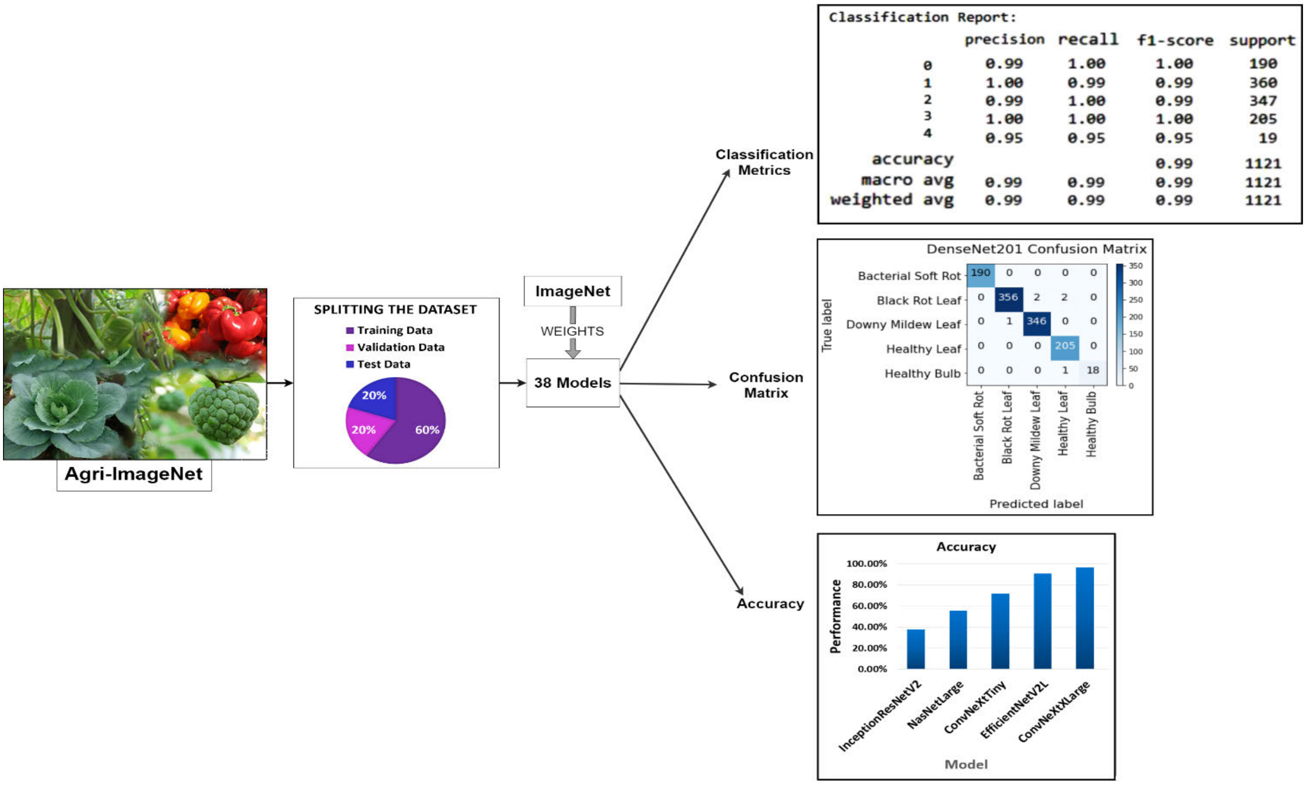


FIGURE 8. **Transfer learning for Agri-ImageNet.**

all 38 Models where Loss is Inversely Proportional to Accuracy.

For the Agri-ImageNet, Sunflower, and Cauliflower dataset, InceptionResNetV2 shows room for improvement with its accuracy being below 40%. This presents an opportunity for growth in the following areas:

1. Improved Dataset Quality
2. Enhanced Feature Learning
3. Reduced Overfitting (Low Variance)
4. Optimal Hyperparameter Selection

Hyper parameters such as optimizer, learning rate, number of epochs, and patience in early stopping are considered to be same for all the 38 models and hence low accuracy is not dependent on hyper parameter. We have used "Adam" optimizer, 0.001 Learning rate, 30 Epoch and 6 patience.

Since the dataset is trained only once on the models, the issue of high variance or overfitting is not encountered, ensuring that the models' performance remains consistent and reliable.

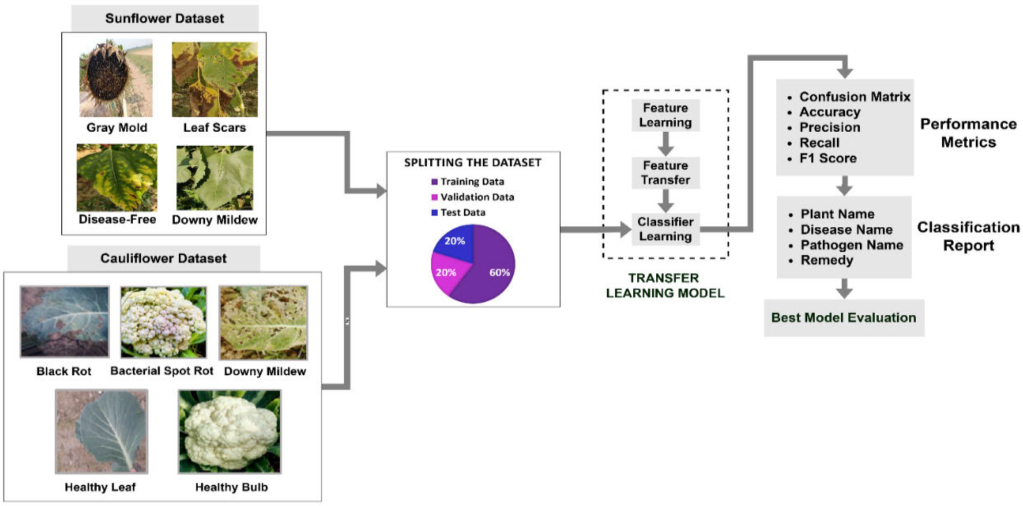


FIGURE 9. Dataflow diagram for pathogen-based plant disease detection.

TABLE 5. Accuracy assessment of the sunflower, cauliflower, and agri-imagenet datasets.

Sl. No .	Models	Accuracy (Sunflower)	Accuracy (Cauliflower)	Accuracy (Agri-ImageNe t)
1	Xception	99.832	93.122	29.375
2	VGG_16	93.619	76.190	3.232
3	VGG_19	89.757	79.894	3.125
4	ResNet50	84.383	52.910	72.625
5	ResNet50V2	92.275	61.905	29.5625
6	ResNet101	99.580	70.370	73.875
7	ResNet101V2	62.963	62.963	74.93
8	ResNet152	99.748	89.947	26.4375
9	ResNet152V2	94.291	73.545	21.8125
10	InceptionV3	99.496	89.418	6.4375
11	InceptionResNetV2	37.867	39.153	30.75
12	MobileNet	98.825	78.307	25.1249
13	MobileNetV2	83.459	59.259	78.875
14	DenseNet121	99.328	85.185	84.4375
15	Densenet169	99.496	88.889	54.1875
16	DenseNet201	99.748	87.831	50.7499
17	NasNetMobile	96.474	87.302	16.625
18	NasNetLarge	55.668	21.164	18.8125
19	EfficientNetB0	98.825	92.063	83.1875
20	EfficientNetB1	99.748	97.884	86.25
21	EfficientNetB2	99.748	95.238	86.125
22	EfficientNetB3	99.832	94.180	87.3125
23	EfficientNetB4	99.832	93.651	89.062
24	EfficientNetB5	99.580	88.889	87.5
25	EfficientNetB6	98.992	87.831	87.75
26	EfficientNetB7	99.076	88.360	88.625
27	EfficientNetV2B0	97.397	93.122	89.125
28	EfficientNetV2B1	99.916	88.889	90.5
29	EfficientNetV2B2	100.000	96.825	90.3125
30	EfficientNetV2B3	100.000	97.354	92.4375
31	EfficientNetV2S	99.832	94.180	92.75
32	EfficientNetV2M	99.244	93.122	92.175
33	EfficientNetV2L	99.832	91.005	91.8125
34	ConvNeXtTiny	71.759	95.238	86.437
35	ConvNeXtSmall	80.456	93.122	92.067
36	ConvNeXtBase	89.649	94.180	92.313
37	ConvNeXtLarge	92.648	95.767	93.97
38	ConvNeXtXlarge	99.175	96.296	94.313

If you experience high variance or overfitting in your model, consider implementing the following techniques to resolve the issue:

TABLE 6. Accuracies of Agri-ImageNet dataset >90%.

Models	Accuracy (Agri-ImageNet)
EfficientNetV2B0	89.125
EfficientNetV2B1	90.5
EfficientNetV2B2	90.3125
EfficientNetV2B3	92.4375
EfficientNetV2S	92.75
EfficientNetV2M	92.175
EfficientNetV2L	91.8125
ConvNeXtSmall	92.067
ConvNeXtBase	92.313
ConvNeXtLarge	93.97
ConvNeXtXlarge	94.313

1. *Regularization techniques such as L1 and L2:* These techniques introduce constraints on the model to prevent overfitting by penalizing high magnitude weight coefficients for Level 1(L1) and Level 2(L2) Regulations.
2. *Early stopping:* This technique stops the training process before the model starts memorizing the training data, which would cause overfitting.
3. *Cross-validation:* By splitting the data into multiple subsets and training the model on one set while validating it on another, this technique provides a more robust evaluation of the model's performance and helps prevent overfitting.
4. *Reducing features:* By reducing the number of features used in the model, the complexity of the model is reduced, reducing the chances of overfitting.

TABLE 7. Accuracies of cauliflower and sunflower datasets with f1 score for selected models.

Models	Sunflower		Cauliflower	
	F1 Score	Accuracy	F1 Score	Accuracy
EfficientNetV2B1	0.885	88.8889	0.996	99.916037
EfficientNetV2B2	0.97	96.8254	1	100
EfficientNetV2B3	0.975	97.3545	1	100
EfficientNetV2S	0.9425	94.1799	0.992	99.832074
EfficientNetV2M	0.9325	93.1217	0.974	99.244332
EfficientNetV2L	0.9125	91.0053	0.996	99.832074
ConvNeXtSmall	0.9325	93.1217	0.958	80.45637283
ConvNeXtBase	0.9425	94.1799	0.988	89.64898563
ConvNeXtLarge	0.9575	95.7672	0.984	92.64809255
ConvNeXtXLarge	0.965	96.2963	0.99	99.17463098

TABLE 8. Values of 'n' for cauliflower and sunflower datasets for selected models.

Sl No	Models	Parameters (in millions)	Sunflower			Cauliflower		
			F1 Score	Accuracy	N	F1 Score	Accuracy	N
1	EfficientNetV2B1	8.2	0.885	88.8889	9.593495935	0.996	99.916	12.13614302
2	EfficientNetV2B2	10.2	0.97	96.8254	9.207905384	1	100	9.803921569
3	EfficientNetV2B3	14.5	0.975	97.3545	6.546250684	1	100	6.896551724
4	EfficientNetV2S	21.6	0.9425	94.1799	4.10946992	0.992	99.8321	4.58488043
5	EfficientNetV2M	54.4	0.9325	93.1217	1.596249611	0.974	99.2443	1.776911394
6	EfficientNetV2L	119	0.9125	91.0053	0.697834689	0.996	99.8321	0.835569291
7	ConvNeXtSmall	50.2	0.9325	93.1217	1.729800375	0.958	80.45637	1.535402494
8	ConvNeXtBase	88.5	0.9425	94.1799	1.002989269	0.988	89.64899	1.000827094
9	ConvNeXtLarge	197.7	0.9575	95.7672	0.463819373	0.984	92.64809	0.461131629
10	ConvNeXtXLarge	350.1	0.965	96.2963	0.265426809	0.99	99.17463	0.280442401

5. Expanding the training dataset: Increasing the size of the training dataset gives the model more examples to learn from, reducing the chances of overfitting.
6. *Ensemble methods:* By combining multiple models, ensemble methods reduce the variance of individual models and increase their robustness.
7. *Bagging and Boosting*: These are methods that create multiple models, either by aggregating predictions from individual models or by iteratively improving a single model.
8. *Dropout and data augmentation*: By randomly dropping out neurons during training or creating new data samples through transformations, these techniques reduce overfitting by forcing the model to learn more robust representations of the data.
9. *Feature selection and dimensionality reduction:* By reducing the number of features used in the model or transforming the data into a lower-dimensional space, these techniques can reduce overfitting by reducing the complexity of the model.
10. *Hyperparameter tuning:* By adjusting the parameters of the model, this technique can help prevent overfitting by finding the optimal settings that prevent the model from memorizing the training data.

A low accuracy with respect to Agri-ImageNet is due to the presence of irrelevant images in both training and testing datasets as shown in Figure.10.

The feature map is an outcome of one filter applied to the layer before it. Each pixel of a certain filter is moved as it is drawn across the entire layer before it. Each location activates a neuron, and the output is collected in the feature map. It can be observed that the field will overlap with the preceding activation by input values if the receptive field is shifted one pixel from activation to activation. We select the best observable convolutional layer for both models since the next layer of batch normalization cannot be seen through adequately. Figures 11 and 12 display the feature maps of the InceptionResNetV2 and EfficientNetV2B2 models, respectively, for the Sunflower Dataset. Specifically, Figure 11 shows that the features in the "conv2d" layer of InceptionResNetV2 are not atomized and therefore not easily observable, whereas in Figure, the features in the "block1a_project_conv" layer of EfficientNetV2B2 are atomized and more observable. Similarly, Figures 13 and 14 compare the feature maps for the cauliflower dataset in the respective first convolutional layers of InceptionResNetV2 and EfficientNetV2B2

FIGURE 10. Irrelevant images of Agri-ImageNet.

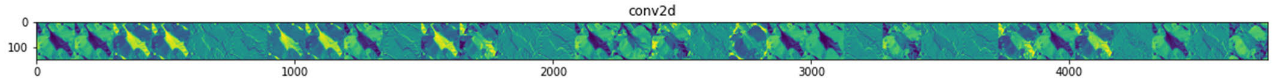


FIGURE 11. Feature map of convolution layer "conv2d" of InceptionResNetV2 model belonging to sunflower dataset.

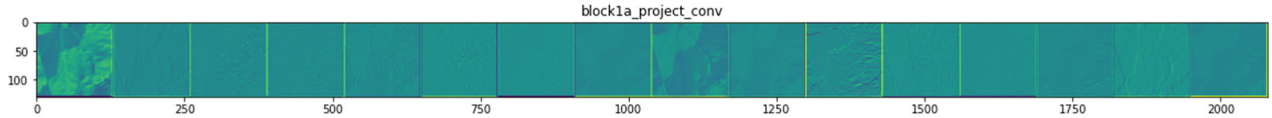


FIGURE 12. Feature map of convolution layer "block1a_project_conv" of EfficientNetV2B2 model-sunflower dataset.

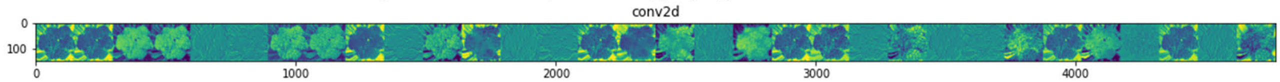


FIGURE 13. Feature map of convolution layer "conv2d" of InceptionResNetV2 model belonging to cauliflower dataset.

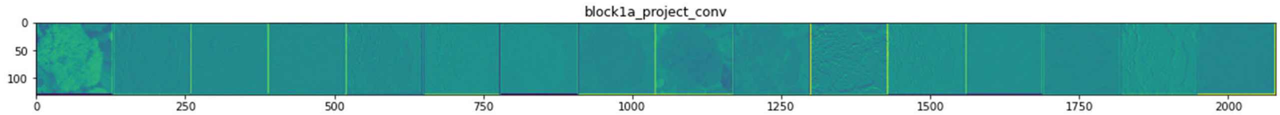


FIGURE 14. Feature map of convolution layer "block1a_project_conv" of EfficientNetV2B2 model-cauliflower dataset.

models. In these feature maps, purple patches indicate learned features, while green portions represent unlearned features. However, it is important to note that the interpretation of colors in feature maps can vary depending on visualization techniques and context.

Analyzing how a model learns to recognize different features in images provides insights into its operations. It can also help identify potential causes of inaccurate categorization of images, leading to fine-tuning for improved accuracy. For example, Figure 9 shows that the EfficientNetV2B2 model is not overfitting as there are still unlearned green features, but the presence of high-intensity purple spots indicates that the model is performing at its best potential. When interpreting feature maps, it is important to take into account the unique architecture and design of the model being employed, as relying solely on color categorization may lead to inaccurate conclusions. Careful analysis and understanding of the underlying model behavior is essential for interpreting feature maps correctly.

V. CONCLUSION

This work presents Transfer Learning approach for plant disease detection by considering most common diseases in sunflower and cauliflower plants using different parts of the plant such as leaf, bulb and flower. Comparative study is done w.r.t different performance metrics on all the currently available transfer learning models in keras to understand how the classification accuracy varies for each model for a specific dataset. 38 transfer learning models are tested on Sunflower, Cauliflower and Agri-Imagenet (subset of ImageNet dataset which includes all the classes pertaining to plants) datasets for plant disease detection with an early stopping of maximum 10 to 16 epochs and patience of 6. The experimental results have shown that InceptionResNetV2 serves as the standard benchmark, and the models of EfficientNetV2B1, EfficientNetV2B2, and EfficientNetV2B3 are exceptional choices for identifying plant diseases.

The lower performance of VGG-16, VGG-19, Inception, NasNet, and some ResNet models in the Agri-ImageNet dataset could be attributed to factors such as their deep architecture, unique design, potential suboptimal fine-tuning, and dataset characteristics. These models may struggle with capturing the specific features and patterns relevant to plant diseases in the Agri-ImageNet dataset, leading to lower accuracy compared to other models. Further analysis and investigation would be needed to determine the exact reasons for their lower performance in the Agri-ImageNet dataset.

After conducting a comprehensive analysis of 38 transfer learning models across 3 different datasets including Sunflower, Cauliflower, and Agri-imagenet, it is apparent that the effectiveness of a model in detecting plant diseases is contingent on several characteristics, including accuracy, dataset, hyper parameters, overfitting, number of parameters, and model complexity. Identification of best transfer learning model involves analysis of the model dependency on the above-mentioned characteristics. In the way through prediction the best model to be considered is EfficientNetV2B2 and EfficientNetV2B3, since these models are acceptable in all the above characteristics.

Proposed system also predicts pathogen type for a specific disease which helps in taking precautionary measures accordingly with the remedy for the same disease in the audio output format. It is observed that "EfficientNetB" series and "EfficientNetV2" series have almost same accuracies as seen for sunflower dataset yet we go on with "EfficientNetV2" series because it is less bulky. The same applies when we take ConVNeXT models into consideration.

VI. FUTURE WORK

In future work, there is a potential to improve the computational time and accuracy of Transfer Learning models by fine-tuning their parameters and updating the final layers. Additionally, exploring techniques to reduce the computational complexity, such as model compression or pruning, can further enhance efficiency. Detection of plant diseases prior to its obviousness is one of the major challenging tasks. Dataset plays a major role and will be working on this issue in future. Moreover, extending the research to predict nutritional deficiencies in plants would be a valuable direction to pursue. Furthermore, it would be beneficial to consider the computational cost in terms of FLOPs (Floating Point Operations) and explore strategies to optimize the models for better efficiency while maintaining accuracy.

REFERENCES

[1] L. Li, S. Zhang, and B. Wang, "Plant disease detection and classification by deep learning—A review," *IEEE Access*, vol. 9, pp. 56683–56698, 2021, doi: 10.1109/access.2021.3069646.

[2] P. A. Nazarov, D. N. Baleev, M. I. Ivanova, L. M. Sokolova, and M. V. Karakozova, "Infectious plant diseases: Etiology, current status, problems and prospects in plant protection," *Acta Naturae*, vol. 12, no. 3, pp. 46–59, Oct. 2020, doi: 10.32607/actanaturae.11026.

[3] F. U. Rehman, N. Naz, and M. Kalsoom, "Plant diseases and food security: An overview," Dept. Plant Pathol., College Agricult., Inst. Food Sci. Technol., Government College Women Univ. Faisalabad, Pakistan, Univ. Sargodha, Pakistan, Tech. Rep., doi: 10.13140/RG.2.2.35480.52487.

[4] T. Gulya, R. Harveson, F. Mathew, C. Block, S. Thompson, H. Kandel, D. Berglund, J. Sandbakken, L. Kleingartner, and S. Markell, "Comprehensive disease survey of U.S. sunflower: Disease trends, research priorities and unanticipated impacts," *Plant Disease*, vol. 103, no. 4, pp. 601–618, Apr. 2019.

[5] G. B. Lucas, C. L. Campbell, and L. T. Lucas, "Causes of plant diseases," in *Introduction to Plant Diseases*. Boston, MA, USA: Springer, 1992, doi: 10.1007/978-1-4615-7294-7_2.

[6] T. Y. Toruño, I. Stergiopoulos, and G. Coaker, "Plant-pathogen effectors: Cellular probes interfering with plant defenses in spatial and temporal manners," *Annu. Rev. Phytopathol.*, vol. 54, no. 1, pp. 419–441, Aug. 2016, doi: 10.1146/annurev-phyto-080615-100204.

[7] K. Houston, M. R. Tucker, J. Chowdhury, N. Shirley, and A. Little, "The plant cell wall: A complex and dynamic structure as revealed by the responses of genes under stress conditions," *Frontiers Plant Sci.*, vol. 7, p. 984, Aug. 2016, doi: 10.3389/fpls.2016.00984.

[8] R. Anand, R. K. Mishra, and R. Khan, "Plant diseases detection using artificial intelligence," in *Application of Machine Learning in Agriculture*, M. A. Khan, R. Khan, and M. A. Ansari, Eds. New York, NY, USA: Academic, 2022, ch. 9, pp. 173–190. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780323905503000072, doi: 10.1016/B978-0-323-90550-3.00007-2.

[9] S. K. S. Durai and M. D. Shamili, "Smart farming using machine learning and deep learning techniques," *Decis. Analytics J.*, vol. 3, Jun. 2022, Art. no. 100041, doi: 10.1016/j.dajour.2022.100041.

[10] V. Singh and A. K. Misra, "Detection of plant leaf diseases using image segmentation and soft computing techniques," *Inf. Process. Agricult.*, vol. 4, no. 1, pp. 41–49, Mar. 2017, doi: 10.1016/j.inpa.2016.10.005.

[11] G. S. Sujawat, "Application of artificial intelligence in detection of diseases in plants: A survey," *Turkish J. Comput. Math. Educ. (TURCOMAT)*, vol. 12, no. 3, pp. 3301–3305, Apr. 2021, doi: 10.17762/turcomat.v12i3.1581.

[12] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, pp. 1–74, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[13] M. Iman, H. R. Arabnia, and K. Rasheed, "A review of deep transfer learning and recent advancements," *Technologies*, vol. 11, no. 2, p. 40, Mar. 2023, doi: 10.3390/technologies11020040.

[14] M. Shoaib, B. Shah, S. EI-Sappagh, A. Ali, A. Ullah, F. Alenezi, T. Gechev, T. Hussain, and F. Ali, "An advanced deep learning models-based plant disease detection: A review of recent research," *Frontiers Plant Sci.*, vol. 14, Mar. 2023, Art. no. 1158933, doi: 10.3389/fpls.2023.1158933.

[15] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," *Comput. Electron. Agricult.*, vol. 173, Jun. 2020, Art. no. 105393, doi: 10.1016/j.compag.2020.105393.

[16] A. Malik, G. Vaidya, V. Jagota, S. Eswaran, A. Sirohi, I. Batra, M. Rakhra, and E. Asenso, "Design and evaluation of a hybrid technique for detecting sunflower leaf disease using deep learning approach," *J. Food Qual.*, vol. 2022, pp. 1–12, Apr. 2022, doi: 10.1155/2022/9211700.

[17] J. Liu, F. Lv, and P. Di, "Identification of sunflower leaf diseases based on random forest algorithm," in *Proc. Int. Conf. Intell. Comput., Autom. Syst. (ICICAS)*, Dec. 2019, pp. 459–463, doi: 10.1109/icicas48597.2019.00102.

[18] U. Sara, A. Rajbongshi, R. Shakil, B. Akter, S. Sazzad, and M. S. Uddin, "An extensive sunflower dataset representation for successful identification and classification of sunflower diseases," *Data Brief*, vol. 42, Jun. 2022, Art. no. 108043, doi: 10.1016/j.dib.2022.108043.

[19] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes, "Deep learning for image-based cassava disease detection," *Frontiers Plant Sci.*, vol. 8, p. 1852, Oct. 2017, doi: 10.3389/fpls.2017.01852.

[20] H. Shuangping, S. Chao, Q. Long, X. Ma, and W. Wang, "Rice panicle blast identification method based on deep convolution neural network," *Trans. Chin. Soc. Agricult. Eng.*, vol. 33, no. 20, pp. 169–176, Oct. 2017.

[21] L. Tianyu and F. Quan, "Detecting grape leaves based on convolutional neural network," *J. Northwest Univ. Natural Sci. Ed.*, vol. 47, no. 4, pp. 505–512, Apr. 2015.

[22] Z. Chuanlei, Z. Shanwen, Y. Jucheng, S. Yancui, and C. Jia, "Apple leaf disease identification using genetic algorithm and correlation based feature selection method," *Int. J. Agricult. Biol. Eng.*, vol. 10, no. 2, pp. 74–83, 2017.

[23] B. Liu, Y. Zhang, D. He, and Y. Li, "Identification of apple leaf diseases based on deep convolutional neural networks," *Symmetry*, vol. 10, no. 1, p. 11, Dec. 2017, doi: 10.3390/sym10010011.

[24] J. Xue, L. Huang, B. Mu, K. Wang, Z. Li, H. Sun, H. Zhao, and Z. Li, "Detection of rotten fresh-cut cauliflowers based on machine vision technology and watershed segmentation method," *Amer. J. Biochem. Biotechnol.*, vol. 18, no. 2, pp. 155–167, Feb. 2022.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[27] A. Rajbongshi, U. Sara, B. Akter, R. Shakil, and S. Sazzad, "Sun flower fruits and leaves dataset for sunflower disease classification through machine learning and deep learning," *Mendeley Data, V1*, to be published, doi: 10.17632/b83hmrzth8.1.

[28] A. Rajbongshi, U. Sara, S. Sara, A. Rashiduzzaman, U. Bonna, and M. Shorif, "VegNet: An extensive dataset of cauliflower images to recognize the diseases using machine learning and deep learning models," *Mendeley Data, V3*, to be published, doi: 10.17632/t5sssfgn2v.3.

[29] A. Hosna, E. Merry, J. Gyalmo, Z. Alom, Z. Aung, and M. A. Azim, "Transfer learning: A friendly introduction," *J. Big Data*, vol. 9, no. 1, p. 102, Oct. 2022, doi: 10.1186/s40537-022-00652-w.

[30] C. Desai, "Image classification using transfer learning and deep learning,," *Int. J. Eng. Comput. Sci.*, vol. 10, no. 9, pp. 25394–25398, Sep. 2021, doi: 10.18535/ijecs/v10i9.4622.

[31] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807, doi: 10.1109/cvpr.2017.195.
[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015.
[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
[34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.
[35] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-V4, Inception-Resnet and the impact of residual connections on learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, no. 1, 2017, doi: 10.1609/aaai.v31i1.11231.
[36] A. G. Howard, *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. Mountain View, CA, USA: Google Inc., Apr. 2017.
[37] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269, doi: 10.1109/cvpr.2017.243.
[38] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8697–8710, doi: 10.1109/cvpr.2018.00907.
[39] J. Wang, Q. Liu, H. Xie, Z. Yang, and H. Zhou, "Boosted EfficientNet: Detection of lymph node metastases in breast cancer using convolutional neural networks," *Cancers*, vol. 13, no. 4, p. 661, Feb. 2021, doi: 10.3390/cancers13040661.
[40] M. Tan and Q. V. Le, "EfficientNetV2: Smaller models and faster training,," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1901–1910.
[41] Z. Liu, H. Mao, C. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11966–11976.
[42] X. Han, "Pre-trained models: Past, present and future," *AI Open*, vol. 2, pp. 225–250, Jan. 2021, doi: 10.1016/j.aiopen.2021.08.002.
[43] Y. Wang, D. Sun, K. Chen, F. Lai, and M. Chowdhury, "Egeria: Efficient DNN training with knowledge-guided layer freezing," in *Proc. 18th Eur. Conf. Comput. Syst.*, May 2023, pp. 851–866, doi: 10.1145/3552326.3587451.
[44] X. Xiao, M. T. Bamunu, C. Ji, J. Hu, and Y. Pan, "Fast deep learning training through intelligently freezing layers," in *Proc. Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Jul. 2019, pp. 1225–1232.
[45] D. R. Wilson and T. R. Martinez, "The need for small learning rates on large problems," in *Proc. IJCNN Int. Joint Conf. Neural Netw.*, Jul. 2001, pp. 115–119, doi: 10.1109/IJCNN.2001.939002.
[46] "Plant diseases and food security: An overview," Institut für Physik, Martin-Luther-Universität Halle-Wittenberg, Halle, Germany, Institut für Festkörpertheorie und -Optik, Friedrich-Schiller-Universität Jena, Jena, Germany.Accessed: Jun. 24, 2023. [Online]. Available: https://www.researchgate.net/publication/348872778_Plant_Diseases_and_Food_Security_An_Overview
[47] M. Hussain, J. J. Bird, and D. R. Faria, "A study on CNN transfer learning for image classification," in *Proc. Adv. Comput. Intell. Syst.*, 2018, pp. 191–202, doi: 10.1007/978-3-319-97982-3_16.
[48] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *Int. J. Data Mining Knowl. Manage. Process*, vol. 5, no. 2, pp. 1–11, Mar. 2015, doi: 10.5121/ijdkp.2015.5201.

K. P. ASHA RANI received the B.E. degree in information science and engineering from Visvesvaraya Technological University (VTU), Belagavi, India, and the M.Tech. degree in computer science and engineering from the National Institute of Technology (NITK), Karnataka. She is currently pursuing the Ph.D. degree in computer science and engineering with VTU.

She is an Assistant Professor with the Department of Computer Science and Engineering, Dr. Ambedkar Institute of Technology, Bengaluru, Karnataka, India. She has published papers in various reputed international journals and conferences. She has earned a patent. Her current research interests include the adoption of cognitive computing, artificial intelligence, machine learning, deep learning, expert systems, generative adversarial networks, computer vision for improving the efficiency of agricultural products, and in-farming techniques.

S. GOWRISHANKAR received the B.E. degree in computer science and engineering and the M.Tech. degree in software engineering from Visvesvaraya Technological University (VTU), Belagavi, India, the M.B.A. degree in marketing management from Indira Gandhi National Open University (IGNOU), New Delhi, India, and the Ph.D. degree in engineering from Jadavpur University, Kolkata, India.

He was a Senior Research Scientist and a Technical Lead with Honeywell Technology Solutions, Bengaluru, India. He is currently a Professor with the Department of Computer Science and Engineering, Dr. Ambedkar Institute of Technology, Bengaluru, Karnataka, India. He has coauthored a book and has earned three patents. He has published several papers in various reputed international journals and conferences. His current research interests include data science, including its technical aspects and its applications and implications. He has served on several panels of academic bodies for universities and autonomous colleges as a BOS and BOE member. He is serving as an editor and a reviewer for various prestigious international journals. He has delivered many keynote addresses and invited talks throughout India on a variety of topics related to computer science and engineering.