International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
DOI: https://doi.org/10.17762/ijritcc.v11i9s.7478
***Article Received**: 25 May 2023 **Revised:** 21 July 2023 **Accepted:** 06 August 2023*

EHFT: An Ensembled Hyperopt Fine-Tuned Neural Network for Disease Detection in Tomato Plants

Sri Silpa Padmanabhuni[1], Pradeepini Gera[2]
[1]Research Scholar, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation
Vaddeswaram, AP, India
e-mail: silpa.padmanabhuni@gmail.com
[2]Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation
Vaddeswaram, AP, India
e-mail: pradeepini_cse@kluniversity.in

Abstract—The identification of unhealthy plants in the crops at the early stage of cultivation helps for good farming. Unhealthy parts can be recognized using shape, color and texture, which are processed using feature extraction techniques. The feature extraction system stores the images in the matrix pixel format, which requires 3 channels for processing the images. Traditional neural networks utilize backpropagation techniques to adjust the random weights, which requires many resources while extracting a more significant number of features from a huge amount of data. These mechanisms also require more trainable parameters during the transformation of data from one layer to another. The proposed model implements the pre-trained model "RESNET152" (Residual Network), which is efficient for feature extraction and designs the last layer of the network as a "Tuned X-Gradient Boosting" ensemble algorithm for performing the binary classification of tomato leaves. RESNET can reduce computational resources because it implements residual blocks which fasten the learning rate by skipping a few connections in the network. The fine-tuned ensemble model helps the model identify the best parameters quickly. The learnable parameters are the essential elements of any ML model because they can easily identify the patterns associated with the different features. In the proposed model for feature extraction, pattern matching is the crucial step. Therefore, it is very necessary to tune the XGBOOST algorithm. Compared to the traditional approaches, the proposed model enhanced the accuracy performance in training and testing with 98.58% and 95.56%, correspondingly.

Keywords- Tuning Process, Pre-trained Model, ResNet-152V2, Boosting Algorithms, Feature Extraction, Residual Blocks, Transfer Learning

I. INTRODUCTION

The proposed research aims to classify healthy and unhealthy plants by customizing the neural networks using the concept of transfer learning. In the existing systems, the researchers have customized the neural networks by increasing the number of layers in the network to make the model learn complex features (edges, texture, objects & others). With the increase of layers, the model's performance is degrading regarding scalability. The model becomes expensive because of the additional operations in the increased layers, popularly known as the "Degradation Problem". So, the researchers need a model which can adjust to the nature of complexity without any saturation, i.e., not in a state to learn the identity function, which activates the gradient values of the neurons. This model was solved using ResNet, a pre-trained model. The efficiency of the ResNet lies in the Residual, i.e., skip and connector components because skip connection helps the model to find the shortest alternative paths to find their respective gradients. A neural network classifies the images by performing the matrix addition operation on the pixel elements, the weights of the layers are gradually increased because of weights multiplication, but in skip connection with the avoidance of a few layers, the multiplication of weight vectors is reduced. The mathematical representation of the identity block is represented as shown in equation (1)

$$Identity\_Block(Layer_i) = F(Input_i) + Input \quad -(1)$$

Where

Input represents the weight vector of the corresponding layer

In the proposed model, it is evident that the size of the images varies from layer to layer. Therefore, the model uses the convolution Residual to maintain the same dimensionality till the output layer and to activate the non-linear elements. The skip connections apply the ReLu activation, as shown in Figure 1.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
DOI: https://doi.org/10.17762/ijritcc.v11i9s.7478
***Article Received**: 25 May 2023 **Revised:** 21 July 2023 **Accepted:** 06 August 2023*

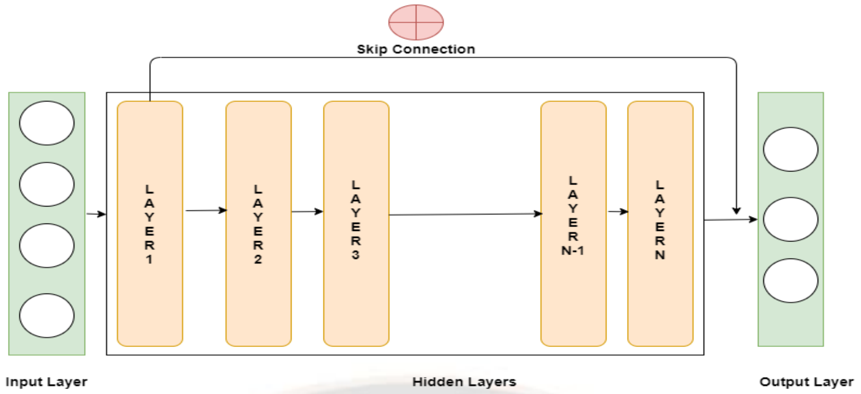


Figure 1: Skip Connections in ResNet Architecture

Residual Network, also known as ResNet, is a NN introduced by Kaiming He et al. in the year 2k15. There are five reasons for introducing this method. It is an ensemble method with a top-5 place in the ILSVRC 2k15. This method has achieved high performances and accuracy prediction in image & coco at identification, localization and segmentation. ResNet has overcome the VGG-16 by 28% high accuracy and performance. Training a method with 100 & 1k layers can be performed. For every layer, the previous layer's output is the Input for the next layer. With this method, skipping some layers will provide a different way to reach the data with later regions in NN. Based on many methods, the CNN performs well when it works with an image recognization system. This is considered a classic CNN method with maximum depth of ResNet152V2

ResNet152V2 is the following model for the ResNet method used in neural networks to extract feature methods. This method was used instead of VGG-19 in the CNN method for image recognization. This method holds with weights which are previously pre-trained methods, it helps to own the best accuracy which is faster than the traditional CNN method. This method has a five-layered structure, which highly resolves performances with good accuracy. The first layer consists of 3 components namely convolution, batch normalization, and max pooling layers. The dimensionality of next layers is computed as shown in equation (2)

$$next\_layer\_dimension = \frac{Original\_size + (2*Padding) - (filter\_number)}{Stride\_Size} + 1 \quad (2)$$

In the proposed model, the original image is uploaded with the size of 224, as per the requirement of ResNet. While performing convolution on image pixels, it is obvious that the corner pixels are operated only for one time as shown in figure 2.

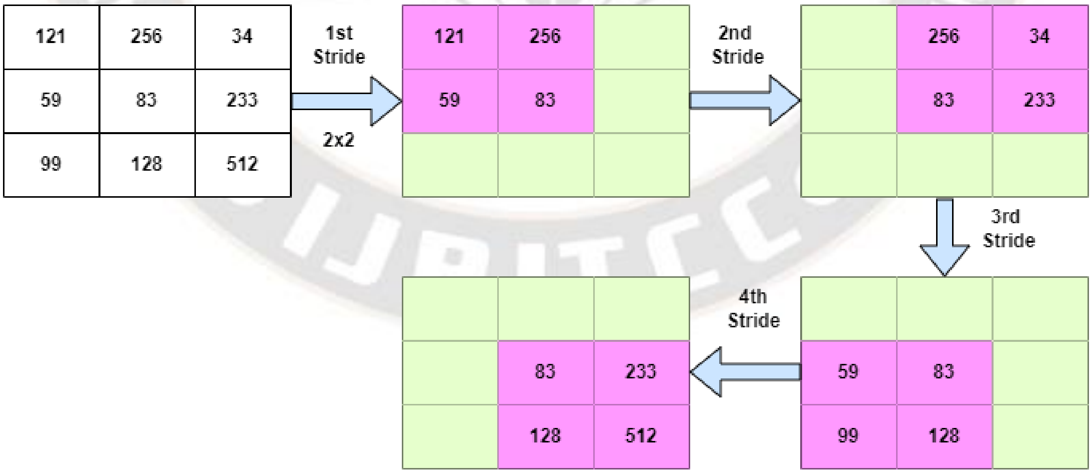


Figure 2: Stride Operations on Kernel Values

Images of the leaves has essential parameters at their borders only in-order to preserve the information at the borders the proposed model need to provide good padding value of size 5*5 to maintain the same size of images through out the network. Finally, the stride component decides the number of pixels that a convolution can jump. Higher the value of stride higher will be the missing rate of the information. So the model initialized the value to 2. The computation of equation 1 is presented in below section.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
DOI: https://doi.org/10.17762/ijritcc.v11i9s.7478
***Article Received**: 25 May 2023 **Revised:** 21 July 2023 **Accepted:** 06 August 2023*

$$next\_layer\_dimension = \frac{224+(2*5)-7}{2} + 1 = 114$$

This method was used in almost every method for its high efficiency and stability. The main advantage of this method is pre-training the method with high learning rate, which is accelerated training * converging for its high accuracy in rapid position. It will be more powerful when added before Deep Learning approaches, which can build efficiently and reliably, resulting in a high accuracy rate. The architecture of ResNet152V2 is presented in Figure 3.

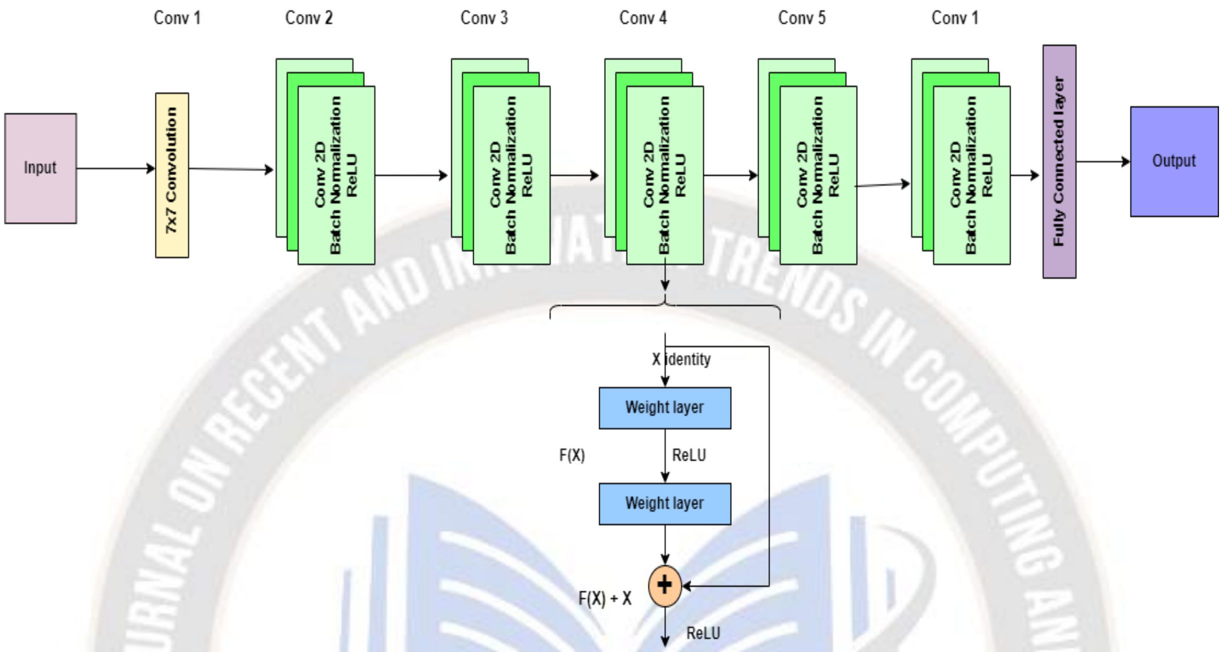


Figure 3: Architecture of ResNet-152V2

The following components are utilized as the building block of the ResNet:

a. Weight Decay: ResNet provides the solution to overfit problem by introducing the regularization "L2-Norm". It helps the networks o extract the features based on the similarity and context. The computation of similarity is shown in equation (3)

$$L2 - Norm(Input\_Pixels) = \sum_{i=0}^{n} \sqrt[2]{Input\_Pixels_i^2} - (3)$$

b. Activation Function: Before sending input to the hidden layers or output layer, model applies the activation function, which is in fact a transformer from non-linear to linear units. Utilizing activation functions helps the network to focus on important features and to ignore irrelevant data. When transferring values to some other layer or set of values, an activation function accepts the values generated by one level in the network and modifies them in a certain way as shown in equation (4)

$$ReLu(Input\_Pixels) = \sum_{i=0}^{n} max\left(0, \begin{bmatrix} Input\_Pixels_{1i} \\ Input\_Pixels_{2i} \\ Input\_Pixels_{ki} \end{bmatrix}\right) - (4)$$

c. Optimizer: The interpretability in a model is improved based on the optimization this is because of reducing parameters. Interpretability needs to be understood which is related to creating predictions. Where optimization improves the features related to the method. optimization uses regularization to the method for prevention of overfitting issues. Overfitting is considered when the method is highly complicated when the data is trained and results are very poor to create new data. This can be overcome by selecting the level of optimal complexity in the model. The equations & their sub computations are shown in equation (5)

a. momentum vector is computed as

$$moment(t) = \beta_{initial} * moment(t-1) + (1 - \beta_{initial}) * \delta(Input\_Pixel(t)) - 5\ (a)$$

b. Exponential weight is computed as

$$Exp\_weight(t) = max(\beta_{final} * Exp\_weight(t-1), \delta(Input\_Pixel(t))) - 5\ (b)$$

c. The updated value of the input pixel is computed as

$$Updated\_Input\_Pixel(t) = Input\_Pixel(t) - \left(\frac{\alpha}{1-\beta_{initial}}\right) * \frac{moment(t)}{Exp\_weight(t)} - (5)$$

In any neural network approach, the model's validity depends on obtaining the optimal weights for the input vectors.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
DOI: https://doi.org/10.17762/ijritcc.v11i9s.7478
Article Received: 25 May 2023 Revised: 21 July 2023 Accepted: 06 August 2023

Adjusting the weights and making the model to learn new parameters is tedious. Transfer learning solves this problem by getting the weights from the pre-trained model and also helps the model to become acquainted with the new environment based on those optimal weights. The proposed research uses the pre-trained model ResNet152V2 to get the optimal weights because the top layers of the network will never degrade their performance more than the lower layers, which means that the model's efficiency gradually increases. The top contributors to the network are presented in Table 1.

Table 1: Augmented Parameters of the ResNet

S.No	Name	Description
1	Include_top	If the value is true, the model considers a dense layer at the top
2	Weights	1. Random Initialization 2. Weights of Imagenet 3. .h5 file, which has optimal weights of the model
3	Input_shape	Image shape is required if parameter1 is false
4	Classes	Number of classes labels available in the dataset

The proposed model replaces the last dense layer of the pre-trained model with the hyper-tuned XGBoost classifier because it performs parallel processing using the attribute nthreads. It can also regularize the models based on alpha and lambda values. It prunes the paths that generate negative losses.

II. LITERATURE SURVEY

In [1], Mohit Agarwal et al. presented a unique way to identify diseases in tomato crops after examining leaf photos. Farmers could recognize the issues with identifying plant diseases because of this planned study. A CNN model that has been developed is used to categorize the photos. The Python library Augmentor was used for the augmentation process. The proposed approach design has a maximum of three pooling layers and three convolution layers. Each layer received a different set of filters. The loss has been calculated using the categorical cross-entropy technique. The results of the researchers' attempt to use the pre-trained values of the VGG-16 and add another output layer of 10 dimensions, each relating to a separate class of tomatoes, were incredibly dismal. The suggested model performs well across a wider range of classes because of its 42-layer deep design, which causes it to significantly outperform small class sizes when the distinguishing features are not visible or substantial.

In [2], Iftikhar Ahmad et al. focus on applying convolutional neural network (CNN) techniques to categorize and identify tomato leaf diseases. To identify and characterize tomato leaf illnesses, 4 CNN architectures—VGG-16, VGG-19, ResNet, and Inception V3—were used, coupled with feature extraction and parameter modification. To validate the models, actual site data is also gathered. Pre-trained models, Four tests use neural network models as feature extractors, while the remaining four use hyperparameter tuning and 10-fold cross-validation. Convolutional neural networks are used for feature extraction, while fully connected layers with softmax and ReLu are used for classification. Inception V3 surpassed all pre-trained models.

In [3], Huang provides a model based on deep Learning developed using photos of healthy and ill crop leaves from a public dataset. The major goal is to offer a program that, using leaf texture similarities, can determine the kind of crop disease. Segmentation is used to extract the important portions of the image selectively. Greyscale photos are created by converting segmented images to handle uneven lighting circumstances. Transfer learning is used for implementation using InceptionV3 and MobileNet. The output is changed from a binary type to a categorical type using label encoding. The variance in the input photos is added using an image data generator. To retrieve the model's result using the softmax activation function. The Adam optimizer is utilized with the category cross-entropy as a loss function. The function of crop detection is performed by the InceptionV3 model more effectively than MobileNet.

In [4], Atallah created a deep learning approach using the transfer learning method, using pre-trained weights taken from the well-known ResNet50 model. The precision of the detection has been improved through fine-tuning. By applying specific modifications to the photographs, such as rotation, height shift, width shift, horizontal flips, and shear range image augmentations were carried out. The suggested method is also used when the batch size is equal to 32, softmax activation, and the stochastic gradient descent (SGD) optimizer. More visual data is needed for the model to generalize as well as possible. The dataset is divided into 20% for validation and 80% for training. 38 completely connected layers are in charge of doing the categorization task. The suggested model does not include any layers that are fully connected.

In [5], Mohit Agarwal et al. devised a method based on convolution neural networks to detect illness in apple fruit. The model consists of three convolution layers, 3 max-pooling layers, and then two closely coupled layers. The standard machine learning approach, including Random Forest, Logistic Regression, LDA, Support Vector Machine (SVM), K-NN, and the pre-trained CNN model, VGGI6, and Inception V3, are also used. Data preprocessing techniques like inversion, the intensity of light changes, zoom in, and zoom out are used to boost the number of samples for each class. Over an NVIDIA DGX vlOO computer, the suggested CNN model was run. The feature set is produced using Hu-moments, HSV flattened histogram, Haralick features, and Local Binary pattern flattened histogram.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
DOI: https://doi.org/10.17762/ijritcc.v11i9s.7478
***Article Received**: 25 May 2023 **Revised:** 21 July 2023 **Accepted:** 06 August 2023*

The suggested model requires just around 7 seconds for testing, whereas existing pre-trained models require about 30 seconds due to the pre-training of more parameters.

In [6], Muhammad Mohsin Kabir et al. employed multilabel classification to generate many classes from the same Input at once. There were 28 illnesses affecting the six plants in the experimental dataset. Implementing the network infrastructure for reliable plant disease diagnosis is the main emphasis of this research project. Batch normalization and an activation layer come after each convolution. The architecture includes the binary cross-entropy loss function. The data for each image is reconfigured into 120x120 pixels. Potatoes, tomatoes, corn, rice, grapes, and apples are the six plants mixed for processing. It is transformed into a continuous set of nodes following the last convolution. Values are passed into a sigmoid activation function at each node.

In [7], Junde Chen et al. considered using the pre-trained models learned from the normal enormous datasets. It then transferred to the particular task educated by data, when studying pre-trained deep convolutional networks to detect the various diseases in plants' leaves. The VGGNet is trained priorly with ImageNet and the Inception module. The VGG-19 is used as a prototype and tweaked for creating new techniques. The generated networks are then trained and fine-tuned using the labelled example images. VGGNet already had transfer learning training on ImageNet and the Inception module. The two Inception modules, Concat and Inception, are then executed, following which a convolutional layer is added using batch normalization and Swish. Two consecutive 3x3 convolution layers are used in place of the original single 5x5 layers, maintaining the range of perceptual fields while utilizing fewer parameters.

In [8], Krishnamoorthy N et al. employed InceptionResNetV2, a CNN concept with the transfer learning strategy, to identify illnesses in photos of rice leaves. The transfer learning method is used to translate the knowledge into weights that are incorporated into the pre-trained InceptionResNetV2 architecture for the feature extraction phase. The input size for this architecture is 224x224x3. The deep learning algorithm has been enhanced to classify better the many types of illnesses affecting rice leaves. ReLU is the activation function employed. The third fully connected layer takes advantage of the Softmax activation feature. The possibility for each class is returned, with the target class having the highest likelihood. For multi-class classification issues, softmax activation is used, and the output possible values can be added up.

In [9], Abhinav Sagar and Dheeba Jacob utilized five pre-trained models as pre-trained weights: MobileNet, Inception v3, ResNet50, InceptionResNet v2, and DenseNet169. Data augmentation techniques like flipping, magnifying, shearing, and brightness modification are used to double the original dataset size virtually. When training takes a long time to provide a decent outcome, ModelCheckpoint is utilized. This novel method of classifying and automatically identifying plant diseases using leaf photos was investigated. The created model could differentiate between healthy leaf samples and several disorders that can be identified visually. There are now 19 different plant disease classifications in use. The last stage uses two thick layers with 2 neurons and 64 neutrons. Softmax is incorporated as an activation function in the last layer, which is used for classification. The selected loss function is binary cross-entropy.

In [10], Guan Wang et al propose deep learning methodologies for automatically identifying plant diseases using images. On the rescaled images, the model refinement and prediction are carried out. The training images are subjected to several random augmentations, such as arbitrary rotation, zooming, shearing, and flipping. The output characteristics of the last convolutional layer are used to train the newly created fully connected network. The accuracy decreases when the network depth goes beyond a certain point since there aren't enough training data for a model with excessive parameters. The SGD optimizer performs end-to-end training of the ANN model on the training dataset. The ResNet building blocks' residual mapping may be prematurely set to zero by the SGD optimizer, resulting in local optimization and inferior generalization in fine-grained classification. The optimized VGG16 model performs the best, with a 90.4% accuracy rate.

In [11], Akshai KP and J.Anithaused the dataset and trained the CNN model alongside previously trained models like VGG, DenseNet, and ResNet. CNN and transfer learning are employed in the working model to categorize various plant leaf diseases. The pooling layers decrease the number of parameters and the image size. A Softmax function is employed for the convolution layer, while a ReLU activation function is used for the output layer. The DenseNet-201 architecture with Imagenet weight is also employed. The training and validation datasets are divided from the dataset in an 80:20 ratio. Adam is the chosen optimizer, while categorical cross-entropy is the chosen loss function. Metrics, including the confusion matrix, accuracy, precision, F1-score, and recall, evaluate each model's effectiveness in classifying data. While the DenseNet model concatenates the output extracted features with the subsequent future feature maps, the ResNet model appends the result of the previous level to the output of the following layer. Every layer will acquire features from levels below it and pass along its saliency map to layers above it.

In [12], Priyanka Sahu et al. suggested using VGG16 pre-trained models and starting from scratch for training a CNN. The results are overfitted because of improper convergence. The overfitting issue is addressed with data augmentation. The leaves of the bean crop are divided into healthy and unhealthy

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
DOI: https://doi.org/10.17762/ijritcc.v11i9s.7478
***Article Received**: 25 May 2023 **Revised:** 21 July 2023 **Accepted:** 06 August 2023*

groups using a pre-trained VGG16 model. The classifier employed in the pre-trained model is changed to a fresh one. This classifier is positioned above the output nodes for classification or prediction. The data preprocessing method used for this feature extraction is slow and expensive to implement because it needs GPU assistance. By choosing the proper hyper-parameters, the pre-trained models were additionally fine-tuned. It performed significantly better than the earlier methods, with an accuracy of 96.06%. The dataset used for this investigation is relatively small.

Table 2: Analysis of the Existing Methodologies

Author	**Methodology**	**Merits**	**Demerits**	**Accuracy**
Mohit Agarwal	ToLeD using CNN	feed the VGG-16 pre-trained values, 42-layer deep design	Can improve using effective test records and should be tested on other than tomato	91.2%
Iftikhar Ahmad	CNN	actual site data is also used,parameter-tuning	Do not perform well on field data	89.06%
Huang	deep learning model	Greyscale photos, Adam optimizer, is utilized with the category cross-entropy	classification of images not captured in a controlled environment and images with multiple orientations.	90.45%
Atallah	ResNet50	softmax activation, and the stochastic gradient descent optimizer	No fully connected layers	92.80%
Tian	convolution neural networks	Data preprocessing based on several characteristics, 7 seconds for testing,	Performed only on apples	98%
Muhammad Mohsin Kabir	multilabel classification FCNN-LDA	28 illnesses affecting the six plants, binary cross-entropy loss function	Batch normalization	96.9%
Junde Chen et al.	VGGNet	Inception, and Concat, are executed.	Real-world application is needed.	97.7%
Krishnamoorthy N et al.	InceptionResNetV2	uses probability functions for the application of classification	Only for rice plant diseases.	95.67%
Abhinav Sagar et al.	Inception v3, InceptionResNet v2, ResNet50, MobileNet, and DenseNet169	It can identify 19 different diseases, and modelcheckpoint is also used.	It Consumes more time in detection...	94.3%
Guan Wang et al.	optimized VGG16 model	SGD optimizer and fine-grained classification.	random augmentation,	90.04%
Akshai KP et al.	DenseNet-201	Adam optimizer is used and included ResNet.	Classes can be increased with more parameters.	98.27%
Priyanka Sahu et al.	VGG16	Fine-tuning of the pre-trained model, data augmentation	A limited examination is performed, slow and costly.	96.06%

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
DOI: https://doi.org/10.17762/ijritcc.v11i9s.7478
***Article Received**: 25 May 2023 **Revised:** 21 July 2023 **Accepted:** 06 August 2023*

Research Gaps Identified: 1. Using the pre-trained model VGGNet, the model suffers from a gradient descent problem, which is claimed as "Vanishing Gradient"

2. Most researchers have implemented high-end models like the Xception module, AlexaNet, etc. All these models are highly expensive because of the more significant number of layers and operations involved in each layer.

3. Implementation of dense layers in the neural networks is good for extracting the features. However, the major task of the research is to classify the plants, which can be performed well using ensemble machine learning approaches.

III. PROPOSED METHODOLOGY

The proposed model initially preprocesses the images using traditional filters and image manipulation operators. Then it uses the pre-trained model ResNet-152 to extract the features. Finally, it passes the extracted features to the tuned gradient boosting algorithm for classifying the tomato leaves as "healthy" or "diseased". The entire process is presented in Figure 4.

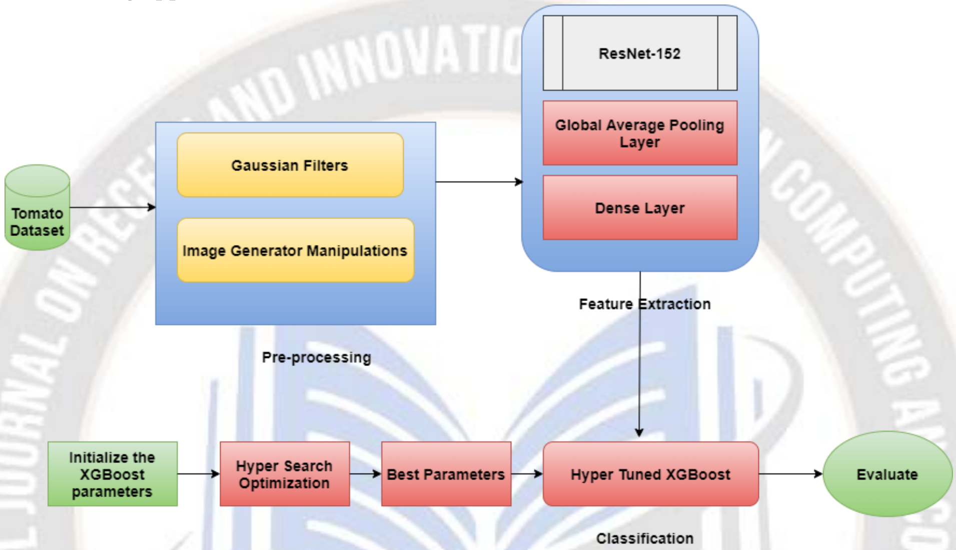


Figure 4: Block Diagram for Fine-Tuned Ensembled Neural Network

3.1. Modified Layered Architecture of Transfer Learning Process: In the transfer learning process, one can change the model's last layers to suit the proposed model's architecture. In general, ResNet-152 the name itself says that it has 152 layers, because of the residual components, a model may only use some of the layers. So, the proposed model has added a few more layers, like the global average pooling layer and dense layer, to extract features. The last layer of ResNet generates feature maps (extracted feature values) in vectors representing n-vectors for n-layers. These vector values may sometimes lead to prone values. To correct or regularize these values, the model needs to add more layers like batch normalizers, regularizes, flattening and dropouts.

a. Batch Normalizers: The model standardizes the input pixels using these layers by grouping them as batches. The goal of this layer is to distribute the pixel values to have standard deviation as 1 and mean as 0. This makes the neurons to learn the parameters of the model without depending on other neurons. The computation is presented in equation (6)

$$Norm\_Pixel_i = \frac{\sum_{i=0}^{n} W_i * Input\_Pixel_i + Bias - \mu_i}{\sigma_i} * \alpha + \beta - (6)$$

Where,

μ_i denotes the i^th pixel vector mean value

σ_i represents the i^th vector standard deviation

α , β are learning parameters

b. Dropouts: Dropouts disconnects some portion of neurons from the network to enhance the efficiency of the model. It limits only few neurons to learn by varying the ratio using the tuning parameters on weight constraints. The selection of ratio is presented in equation (7)

$$Weight\_Proportion = \frac{1-(learning\_rate)}{number\_neurons} * \text{loss\_rate-} (7)$$

This process makes the model more complicated, so the proposed research added a global pooling layer which constructs a single vector for each classification. The dense layer is added as the last layer of the model, with softmax as an activation function to categorize the extracted patterns. It builds a model based on input tensor and predictions of the categorizations. The modified architecture of the last layers of ResNet is presented in Figure 5.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
DOI: https://doi.org/10.17762/ijritcc.v11i9s.7478
***Article Received**: 25 May 2023 **Revised:** 21 July 2023 **Accepted:** 06 August 2023*

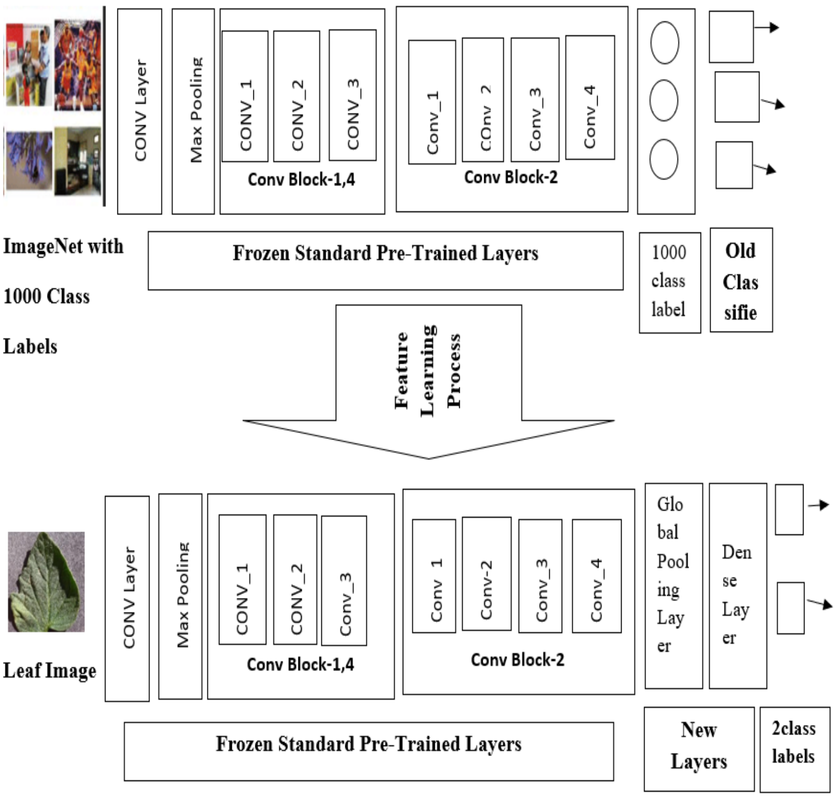


Figure 5: Modified Architecture of ResNet for Extracting the Features

The major advantage of any pre-training model lies in learning and adjusting the weights without starting from scratch. This residual network is constructed to solve the vanishing gradient problem by using skip connections between the convolution blocks. The generation of gradients using skip connections is presented in equation (8)

$$\partial(gradient^{i}) = \sum_{i=0}^{n} W_i^T * \partial(gradient^{i-1}) \odot \sigma_i\ (Normal\_Pixel_i) - (8)$$

The initial two elements of the RESNET are known as "Identity Block", but it receives the original Input and cannot skip any deviating values or saturation points. The model can skip one or more blocks simultaneously. The proposed system customized the last two layers as pooling and dense layers using the softmax function to generate output for binary classification. The mathematical representation of softmax is shown in equation in (9)

$$Activated\_Input_i = \frac{e^{input\_pixel_i}}{\sum_{i=0}^{n} e^{input\_pixel_i}} - (9)$$

3.2. Obtaining the Optimal Weights using ImageNet Dataset: During the training of neural networks, the initialization of weights plays a vital role in minimizing the loss. Before transfer learning approaches, researchers used gradient optimization algorithms to identify the heuristic weights of the input neurons. The model uses the transfer learning model, and benchmark functions are trained on various datasets and their optimal weights are used to train the new datasets. Because of this, the backpropagation iterations are reduced during the error minimization process. The model trains the dataset from the Kaggle repository but obtains the optimal weights by utilizing the pre-trained model on the "ImageNet" dataset. ImageNet is an annotated image developed by scientists to research real-time applications in the computer vision sector.

Algorithm for Feature Extraction:

Input: Tomato Leaf Dataset, TLD
Output: Train & Test Features
Begin:
1. source_dir←read_path("load_images")
2. for i in len(source_dir):
   a. initialize ratios of the train, test, and valid
   b. train[i],test[i]←splitfolders.prefix(source_dir[i],ratios)
   c. image_tensor←input(train[i],imagenet,channels=3,"weights.h5")
3. resnet_model←ResNet152V2(image_tensor)
4. global_pool←GlobalAveragePool(resnet_model)
5. build_model←Model(Dense(2,global_pool))
6. train_features, test_features←fit_generate.compile(build_model,generators)
End

3.3. Fine Tuning the XGBoost using the Hyper Optimization Search Space: It is important to control the parameters of the ML techniques, which are not dependent on data because their goal is to improve the model's efficiency. The proposed model

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
DOI: https://doi.org/10.17762/ijritcc.v11i9s.7478
***Article Received**: 25 May 2023 **Revised:** 21 July 2023 **Accepted:** 06 August 2023*

defines a search space by defining the parameters associated with XGBOOST and tunes them with an objective function as "Minimizing the Loss Function".

Algorithm for Fine-Tuned Classification Process
Input: Train Features Generated from ResNet
Output: Classification of Images (Healthy, Diseased)
Begin:
1. hxgb_space←{initialize estimators, max depth, objective,learning_rate, gamma}
2. for i in hxgb_space
   a. evaluate←min(trails,hxgb[i],iteartions=10)
3. test_labels←argmax(predictions(test_generator,evaluate))
4. train_labels←train_generator.classes
5. for id_num in random(train_labels):
   a. test_img[id_num]←load_img(labels[id_num],size)
   b. predict(test_img[id_num])
End

Table 3 presents the best parameters identified by the search space of the hyper objective function to perform the classification using the boosting algorithm. It describes each parameter's need to control the model to reach loss minimization and accuracy maximization.

Table 3: Best Parameters Identified by HyperOpt Search

S.No	Name	Description	Best Value
1	n_estimators	The number of weak learners or base estimators that we want to utilize with our dataset. The n estimator is 50 by default.	10
2	learning_rate	This setting is offered to reduce each classifier's overall contribution. It is assigned a value of 1 by default.	0.05
3	max_depth	The lengthiest path between both the parent node and the leaf node, or the degree of saturation of the tree.	7
4	Objective	It specifies the learning parameters	Logistic
5	Gamma	It minimizes the loss reduction by splitting the leaf nodes at desired rate	0.1

IV. RESULTS & DISCUSSION

In Figure 6, every epoch represents the loss and accuracy of the training and test data. The figure shows that the loss is gradually decreased, and the model has achieved nearly a good accuracy in testing and training. In the literature review, ResNet-50 has 92.8% accuracy, but the proposed model has achieved 98.58, nearly a 5.7% improvement.

```
Epoch 1/10
7/7 [==============================] - 99s 13s/step - loss: 0.7363 - accuracy: 0.6368 - val_loss: 0.5808 - val_accuracy: 0.6667
Epoch 2/10
7/7 [==============================] - 75s 12s/step - loss: 0.4961 - accuracy: 0.7547 - val_loss: 0.3916 - val_accuracy: 0.8889
Epoch 3/10
7/7 [==============================] - 83s 13s/step - loss: 0.3367 - accuracy: 0.8632 - val_loss: 0.2902 - val_accuracy: 0.9333
Epoch 4/10
7/7 [==============================] - 83s 12s/step - loss: 0.2642 - accuracy: 0.9198 - val_loss: 0.2366 - val_accuracy: 0.9556
Epoch 5/10
7/7 [==============================] - 82s 12s/step - loss: 0.2065 - accuracy: 0.9575 - val_loss: 0.1953 - val_accuracy: 0.9556
Epoch 6/10
7/7 [==============================] - 83s 12s/step - loss: 0.1733 - accuracy: 0.9670 - val_loss: 0.1825 - val_accuracy: 0.9333
Epoch 7/10
7/7 [==============================] - 83s 12s/step - loss: 0.1495 - accuracy: 0.9717 - val_loss: 0.1569 - val_accuracy: 0.9556
Epoch 8/10
7/7 [==============================] - 82s 12s/step - loss: 0.1313 - accuracy: 0.9858 - val_loss: 0.1535 - val_accuracy: 0.9333
Epoch 9/10
7/7 [==============================] - 91s 13s/step - loss: 0.1205 - accuracy: 1.0000 - val_loss: 0.1402 - val_accuracy: 0.9556
Epoch 10/10
7/7 [==============================] - 76s 11s/step - loss: 0.1086 - accuracy: 0.9858 - val_loss: 0.1318 - val_accuracy: 0.9556
```

Figure 6: Training & Testing Accuracy for Epoch

Figure 7 presents the visualization of the accuracy values obtained in the epochs for a better understanding of the model in terms of its efficiency. The model has linearly obtained the accuracy of training and testing data. Many existing models obtain good accuracy in the case of training data, but it needs to exhibit efficiency in the case of test data. In the visualization, epochs iterations are plotted on the X-axis and their level of accuracy measurement is plotted on Y-axis. Similar to accuracy analysis, figure 8 presents the loss visualization for each epoch. Training and testing loss have started linearly, but after some point, both losses are almost the same. This represents that loss incurred by testing data is approximately the same as training data, which is incurred in negligible amounts by the proposed model. In the visualization, epochs iterations are plotted on the X-axis and their level of loss measurement is plotted on Y-axis.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
DOI: https://doi.org/10.17762/ijritcc.v11i9s.7478
Article Received: 25 May 2023 Revised: 21 July 2023 Accepted: 06 August 2023

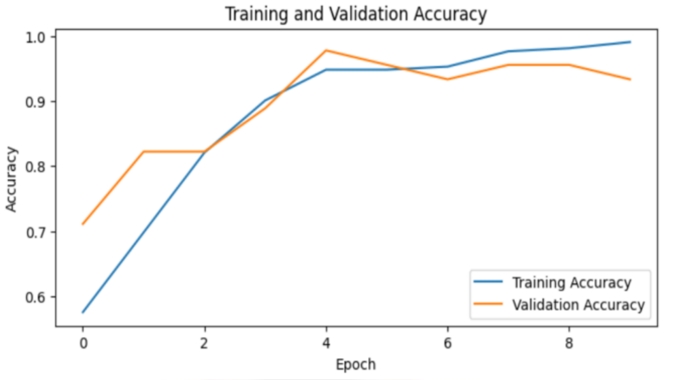


Figure 7: Accuracy Analysis of Fine-Tuned Model

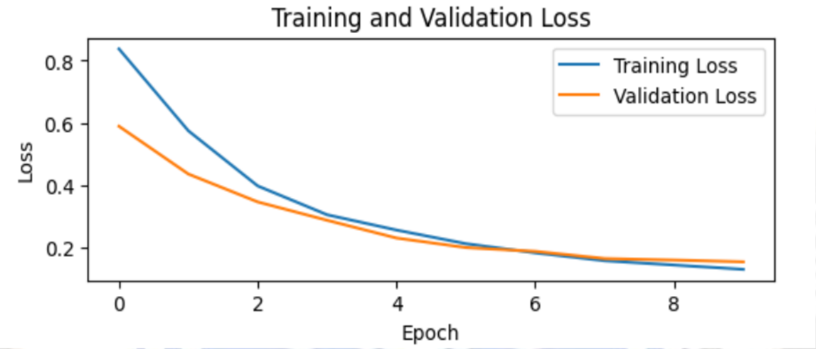
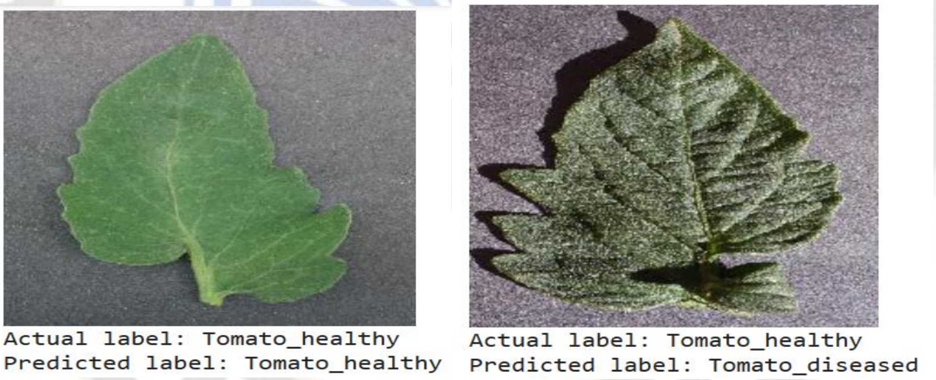


Figure 8: Loss Analysis of Fine-Tuned Model



(a)

(b)

Figure 9: Predictions of Fine-Tuned Model

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9s
DOI: https://doi.org/10.17762/ijritcc.v11i9s.7478
***Article Received**: 25 May 2023 **Revised:** 21 July 2023 **Accepted:** 06 August 2023*

Figure 9 represents a similar confusion matrix to represent each possible prediction made by the tuned model of XGBoost on the features extracted by the ResNet152. Figure 9(a) denotes the leaves marked as healthy in the data repository but predicted labels are also healthy and diseased. Similarly, Figure 9(b) denotes the leaves that are marked as disease along with the different possible predictions made by the model.

The classification of diseased and healthy tomato leaves can be evaluated using the confusion matrix as shown in the above diagram. These different combinations are evaluated using the 3 different metrics as shown in equation (10)

$$Fine\_Tuned\_Auccracy = \frac{TP\_Healthy+TN\_Diseased}{TP\_Healthy+TN\_Diseased+FP\_Healthy+FN\_Diseased} - 10\ (a)$$

$$Fine\_Tuned\_Recall = \frac{TP\_Healthy}{TP\_Healthy+FN\_Diseased} - 10\ (b)$$

$$Fine\_Tuned\_Precision = \frac{TP\_Healthy}{TP\_Healthy+FP\_Healthy} - 10\ (c)$$

Table 4 presents the metrics values obtained by the three different types of ResNet and proposed fine-tuned ResNet to prove the state of efficiency.

Table 4: Comparative Analysis of ResNet Approaches

	Accuracy	**Recall**	**Precision**	**F1-Score**
ResNet-50	92.8	90.3	89.1	89.5
Customized ResNet	93.14	93.2	95.4	94.3
Inception+ResNet	95.67	95.80	93.8	94.8
Fine Tuned ResNet	98.58	99.3	98.9	99.4

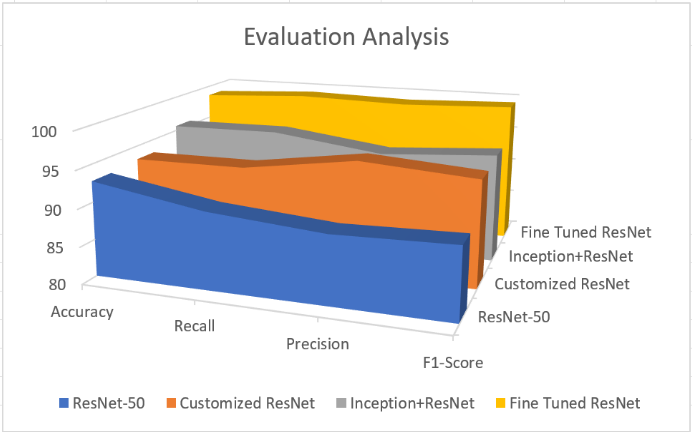


Figure 10: Evaluation Analysis on ResNet Approaches

Figure 10 presents the evaluation metrics of confusion matrix on different ResNet approaches studied from the survey. All the four important metrics of these approaches are compared to the proposed methodology and metrics are presented on X-axis and their percentages of efficiency on Y-axis.

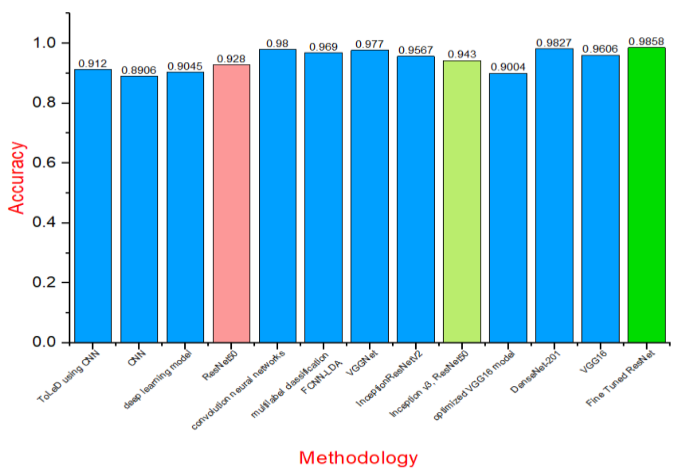


Figure 11: Analysis of Existing Approaches

Figure 11 represents the analysis of the existing systems studied in the literature survey with the proposed, i.e., fine-tuned model. The proposed model has fine-tuned the ResNet, which has obtained only 92.8% accuracy and it also observed another model which is a combination of Inception and ResNet, the accuracy is 94.3%. So compared to these models, the proposed model has high accuracy, which is marked at the end of the visualization graph. The model has achieved high accuracy over other mechanisms also. In the visualization, methodologies are plotted on the X-axis and their level of loss measurement is plotted on Y-axis.

V. CONCLUSION

Early disease detection plays a crucial role in smart agriculture because the quantity of crop production depends on it. Many researchers implemented computer vision techniques to identify diseases in plants. This research focuses on transfer learning enhancement by integrating the machine learning classifiers as the last layers of the pre-trained model because CNNs are good for feature extraction. However, when it comes to classification, the last layers, i.e., dense layers with either sigmoid or softmax functions, cannot handle the real-time environment images because of their learning rate over the controlled conditions. The proposed model has trained the network with the help of a real-time environment by storing the optimal values in the .h5 files. Then, it uses the concept of global pooling over the dense layers of ResNet152 and extracts the essential features. These essential features are trained by the tuned boosting algorithms to generate weak trees based on the misclassification rate and combine them in parallel to construct a stronger classifier than the sigmoid activation function, which focuses on activating input features rather than the relation between the features. The proposed model has obtained +5.7% better than traditional ResNet-50 and +4.2% better than integrated pre-trained models. In future work, tuning the supervised approaches using the optimization techniques helps the model to predict generalized diseases in plants irrespective of crop cultivation.

REFERENCES

[1] Agarwal, M., Singh, A., Arjaria, S., Sinha, A., & Gupta, S. (2020). ToLeD: Tomato Leaf Disease Detection using Convolution Neural Network. Procedia Computer Science, 167, 293-301. https://doi.org/10.1016/j.procs.2020.03.225

[2] Iftikhar Ahmad, Muhammad Hamid, Suhail Yousaf, Syed Tanveer Shah, Muhammad Ovais Ahmad, “Optimizing Pretrained Convolutional Neural Networks for Tomato Leaf Disease Detection”, Complexity, vol. 2020, Article ID 8812019, 6 pages, 2020. https://doi.org/10.1155/2020/8812019

[3] Huang, X., Chen, A., Zhou, G. et al. Tomato Leaf Disease Detection System Based on FC-SNDPN. Multimed Tools Appl 82, 2121–2144 (2023). https://doi.org/10.1007/s11042-021-11790-3

[4] Attallah, O. (2023). Tomato Leaf Disease Classification via Compact Convolutional Neural Networks with Transfer Learning and Feature Selection. Horticulturae, 9(2), 149. MDPI AG. Retrieved from http://dx.doi.org/10.3390/horticulturae9020149

[5] Tian, K., Zeng, J., Song, T., Li, Z., Evans, A. and Li, J. (2022) “Tomato leaf diseases recognition based on deep convolutional neural networks”, Journal of Agricultural Engineering, 54(1). doi: 10.4081/jae.2022.1432.

[6] Uddin, M. S., & Bansal, J. C. (Eds.). (2021). Computer Vision and Machine Learning in Agriculture. Algorithms for Intelligent Systems. doi:10.1007/978-981-33-6424-0

[7] Chen, J., Chen, J., Zhang, D., Sun, Y., & Nanehkaran, Y. A. (2020). Using deep transfer learning for image-based plant disease identification. Computers and Electronics in Agriculture, 173, 105393. doi:10.1016/j.compag.2020.105393

[8] N, K., Narasimha Prasad, L. V., Pavan Kumar, C. S., Subedi, B., Abraha, H. B., & V E, S. (2021). Rice leaf disease prediction using deep neural networks with transfer learning. Environmental Research, 198, 111275. doi:10.1016/j.envres.2021.111275

[9] Sagar, Abhinav, and Dheeba Jacob. "On using transfer learning for plant disease detection." BioRxiv (2021): 2020-05. https://doi.org/10.1101/2020.05.22.110957

[10] Guan Wang, Yu Sun, Jianxin Wang, "Automatic Image-Based Plant Disease Severity Estimation Using Deep Learning", Computational Intelligence and Neuroscience, vol. 2017, Article ID 2917536, 8 pages, 2017. https://doi.org/10.1155/2017/2917536

[11] KP, A., & Anitha, J. (2021). Plant disease classification using deep Learning. 2021 3rd International Conference on Signal Processing and Communication (ICPSC). doi:10.1109/icspc51351.2021.9451696

[12] Sahu, Priyanka, Anuradha Chug, Amit Prakash Singh, Dinesh Singh, and Ravinder Pal Singh. "Implementation of CNNs for crop diseases classification: a comparison of pre-trained model and training from scratch." IJCSNS 20, no. 10 (2020): 206. https://doi.org/10.22937/IJCSNS.2020.20.10.26

[13] Auma, G., Levi, S., Santos, M., Ji-hoon, P., & Tanaka, A. Predicting Stock Market Trends using Long Short-Term Memory Networks. Kuwait Journal of Machine Learning, 1(3). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/136

[14] Sri Silpa Padmanabhuni and Pradeepini Gera, “Synthetic Data Augmentation of Tomato Plant Leaf using Meta Intelligent Generative Adversarial Network: Milgan” International Journal of Advanced Computer Science and Applications(IJACSA), 13(6), 2022. http://dx.doi.org/10.14569/IJACSA.2022.0130628

[15] Swarajya Lakshmi V Papineni, Snigdha Yarlagadda, Harita Akkineni, A. Mallikarjuna Reddy. Big Data Analytics Applying the Fusion Approach of Multicriteria Decision Making with Deep Learning Algorithms International Journal of Engineering Trends and Technology, 69(1), 24-28, doi: 10.14445/22315381/IJETT-V69I1P204.

[16] Mallikarjuna Reddy, A., Rupa Kinnera, G., Chandrasekhara Reddy, T., Vishnu Murthy, G., et al., (2019), “Generating cancelable fingerprint template using triangular structures”,

Journal of Computational and Theoretical Nanoscience, Volume 16, Numbers 5-6, pp. 1951-1955(5), doi: https://doi.org/ 10.11 66 /jctn.2019.7830.

[17] P. S. Silpa et al., "Designing of Augmented Breast Cancer Data using Enhanced Firefly Algorithm," 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), 2022, pp. 759-767, doi: 10.1109/ICOSEC54921.2022.9951883

[18] A. Mallikarjuna Reddy, K. S. Reddy, M. Jayaram, N. Venkata Maha Lakshmi, Rajanikanth Aluvalu, T. R. Mahesh, V. Vinoth Kumar, D. Stalin Alex, "An Efficient Multilevel Thresholding Scheme for Heart Image Segmentation Using a Hybrid Generalized Adversarial Network", Journal of Sensors, vol. 2022, Article ID 4093658, 11 pages, 2022. https://doi.org/10.1155/2022/4093658.9

[19] Mallikarjuna A. Reddy, Sudheer K. Reddy, Santhosh C.N. Kumar, Srinivasa K. Reddy, “Leveraging bio-maximum inverse rank method for iris and palm recognition”, International Journal of Biometrics, 2022 Vol.14 No.3/4, pp.421 - 438, DOI: 10.1504/IJBM.2022.10048978.

[20] Srinivasa Reddi, P. ., & Sri Krishna, A. . (2023). CNN Implementing Transfer Learning for Facial Emotion Recognition. International Journal of Intelligent Systems and Applications in Engineering, 11(4s), 35–45. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2569

[21] Sudeepthi Govathoti, A Mallikarjuna Reddy, Deepthi Kamidi, G BalaKrishna, Sri Silpa Padmanabhuni and Pradeepini Gera, “Data Augmentation Techniques on Chilly Plants to Classify Healthy and Bacterial Blight Disease Leaves” International Journal of Advanced Computer Science and Applications(IJACSA), 13(6), 2022. http://dx.doi.org/10.14569/IJACSA.2022.0130618.

[22] V. NavyaSree, Y. Surarchitha, A. M. Reddy, B. Devi Sree, A. Anuhya and H. Jabeen, "Predicting the Risk Factor of Kidney Disease using Meta Classifiers," 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), 2022, pp. 1-6, doi: 10.1109/MysuruCon55714.2022.9972392.

[23] Kulkarni, O. (2018). Crop Disease Detection Using Deep Learning. 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). doi:10.1109/iccubea.2018.8697390

[24] Mukti, I. Z., & Biswas, D. (2019). Transfer Learning Based Plant Diseases Detection Using ResNet50. 2019 4th International Conference on Electrical Information and Communication Technology (EICT). doi:10.1109/eict48899.2019.9068805

[25] Dr. M. Varadharaj. (2019). Density Based Traffic Control System with Smart Sensing Of Emergency Vehicles. International Journal of New Practices in Management and Engineering, 8(02), 01 - 07. https://doi.org/10.17762/ijnpme.v8i02.75

[26] A. Mallikarjuna Reddy, V. Venkata Krishna, L. Sumalatha," Face recognition based on stable uniform patterns” International Journal of Engineering & Technology, Vol.7 ,No.(2),pp.626-634, 2018,doi: 10.14419/ijet.v7i2.9922 .

[27] Agarwal, M., Kaliyar, R. K., Singal, G., & Gupta, S. K. (2019). FCNN-LDA: A Faster Convolution Neural Network model for Leaf Disease identification on Apple’s leaf dataset. 2019 12th International Conference on Information & Communication Technology and Systems (ICTS). doi:10.1109/icts.2019.8850964