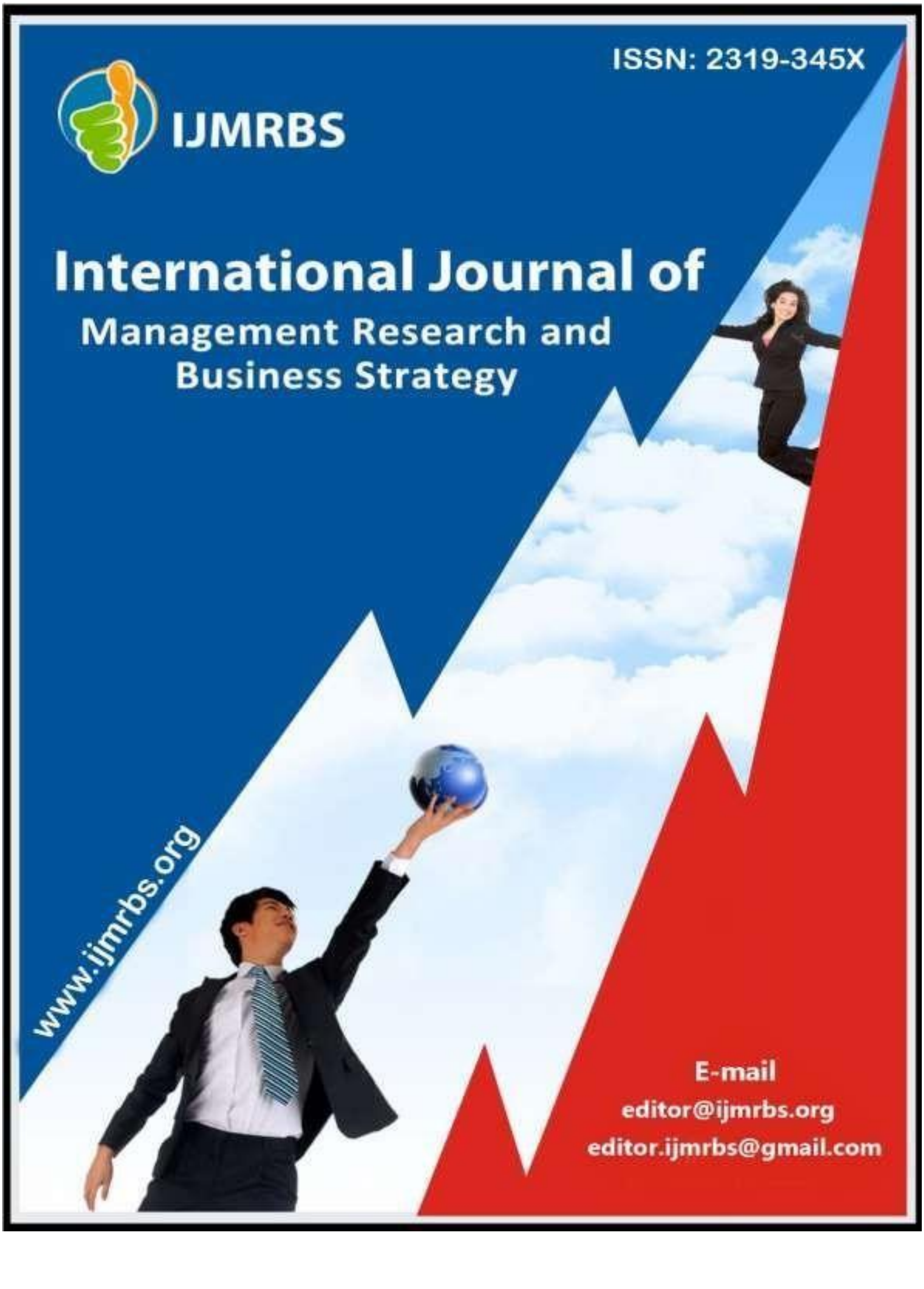

ISSN: 2319-345X
IJMRBS
International Journal of
Management Research and
Business Strategy
www.ijmrbs.org
E-mail
editor@ijmrbs.org
editor.ijmrbs@gmail.com

ISSN 2319-345X

www.ijmrbs.com

Vol. 9, Issue 4, 2020

INCEPTION V3 MODEL DETECTION OF TOMATO LEAF DISEASE

Authors: Dr. B. Rupkumar[1], (PhD) Department of EEE, SRM University
KB Jagadish kumar[2] (PhD), Sravan Vardhan Rao SOUDA [3] (PhD), Manohar Kumar [4], (PhD), K. Vijay Bhaskar [5], (PhD),
[2,3,4,5] Department of CSE, SRM University

Abstract- ***There are a lot of frequent tomato plant diseases that might significantly ruin crops. Common in conventional methods of disease identification, but time-consuming and subjective, are manual examinations and expert opinions. To get over these limitations, a deep learning model named Inception V3 was used to automate the process of disease detection in tomato leaves. The objective of this research is to detect tomato leaf diseases using the inception V3 model. Using a Wiener filter, the input image of the tomato leaves is cleaned up before processing starts. After that, the segmentation parts use fuzzy c means (FCM) on the image that has already been processed. This method of segmentation makes use of an updating technique (iterations) and a fuzzy partition matrix. The recovered image is appropriately labeled by Inception v3. We see an improvement in both training time and accuracy. The tomato leaf disease is identified by Atlast using Inception v3.Users may upload images of tomato leaves using the system's user-friendly internet software.At the time of upload, the image undergoes the system's preprocessing, segmentation, and classification stages. The last step is to give the user with a dedicated webpage that will display the anticipated disease. You can learn everything about the system and its features on the web app's site. The foundation of this project is the Python programming language.***

1.INTRODUCTION

A major factor influencing agricultural output yield is the emergence of crop diseases. It just takes a widespread disease epidemic to wipe out an entire harvest, resulting in permanent damage. Minor disease outbreaks may nonetheless have a major impact on agricultural production and quality, even in the absence of a widespread epidemic. One of the primary goals of improving agricultural production and quality to encourage economic development is the detection and categorization of crop diseases. Classified objects in generic image classification may vary from one another as they are members of coarse-grained meta-categories like Food-101, Oxford 102, Flowers, etc. Objects in fine-grained visual categorization (FGVC) seem homogeneous due to their shared subcategory membership. For example, FGVC is more challenging than regular picture categorization jobs. There is little distinction between tomato leaf mold and tomato mosaic virus, two diseases that affect the same crop (Figure 2), and there are other tomato illnesses as well. Consequently, FGVC is the rightful owner of the tomato disease classification research. We were unable to locate any FGVC research pertaining to agricultural crop disease picture classification, despite the fact that deep learning has made it easier to investigate several computer vision classification problems in the agricultural domain. Many studies continue to concentrate on coarse-grained crop image classification at this time because of the difficulty in locating and discovering judgment-value and numerous informative areas in crop health/disease pictures. The ability to precisely locate informative areas in images is, hence, fundamental to FGVC in the agricultural domain. The annotation of bird categories in CUB-200-2011 is one example of a prior study in the realm of computers that made use of fine-grained human annotations. Despite promising outcomes, these technologies are not widely used because to the high cost of the fine-grained human annotation required. In contrast, other approaches find informative areas using unsupervised learning strategies. Because human annotation is not taken into account, the model struggles to ensure that informative areas are given attention, leading to generally poor classification accuracy. In response to these issues, we provide LFC-Net, a new training paradigm for image classification of tomato health/diseases. It is composed of three networks:

Location, Feedback, and Classification. The three rectangular portions stand for the three areas of information. Additionally, the model suggests a self-supervision approach that may efficiently identify relevant tomato picture sections without human annotation of bounding boxes or portions. Regions of the picture that have been given a greater likelihood of being in a certain category should, in theory, include more of the image's features, therefore it stands to reason that the model would focus on these areas more. We suggest a loss function that

ISSN 2319-345X

www.ijmrbs.com

Vol. 9, Issue 4, 2020

corresponds to the new self-supervision mechanism, and we train the whole image category to be the genuine informative region category. This will assist find informative areas.

2.LITERATURE SURVEY

Qiufeng Wu *et al* [2022] suggested a novel approach to leaf disease detection using generative adversarial networks (GANs) to complement existing data. Using both actual photos as input to GoogLeNet and generated images enhanced by deep convolutional generative adversarial networks (DCGAN), this model is able to attain an average identification accuracy in the top 1%. The hyper-parameters, convolutional neural network architecture, and generative adversarial network choices were fine-tuned to provide a better model for training and testing 5 classes of tomato leaf photos. Also, DCGAN-generated pictures are diverse, which gives the model a leg up when it comes to generalizing, and they increase the data set size. Both the data variety and the recognition models' capacity for generalization are improved by this strategy. After the first layer of convolution, batch normalization is usually unnecessary.

Sabbir Ahmed *et al* [2022] This research proposes a lightweight method for disease detection in tomato leaves that is based on transfer learning. In order to improve categorization, it applies a powerful preprocessing technique to the leaf pictures, which improves them via lighting correction. For accurate feature extraction, our system employs a mixed model that incorporates both a pretrained MobileNetV2 architecture and a classifier network. Runtime augmentation is used instead of traditional augmentation methods to fix the class imbalance problem and prevent data leakage. An efficient preprocessing technique for improving classification by enhancing leaf pictures with light correction. On the other hand, more calculations are needed for a model to conduct inference when the value is greater.

Changjian Zhou *et al* [2021] established a redesigned residual dense network for disease detection in tomato leaves. To increase computation accuracy, information flow, and gradients, and to minimize the number of training process parameters, this hybrid deep learning model combines the best features of dense networks and deep residual networks. Due to the first use of the RDN model in image super resolution, a reorganization of the network architecture is required for classification tasks, including modifications to the input image features and hyper parameters. Improvements in computation accuracy, information flow, and gradients may be achieved by adjusting the parameters of the training process. Nevertheless, it carries a hefty amount of weight in the classification process, impacting the accuracy and efficiency of the classification.

Guofeng Yang *et al* [2020] proposed in this research is a Self-Supervised Collaborative Multi-Network. The LFC-Net concept is unique in that it integrates three networks—a Location network, a Feedback network, and a Classification network. Simultaneously, the model provides a self-supervision technique that can efficiently identify informative tomato picture sections without human annotation of bounding boxes or portions. We develop a new training paradigm based on the idea that images are most helpful when they are consistent across categories. Prior to optimizing iterations with the help of the Feedback network, the model's Location network finds important areas in the tomato picture. Without human annotation, our method successfully finds relevant parts of tomato images. Nevertheless, it carries a hefty amount of weight in the classification process, impacting the accuracy and efficiency of the classification.

Shengyi Zhao *et al* [2021] created a deep convolutional neural network with an attention mechanism that can adapt better to diagnose different types of illnesses affecting tomato leaves. Blocks for residuals and modules for attention extraction make up the bulk of the network architecture. The program is able to properly extract intricate characteristics of different illnesses. Model attains average detection accuracy on tomato leaf diseases dataset, according to extensive comparison trial findings. In comparison to competing models, it demonstrates that our approach excels in both network complexity and real-time performance. For accurate crop diagnosis in a real-world agricultural setting, this model offers a high-performance option. Having said that, training and detection speeds were both slowed down by the CNN models.

Xiao Chen *et al* [2020] discussed the use of a hybrid approach combining the Artificial Bee Colony algorithm (ABCK) with the Binary Wavelet Transform for the diagnosis of tree illnesses. Initially, the picture is denoised and improved using Binary Wavelet Transform in conjunction with Retinex (BWTR). This process eliminates noise and edges while preserving crucial texture details. The artificial bee colony algorithm (ABCK) was used

ISSN 2319-345X
www.ijmrbs.com
Vol. 9, Issue 4, 2020

to optimize KSW, and then the tomato leaves were isolated from the background. Lastly, the images were identified using the model of the Both-channel Residual Attention Network (B-ARNet). The images were recognized using the B-ARNet model, which stands for Both-channel Residual Attention Network. The initial picture of the tomato leaves included noise and feature ambiguity, thus incorrect feature extraction might still happen.

Suk-Ju Hong *et al* [2023] created a model that can predict whether tomato seeds would germinate using X-ray imaging. This study focused on growing X-ray-imaged seeds to the seedling stage and then classifying the seedlings based on their condition. Using X-ray image processing, we determined the seeds' structural integrity and tested a viability prediction model based on that. Additionally, viability prediction models based on convolutional neural networks (CNNs) were created and assessed. In terms of identifying germinated seeds from non-germinated ones, both models performed well. In contrast to the image processing-based model, the CNN-based model showed much higher accuracy in predicting seed viability.

Chunhua Hu *et al* [2019] offered a deep learning-based edge contour detection approach for ripe tomato identification. In order to identify specific fruits, this method effectively distinguishes between target tomatoes and overlapping tomatoes. This method results in several enhancements. To start, as compared to more conventional ways of determining which areas of tomatoes are ripe, deep learning takes far less time and extracts more detailed information. To differentiate ripe tomatoes from their backgrounds, we first utilize the HSV color space's Gaussian density function of H and S to segment tomato areas. Then, we apply erosion and dilation on the tomato body to separate nearby tomatoes and remove peripheral subpixels from all of the tomatoes that were identified. Thirdly, an adaptive threshold intuitionistic fuzzy set (IFS) approach was devised to detect the tomato's edge; it is effective in identifying fuzzy edges in overlapping areas. Due to the use of Faster R-CNN to identify and locate potential tomatoes, followed by processing just those tomatoes discovered by Faster R-CNN, this technique offers a significant speed benefit. Unfortunately, real-time robotic harvesting is a challenge for this system due to its computing intensity (it spends a lot of time on processing data that does not pertain to tomatoes).

Manuele Bettelli *et al* [2023] validated the use of biorister data for water stress classification and prediction in tomato plants. At first, we used Decision Trees and Random Forest, two classification algorithms, to attempt to differentiate between four distinct tomato plant stress states. Afterwards, we have used recurrent neural networks to forecast a plant's future state in a binary (water strained or not) and a four-state scenario. This method has been effectively used in smart irrigation scenarios that take place in the actual field. One major drawback of RNNs is that, due to the vanishing gradient problem, they cannot retain information about long-term dependencies.

Emre Ozbilge *et al* [2022] recommended a small convolutional neural network (CNN) for a disease detection job. A transfer learning comparison with the popular pre-trained ImageNet deep networks was the first step. The findings demonstrate that the proposed network outperforms deep network models trained with pre-learned information, and that complex, massive network topologies are unnecessary to get higher disease detection performance in tomatoes. In addition, data augmentation methods are used to enhance the recommended network's performance during training. Even though it uses the most inexpensive architecture, this method outperforms or is on par with state-of-the-art deep neural network techniques that utilize the Plant Village database.

Zhi-Feng Xu *et al* [2020] created a way to quickly identify tomatoes using enhanced YOLOv3-tiny. First, we strengthen the model's backbone network to make it more accurate; second, we apply picture enhancement to make the system better at detecting complicated situations. Lastly, to demonstrate the method's viability and logic, create many sets of comparison tests. There is an increase in detection time using the deep network model. But since there are more items and the picture is smaller, adhesion and occlusion are more likely to occur in a multi-object image, making identification more challenging.

Yo-Ping Huang *et al* [2020] showcased a fuzzy Mask R-CNN model that can detect cherry tomato ripeness levels automatically. To begin with, a fuzzy c-means model was used to automatically annotate the photographs while preserving the spatial information of different features in the image's foreground and background. To find the exact geometric edge locations of the tomatoes, a Hough transform approach was then performed. An image space annotation file was created for every data point. Secondly, in order to accurately identify each tomato, Mask R-CNN was used to train the annotated photos. Lastly, a color model based on hue saturation value and

ISSN 2319-345X

www.ijmrbs.com

Vol. 9, Issue 4, 2020

fuzzy inference rules were used to forecast when the tomatoes would be ready, hence avoiding pre-harvest abscission. Farmers may improve their decision-making and overall production efficiency and productivity with the use of automatic tomato harvesting software.

Yan Long *et al* [2022] suggested a way to use chlorophyll fluorescence imaging to detect drought stress in tomato seedlings. For four distinct drought stress levels, we measured chlorophyll fluorescence and took pictures of the resulting data. Next, three feature optimization algorithms—the Successive Projections Algorithm, the Iteratively Retains Informative Variables method, and the Variable Iterative Space Shrinkage approach—were used to choose crucial features. We found five shared parameters and used their related chlorophyll fluorescence pictures to make our selections. Drought stress classes were studied and analyzed using two kinds of picture characteristics: texture features and histogram features. In order to determine the different types of drought stress, we first determined the features' Pearson correlations. Then, we fed the features with the highest correlations into three different models: Linear Discriminant Analysis (LDA), Support Vector Machines (SVM), and k-Nearest Neighbor (KNN). This technology offered a fresh approach to tracking different types of drought stress and showed great promise for non-destructive identification of drought stress in plants.

Jun Guo *et al* [2020] revolutionized the process of harvesting tomatoes by creating a novel method that uses the YOLOv7 algorithm to identify when tomatoes are ripe. Tomatoes in the field may obscure target characteristics, making it difficult to identify them, or perhaps miss them altogether. Method for detecting tomatoes using enhanced YOLOv7. You can see the uniqueness down below. The first step in expanding the receptive field is the redesign of a new structure named ReplkDext. ReplkDext is included in the backbone prior to the final CBS layer. Furthermore, YOLOv7's revised head structure addresses the issue of poor FLOPS that conventional neural networks experience due to frequent memory access. The model achieves a balance between processing speed and detection accuracy by optimizing the structure between Concat and CBS in the brain using FasterNet. After the final ELANN-2 structure in the Head layer, ODConv is applied to enhance the convolution capabilities. Complex weather conditions, such as rain, fog, snow, and more, are ideal for tomato growth. The collection only contains photos captured in one setting because of the constraints of the experiments.

Chengyuan Song *et al* [2023] released TDPPL-Net, an integrated network model for real-time tomato recognition and point-picking that is lightweight and built on YOLOv5. To simplify the model, the algorithm first swaps out the YOLOv5 backbone for a four-group lightweight down sampling model that includes Ghost Conv and Ghost Bottleneck. To boost detection accuracy after each scale's feature map, it adds the attention mechanism Sim AM module. In order to decrease computing effort, the second step is to adopt the Spatial Pyramid Pooling Fast (SPPF) network structure. In the (Feature Pyramid Network and Path Aggregation Network) FPN+PAN structure, the convolutional layers are replaced by a depth-separable convolution. Last but not least, following hand-eye calibration, the depth information is translated into 3D coordinates under the robot arm coordinate system using the Intel Real Sense D435 camera in conjunction with the picking point in the middle of the bounding box. Without sacrificing accuracy, the TDPPL-Net speeds up detection on low-performance hardware. While YOLOv7 does a better job than YOLOv5s in terms of accuracy and speed, it is still not fast enough to compete with TDPPL-Net in real time.

Jongpyo Jun *et al* [2021] used the principles of 3D sensing, manipulation, and an end-effector to create an effective robot for picking tomatoes. This robot uses deep learning to identify tomatoes, then uses 3D coordinates to extract the target crop and 3D coordination to manage the manipulator's movements. Also, to secure individual tomatoes in clusters, a suction pad with a kirigami pattern—an element of the suction gripper—was created. Another component that was designed and tested was an aid unit-equipped scissor-shaped cutting module. This module helps to overcome structural restrictions and perform successful cutting. It is challenging to simulate soft elastic materials because of their malleability; they may twist, compress, or stretch.

Amora Amir *et al* [2021] used machine learning algorithms to predict how much sap would flow through a greenhouse's cherry tomato plants. A variety of machine learning (ML) methods, such as decision trees, random forests, support vector regression (SVR), elastic net regression (ENR), and least absolute shrinkage and selection operator (LASSO) are used in the proposed research. Various ML approaches are tested for their ability to make accurate predictions. Nevertheless, SVR's dependability is diminished due to the low correlation between predicted and actual values and the comparatively large mean square error. Greenhouse automation management, water usage efficiency, and production waste may all be improved with this methodology.

ISSN 2319-345X
www.ijmrbs.com
Vol. 9, Issue 4, 2020

Mayar Haggag ***et al*** **[2019]** put up the results and potential outcomes of a variety of successful AI approaches and initiatives. Unripe, ripe, and flawed (overripe and spoiled) are the three primary classifications that are taken into account. In addition, we build, test, and refine an experimental setup to validate the neural networks' computational output and evaluate the proposed algorithm's real-time performance. It is possible that our approach might make comparable fruit and vegetable sorting machines more accurate classifiers. But many of these methods rely on simplistic qualities like color separation, which makes it hard to tell ripe tomatoes apart from slightly overripe ones.

Kyamelia Roy ***et al*** **[2023]** created an innovative PCA DeepNet system for the detection of diseases in tomato leaves for use in agro-based industries. In this study, we provide a new approach to agro-based industry strengthening via the detection of tomato leaf diseases using Deep Neural Networks. The current innovative framework makes use of a hybrid of the traditional Principal Component Analysis (PCA) approach of machine learning and a specially-built Deep Neural Network called PCA DeepNet. Generative Adversarial Network (GAN) is another component of the hybridized framework that helps to get a decent mixing of datasets. Faster Region-Based Convolutional Neural Networks (F-RCNNs) are used to do the detection. By doing away with the possibility of human mistake, the study using PCA DeepNet makes illness diagnosis very quick and accurate. In addition, the proposed technique isn't applicable to any crops other than tomatoes that need a pipeline for disease detection; in reality, there are a number of crops that fall under this category.

Takefumi Hiragur ***et al*** **[2023]** another strategy, which involves the autonomous use of tiny drones to scout for flowers and pollinate them instead of bees. Using these drone technologies, researchers performed field studies. It is the drone's job to find blooms that are prepared to be pollinated. To help with flower identification, we built an AI classifier that uses machine learning to sort images of flowers. The drone uses its built-in AI classifier to navigate itself and look for flowers using its location and autonomous flying capabilities. During its search, the drone finds an appropriate bloom and then contacts it to pollinate it. The drone can really pollinate flowers thanks to an AI-powered flower detector, autonomous flight control for flower searching, and a pollination control device. Experimental validation led to the system's successful implementation as an autonomous system capable of pollination and flower search. On the other hand, this approach is predicated on the field being operational at the moment, which might restrict operating conditions and the efficiency of cultivation.

METHODOLOGY

3.1 EXISTING WORK

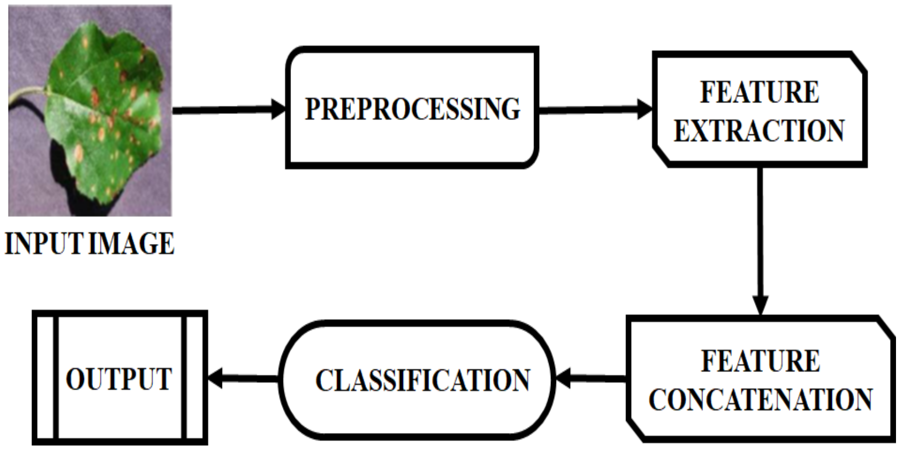


Figure 1 Block Diagram of Existing System

The current technique utilizes feature fusion of deep convolutional neural networks and local binary patterns to classify plant leaf diseases into many categories. To get high-level hidden feature representations, this study presents a new model of a lightweight deep convolutional neural network (CNN). To get information about the local texture in photos of plant leaves, the deep features are combined with classic, hand-crafted LBP features. Three datasets that are accessible to the public are used to train and test the current model: Apple Leaf, Tomato Leaf, and Grape Leaf. The current method improves validation accuracy on each of the three datasets. The experimental findings demonstrate that the current method offers a superior means of controlling plant diseases.

ISSN 2319-345X
www.ijmrbs.com
Vol. 9, Issue 4, 2020

3.2 DATASET

The suggested architecture was evaluated using three publicly available datasets from Plant Village that were obtained from a reputable source. The apple leaf dataset, which includes 3,171 pre-augmentation photos, is the first. A selection of these pictures. There are 630 pictures of scab, 621 of black rot, 275 of cedar rust, and 1645 of healthy. Increasing the size of the apple leaf dataset to 4645 photos using data augmentation methods, allowing it to handle unbalanced scenarios, while maintaining its biological qualities. We only employ the most widely used geometric transformations including scaling, translation, rotation, and vertical translation. A training dataset and a testing dataset are created from the data. To prevent over-fitting, the training dataset is split into two parts: training data (80% of the total) and validation data (20%). Another dataset that was used was the tomato Leaf dataset, which has 18,160 photos that were not enhanced in any way. There were 10 different levels. Each class provided a single example. There are several photos for each class. Training, validation, and testing were the same divisions applied to the data, just as in the apple leaf dataset. Also, prior to data augmentation, we utilized the grape leaf data collection, which had 4062 photos. The information is categorized into four groups. The grape leaf dataset was also augmented with 4639 more photos using data augmentation methods. The data was divided into three parts, the same as the datasets for tomato leaves and apples: training, validation, and testing.

3.3 ARCHITECTURE MODEL

There are two primary steps to the architecture's process of plant disease classification—feature extraction and fusion—and classification itself. Deep features and local binary pattern (LBP) features are fused together. A plant leaf's deep characteristics are extracted first by the architecture. In order to create the last unique characteristics of a leaf picture for plant disease classification, it merges the deep features with the local binary features. The deep features are captured using a CNN model, while the local texture information is successfully extracted by LBP. The outcome of merging these two sets of characteristics is the feature vector for plant disease samples. This study recommends using the flattened layer in the DCNN model to directly combine the deep features with the retrieved LBP features. In order to classify the data, the first fully connected layer receives the aggregated characteristics and applies softmax.

3.4 DEEP CNN ARCHITECTURE

The elimination of a feature engineering process is the primary benefit of using deep CNN for picture classification. Architecture of deep convolutional neural networks for disease classification of plant leaves. A model with three convolutional layers (with kernel size of 3 × 3), three max-pooling layers (with kernel size of 2 × 2), and four dense (fully connected) layers is used to pre-process the captured images through image filtering, image sharpening, and resizing. The images are fed into the model in a 64× 64 size. In order to extract features from photos, the convolution layers are used. Consequently, the suggested model's first convolutional layer has 32 filters, while the final convolutional layer includes 128 filters. According to the research, the model's performance changes as the filter sizes are raised incrementally.For tomato leaf diseases, there are ten groups, while for apple and grape leaf diseases, there are four. One popular method for classifying data into several categories is the softmax function. High accuracy was attained while drastically cutting down on training parameters and iteration time via the use of deeper convolutional layers and appropriately adjusting the parameters of the deep convolutional neural network (CNN) model.

3.5 LOCAL BINARY PATTERN

One of the strongest texture descriptors is local binary patterns (LBP). Its primary function is to depict an image's local characteristics, or its most salient elements. The traditional LBP operator is described as a window with dimensions of 3 × 3. A threshold is established using the middle pixel of this window. A pixel is marked as having a value of 0 if its surrounding pixel's value is lower than the threshold value. If not, it is marked as 1. An 8-bit binary number may be generated using this method.

ISSN 2319-345X
www.ijmrbs.com
Vol. 9, Issue 4, 2020

3.6 DRAWBACKS OF EXISTING SYSTEM

- Using just input data, convolutional neural networks (CNNs) are able to automatically learn hierarchical features. Nevertheless, a substantial quantity of labelled data and substantial computer resources are necessary for training deep CNNs.
- Local binary pattern based texture information is what LBP excels at capturing, but it has a hard time grasping the more nuanced spatial connections and semantic elements seen in photos of plant leaves.

4.PROPOSED SYSTEM

4.1PROPOSED WORK

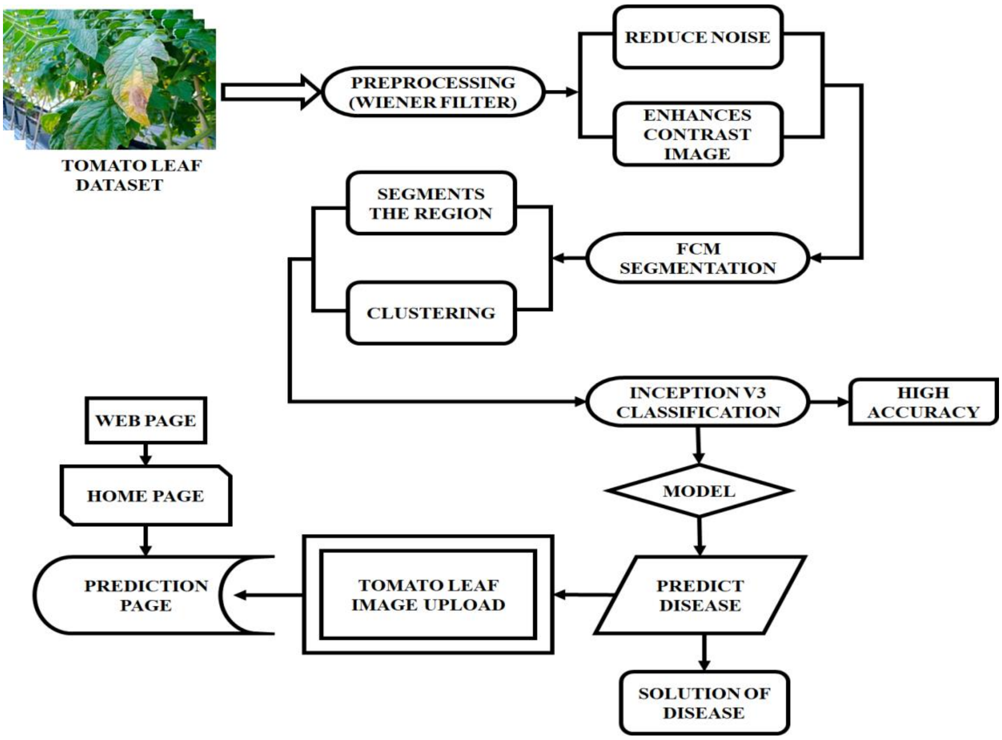


Figure 2 Block Diagram of Proposed System

This system suggests a method to identify tomato leaf diseases by using the Inception V3 model. In the preprocessing stage, the input picture of a tomato leaf is cleaned up by using a Wiener filter. The segmentation portions then apply fuzzy c means (FCM) to the pre-processed image. Iteratively updating the matrix via fuzzy partitioning is at the heart of this segmentation technique. The retrieved picture is appropriately classified by Inception v3. It reduces training time while improving accuracy. The tomato leaf disease is now accurately identified by Atlast with Inception v3. Users may simply submit photos of tomato leaves using the system's intuitive interface. A web application is used to implement the system. The system's functioning starts with the upload of a picture and continues via preprocessing, segmentation, and classification. The web app also has a site that describes the system and its capabilities.

All things considered, this technique is a practical and easy way to forecast tomato leaf illnesses, which may help farmers catch them early and take action. This project is implemented using Python software.

4.2 PREPROCESSING

The purpose of pre-processing is to improve the picture data by reducing undesired distortions or by enhancing certain image properties that are crucial for further processing, even when the images undergo geometric alterations.

ISSN 2319-345X
www.ijmrbs.com
Vol. 9, Issue 4, 2020

4.3 STEPS FOR PREPROCESSING

Resize

Although square input photos are required by many model designs, very few devices actually record square images. Stretching the image's dimensions to make it square is one option, while maintaining the image's aspect ratio and adding pixels to fill in the newly generated "dead space" is another. In addition, the input photos could be of different sizes, with some being smaller than the target input size.

Grayscale

- One kind of picture alteration is color changes; these may be applied to both the train and test pictures, or they can be used as augmentations in the training set alone.
- Changing the color to grayscale is a common practice for all photos. Even though our first instinct is to say, "more signal is always better; we should show the model color," grayscale photos may actually reveal faster model performance.
- Furthermore, color may not always be the most important factor for a model. Your model will learn more generic qualities about an item that are not dependent on color if you utilize greyscale instead of worrying about collecting photos for every color of an object.
- RGB values are used to represent color pictures, while the black-and-white spectrum is used to represent grayscale images. This implies that our model can manage with a single matrix per picture instead of three, which is great for CNNs.

4.4 WIENER FILTER

To have a better idea of the distorted signal's underlying signal, the Wiener filter may remove the noise. A more statistical explanation of the idea is provided in the page on the minimal mean square error (MMSE) estimator, which is relevant to the Wiener filter since it is based on statistics. The Wiener filter is designed to take a related signal as input and filter it to provide an estimate of an unknown signal. Its purpose is to do this statistically. As an example, the known signal may be an interest signal that is unknown but has been contaminated by additive noise. To have a better idea of the distorted signal's underlying signal, the Wiener filter may remove the noise. A desired frequency response is often achieved when designing deterministic filters. But the Wiener filter is designed in a different way. The goal is to find the linear time-invariant filter that produces an output that closely resembles the original signal, given that you are aware of the spectral features of both the signal and the noise. The following are the characteristics of Wiener filters:

Assumption

Linear stationary stochastic processes with established spectral properties or autocorrelation and cross-correlation are signal and (additive) noise.

Requirement

It is possible to omit the constraint that the filter be physically realizable or causal, but then the solution would not be causal.

Performance criterion

Mean squared error (MMSE) as a minimum

4.5 SEGMENTATION AND ITS TYPES

For the purpose of image segmentation, a mask or labelled image is used to divide a picture into smaller, more manageable pieces. Instead of processing the whole picture, you may focus on processing the relevant parts by separating it into segments.

Semantic segmentation

- In semantic segmentation, each pixel in an input picture is given a class label. You may learn about the many approaches to computer vision and how they vary from one another by consulting the materials provided below.

ISSN 2319-345X

www.ijmrbs.com

Vol. 9, Issue 4, 2020

- Each pixel in a picture is given a name that specifies its content, such "road," "tree," or "building," in semantic segmentation, a kind of image segmentation.

Instance segmentation

Assigning a distinct name to each item in the image and being able to determine its bounds are two advantages of instance segmentation, a kind of picture segmentation, over other methods. The capacity to detect and isolate certain visual elements within a picture is known as instance segmentation. More and more, it is finding uses in other industries, and these uses are having an impact on people's everyday life. In most cases, the Mask R-CNN Architecture model is used to construct instance segmentation.

It is possible to classify instance segmentation techniques into two broad groups.

- One approach is the bottom-up technique, which involves identifying individual pixels in a picture before assembling them into a shape.
- The top-down approach: find everything in the picture, pick out certain items, and then segment them.

Panoptic segmentation

Panoptic segmentation models are able to analyze the Improved to detect semantic information in addition to identifying and segmenting specific items in a picture. The idea behind this is that semantic segmentation and instance segmentation work hand in hand; one may utilize the other to find and segment specific objects in a scene, while semantic segmentation helps you locate and understand the scene's semantic information. The accuracy and detail achieved by panoptic segmentation are superior than those of each job performed alone.

4.5.1 FUZZY C MEANS SEGMENTATION

picture Objects are created during the segmentation process by dividing a picture into smaller groupings of pixels. To simplify the picture, this system employs the Fuzzy-C-Means method for segmentation. A well-liked data clustering technique, fuzzy c-means algorithm (FCM) or fuzzy ISODATA assigns a membership grade to each data point, indicating the extent to which it belongs to a cluster. In order to minimize a distance-based cost function, FCM divides a set of n vectors, X j, j = 1,..., n, into c fuzzy groups and locates a cluster center inside each group. Fuzzy clustering enables an item to be partially associated with all clusters and accounts for cluster overlap. A data point may have a degree of membership between ° and 1, meaning it can belong to any category.

By minimizing the weighted within group sum of squared error objective function J FCM, the technique, an iterative clustering approach, creates an ideal c partition. Fuzzy c-partitions, which are the end product of fuzzy clustering, may be described by the following equation:

$$J_{FCM} = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^q \, d^2(x_k, v_i) \qquad (4.1)$$

The data set in the p-dimensional vector space is represented by X = {x1, x2, • • •, xn} ≈ R p.The data set has n elements.c is the count of clusters where 2 ≤ c < n.u_ik represents the extent to which x_k is a member of the i-th cluster.Does each fuzzy membership have a weighted exponent, q?d^2 (x_k,v_i) is a distance measure between item x_k and cluster Centrev_i, where v_i is the prototype of the Centre of cluster i.

An algorithm that iteratively finds the best solution in FCM has been developed. One method that uses the fuzzy clustering paradigm is the FCM. To reduce the murkiness of within-cluster variation, it revises the membership levels in a certain way. The most common way for image processing pixels to be assigned multiclass membership values is via FCM. After we have the memberships of the data points, like pixels, for each cluster, we may put pixels in the class with the highest membership. Fuzzy c-mean is an unsupervised method since it doesn't need a starting set of training data to generate membership functions for further processing.

RESULT AND DISCUSSIONS

International Journal of Management Research and Business Strategy

ISSN 2319-345X

www.ijmrbs.com

Vol. 9, Issue 4, 2020

5.1 INTRODUCTION
PYTHON

Python is a general-purpose advanced computer language. Its design philosophy places a premium on code readability by making extensive use of indentation. Dynamic typing and garbage collection are features of Python. Functional, object-oriented, and structured programming paradigms are among those it supports. Python has a plethora of integrated development environments (IDEs). Python is very popular in many different industries, including data science, data analysis, data engineering, web development, software development, and machine learning. Our blog post on the basics of the popular programming language Python may teach you everything you need to know about it.

ANACONDA JUPYTER

To carry out this project, the python programming language is used. An environment compatible with several versions of Python and packages may be easily set up using the Anaconda tool. A graphical user interface (GUI) program named Anaconda Navigator is included with the Anaconda distribution. It streamlines the procedure for setting up and running programs like Jupyter Notebook. A fully self-contained environment is what you get with the Anaconda Python setup. You can easily install packages without modifying your system's Python installation. An environment compatible with several versions of Python and packages may be easily set up using the Anaconda tool. Anaconda also allows you to install, remove, and upgrade packages in your project environments. Anaconda and Jupyter are very important to data scientists and Python programmers. Many important libraries are pre-installed in Python installations like Anaconda, making them ideal for scientific computing and data processing. Jupyter, on the other hand, offers an interactive environment primarily via its notebook interface, making it simple to execute code, see data, and analyze results.

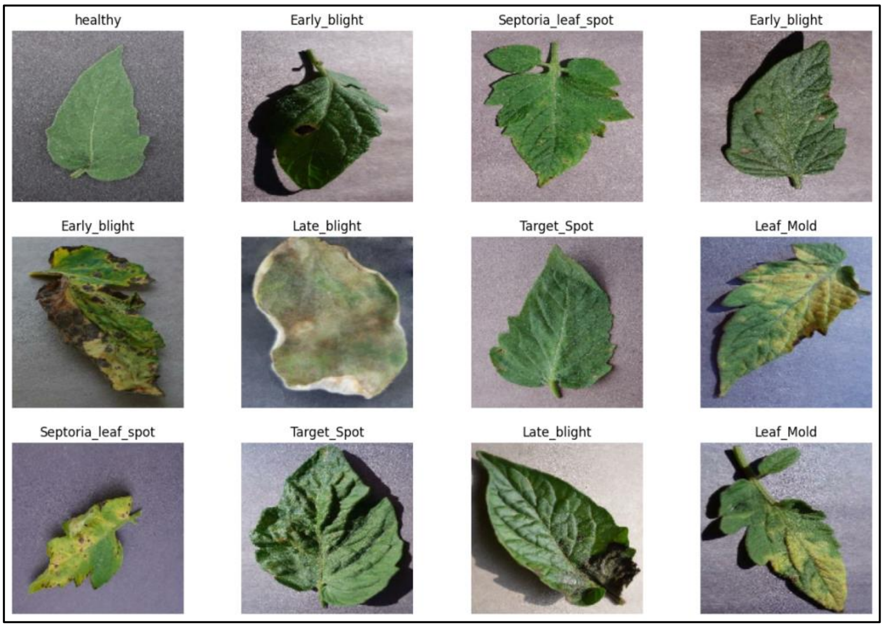


Figure 3 Input Dataset

The input dataset is shown in Figure 3. Images of many tomato leaf diseases are included in this collection. Images representing different types of tomato leaf diseases (e.g., early blight, late blight, bacterial spot, etc.) make up the dataset used to train the algorithm. To facilitate supervised learning, each picture is tagged with the illness class to which it corresponds.

ISSN 2319-345X
www.ijmrbs.com
Vol. 9, Issue 4, 2020

Figure 4 Classes of the Data

In Figure 4, you can see the data classes. In the context of identifying illnesses in tomato leaves, the data classes usually correlate to several diseases that might impact tomato plants. Machine learning algorithms that attempt to detect and categorize illnesses using visual traits seen in leaf pictures are trained and evaluated using these classes, which stand for separate categories or labels.

.

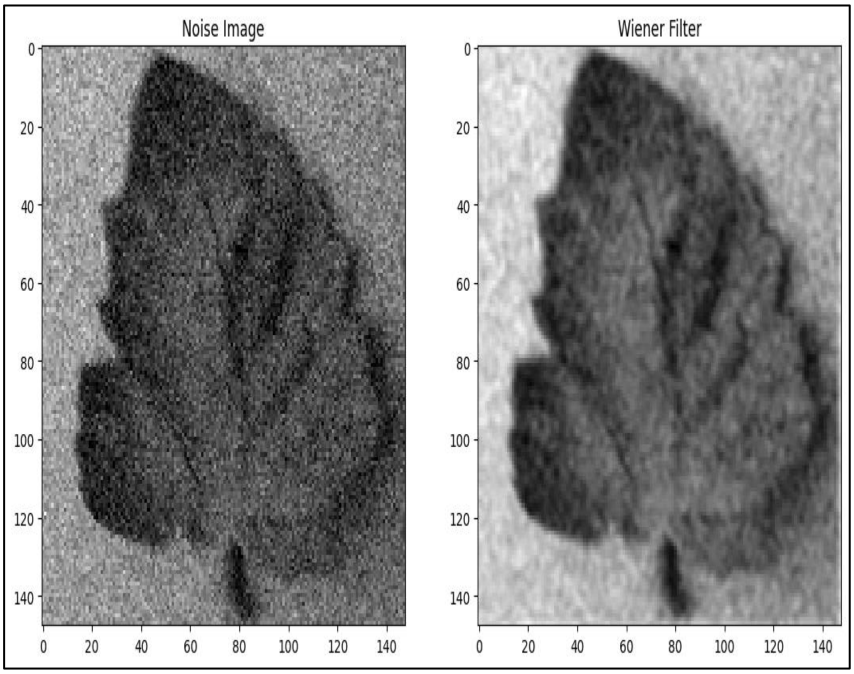


Figure 5 Wiener Filter Process

Figure 5 displays the wiener filter. To lessen the amount of background noise in the pictures of tomato leaves, a spatial-domain technique called the Wiener filter is used. The Wiener filter improves picture clarity and gets it ready for analysis by adaptively filtering it according to local signal-to-noise ratio properties.

ISSN 2319-345X
www.ijmrbs.com
Vol. 9, Issue 4, 2020

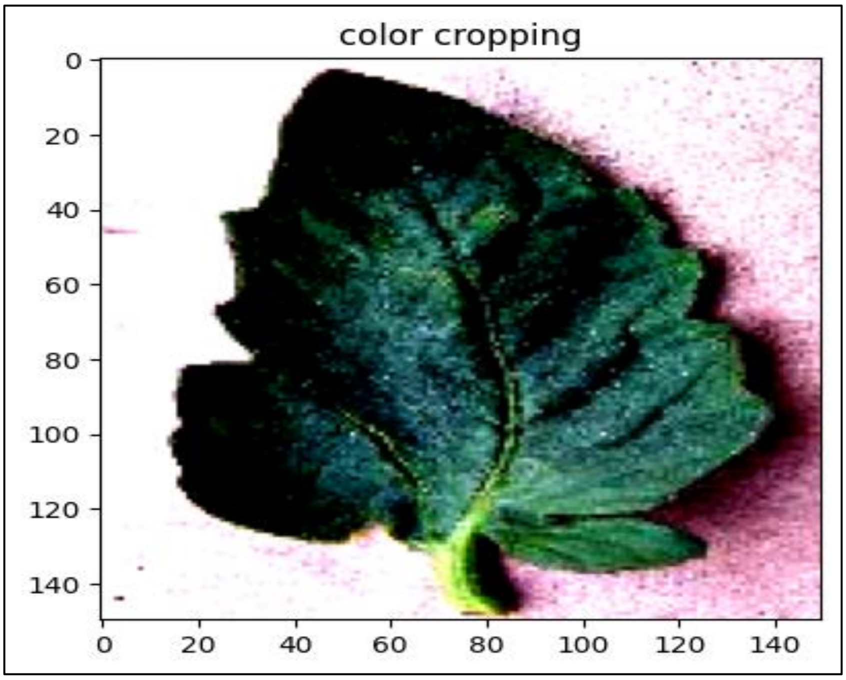


Figure 6 Color Cropping

Figure 6 shows the results of color cropping. The tomato leaves are the only objects that are cropped using color in order to extract them from the preprocessed photos. This optimizes computational efficiency and accuracy by ensuring that the next segmentation and classification algorithms are applied only to the relevant portions of the picture.

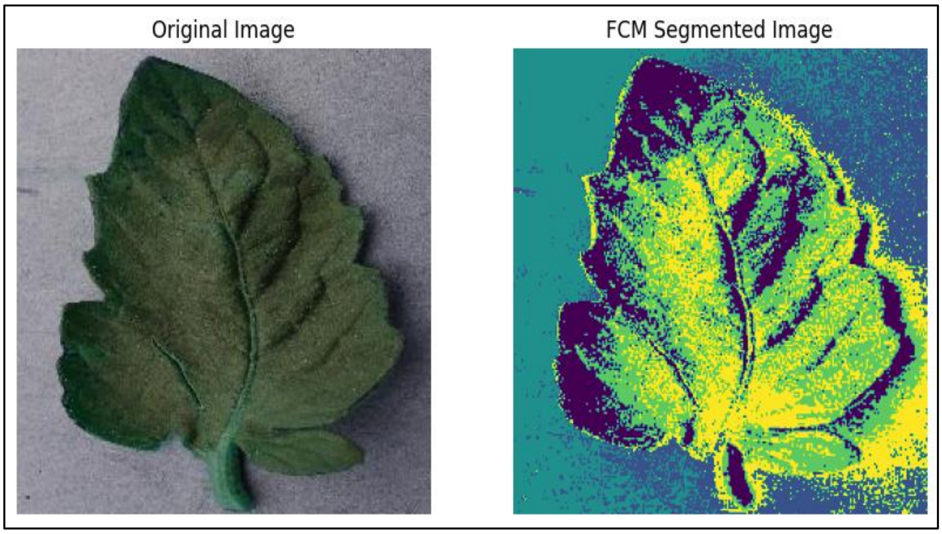


Figure 7 FCM Segmentation

As shown in Figure 7, the FCM segmentation method In order to segment images, Fuzzy C-Means (FCM) clustering is used. In this method, membership values are given to each pixel in the picture across several clusters that represent different areas of the image. The FCM method successfully divides the picture into relevant segments that represent various parts of the tomato leaf by optimizing the cluster centers and membership functions sequentially.

ISSN 2319-345X

www.ijmrbs.com

Vol. 9, Issue 4, 2020

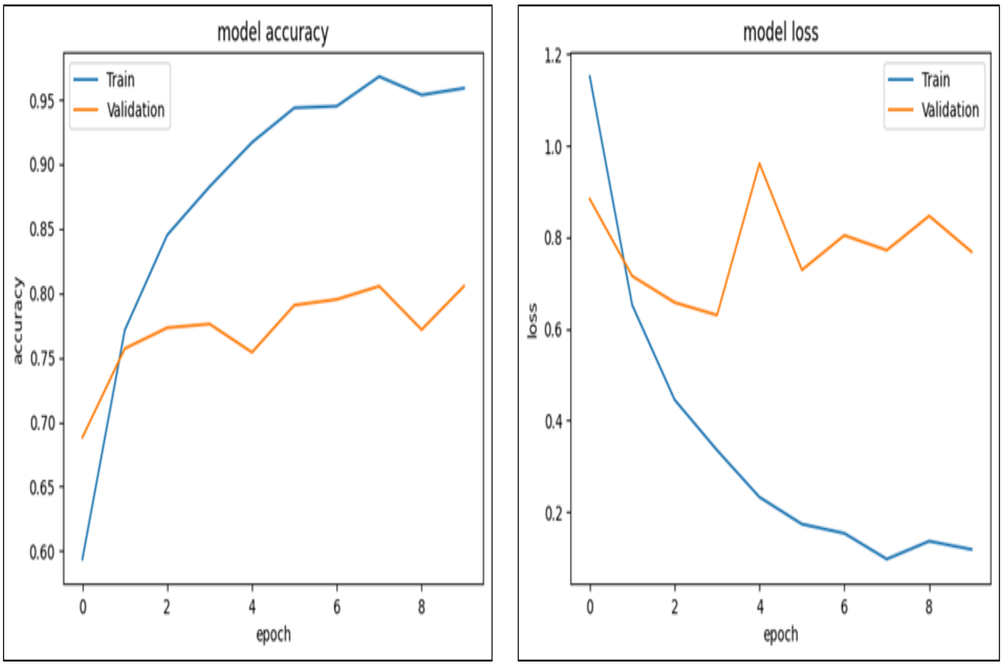
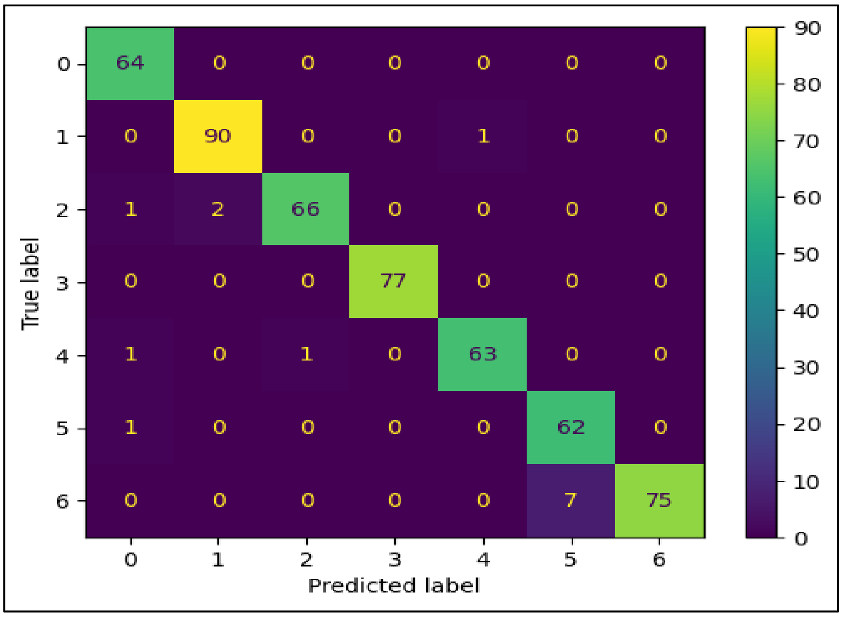


Figure 8 Training Models

Figure 8 shows the training model. The Inception V3 model is fine-tuned during training by minimizing a specified loss function, such categorical cross-entropy, with the use of backpropagation and gradient descent. In order to train a model to better identify tomato leaf diseases, it goes through the dataset several times, or epochs, adjusting its parameters each time. During training, the model's performance is evaluated using metrics including F1-score, recall, accuracy, and precision.



ISSN 2319-345X

www.ijmrbs.com

Vol. 9, Issue 4, 2020

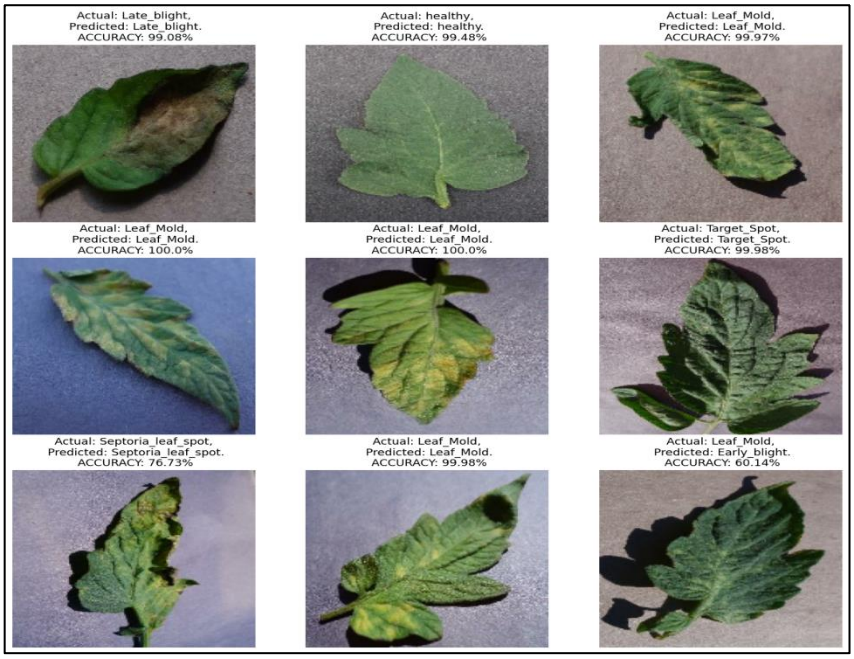
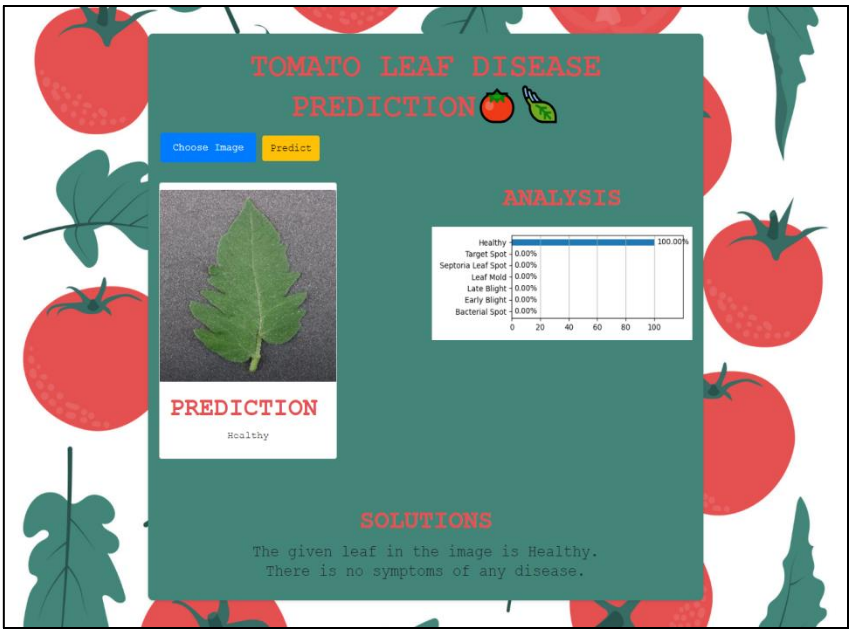


Figure 9 Predicted Image

The expected label and pictures are shown in Figure 9. Every time the web app receives a fresh photo of a tomato leaf, the trained model determines which disease class is most likely impacting the leaf. The user may learn a lot about the illness kind with this anticipated label. Additionally, to help with visual interpretation and decision-making, the model creates a forecasted picture that highlights the unhealthy parts of the leaf.

WEB CODE RESULT

International Journal of Management Research and Business Strategy
ISSN 2319-345X
www.ijmrbs.com
Vol. 9, Issue 4, 2020



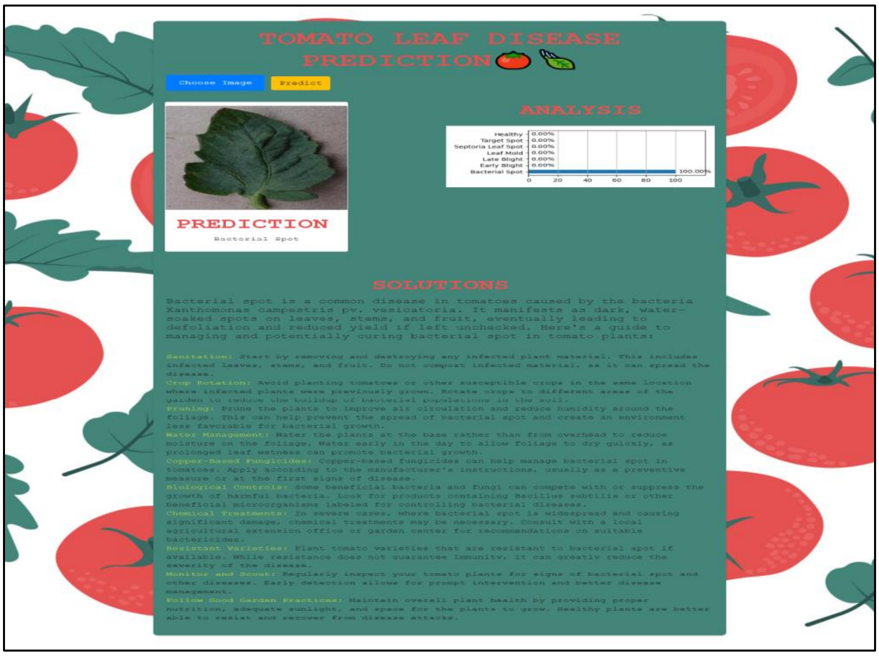


Figure10 Prediction Output

Figure 10 shows an example of a prediction output picture.After then, on a designated page inside the web app, the user sees the forecast results. Website visitors may easily see the predicted disease label and associated details for the given tomato leaf picture. In order to increase agricultural output and crop yield while decreasing the effect of diseases, users are urged to promptly assess the health status of tomato plants and take necessary measures.

CONCLUSION

In order to detect illnesses affecting tomato leaves, this research makes use of Inception V3. The Python programming language was used to execute this project with great success. There are many steps to the method, beginning with preprocessing that uses a Wiener filter to successfully reduce noise. Then, fuzzy c-means (FCM) clustering is used for segmentation. The segmentation procedure correctly recovers important characteristics

ISSN 2319-345X
www.ijmrbs.com
Vol. 9, Issue 4, 2020

from the preprocessed pictures via iterations. Afterwards, Inception V3 quickly and accurately sorts the segmented pictures, cutting down on training time while improving accuracy. When all of these steps are taken, illnesses affecting tomato leaves may be identified. Built as an intuitive online tool, the system facilitates quick and easy picture uploading and analysis, leading to the immediate presentation of anticipated illnesses. Having a site that describes the system and all of its features in great detail also improves the user experience. In conclusion, this experiment demonstrates how effective it is to combine conventional image processing methods with deep learning approaches when it comes to managing agricultural diseases. Through its user-friendly and effective solution, it has the potential to greatly aid agricultural practitioners in the prompt identification and control of diseases, leading to improved crop health and output.

REFERENCES

1. Y. Chen and J. Meng, Q. Wu, (2020) "DCGAN-Based Data Augmentation for Tomato Leaf Disease Identification," in IEEE Access, vol. 8, pp. 98716-98728.
2. S. Ahmed, T. Ahmed, M. B. Hasan, M. H. Kabir and M. R. K. Sony (2022) "Less is More: Lighter and Faster Deep Neural Architecture for Tomato Leaf Disease Classification," in IEEE Access, vol. 10, pp. 68868-68884.
3. J. Song, J. Xing and S. Zhou (2021)"Tomato Leaf Disease Identification by Restructured Deep Residual Dense Network," in IEEE Access, vol. 9, pp. 28822-28831.
4. G. Chen, J. Ding, Y. Guo, Y. He and G. Yang, (2020), "Self-Supervised Collaborative Multi-Network for Fine-Grained Visual Categorization of Tomato Diseases," in IEEE Access, vol. 8, pp. 211912-211923.
5. Y., Liu, S., Peng, Wu and Zhao (2021), Tomato leaf disease diagnosis based on improved convolution neural network by attention module. Agriculture, 11(7), 651.
6. Chen, X., G., Chen, A., Hu. H, Yi, J., Zhang, W and Zhou (2020), Identification of tomato leaf diseases based on combination of ABCK-BWTR and B-ARNet. Computers and Electronics in Agriculture, 178, 105730.
7. S. -J. Hong et al., (2023),"Application of X-Ray Imaging and Convolutional Neural Networks in the Prediction of Tomato Seed Viability," in IEEE Access, vol. 11, pp. 38061-38071.
8. C. Hu, P. Li, X. Liu, and Z. Pan (2019),"Automatic Detection of Single Ripe Tomato on Plant Combining Faster R-CNN and Intuitionistic Fuzzy Set," in IEEE Access, vol. 7, pp. 154683-154696.
9. M. Bettelli et al., (2023),"Classification and Forecasting of Water Stress in Tomato Plants Using Bioristor Data," in IEEE Access, vol. 11, pp. 34795-34807.
10. E. Özbılge, E. Ozbılge, Ö. Toygar and M. K. Ulukök (2022), "Tomato Disease Recognition Using a Compact Convolutional Neural Network," in IEEE Access, vol. 10, pp. 77213-77224.
11. R. -S. Jia, Y. -B. Liu, Z. -F. Xu, H. -M. Sun and C. -Y. Zhao (2020),"Fast Method of Detecting Tomatoes in a Complex Scene for Picking Robots," in IEEE Access, vol. 8, pp. 55289-55299.
12. H. Basanta, Y. -P. Huang and T. -H. Wang (2020), "Using Fuzzy Mask R-CNN Model to Automatically Identify Tomato Ripeness," in IEEE Access, vol. 8, pp. 207672-207682, 2020.
13. Y. Long and M. Ma(2022), "Recognition of Drought Stress State of Tomato Seedling Based on Chlorophyll Fluorescence Imaging," in IEEE Access, vol. 10, pp. 48633-48642.
14. J. Guo et al., "Revolutionizing Agriculture: Real-Time Ripe Tomato Detection with the Enhanced Tomato-YOLOv7 System," in IEEE Access.
15. C. Song et al., (2023), "TDPPL-Net: A Lightweight Real-Time Tomato Detection and Picking Point Localization Model for Harvesting Robots," in IEEE Access, vol. 11, pp. 37650-37664.

ISSN 2319-345X
www.ijmrbs.com
Vol. 9, Issue 4, 2020

16. J. Jun, J. Kim, H. I. Son and J. Seol, (2021) "Towards an Efficient Tomato Harvesting Robot: 3D Perception, Manipulation, and End-Effector," in IEEE Access, vol. 9, pp. 17631-17640.
17. A. Amir, M. Butt and O. Van Kooten (2021), "Using Machine Learning Algorithms to Forecast the Sap Flow of Cherry Tomatoes in a Greenhouse," in IEEE Access, vol. 9, pp. 154183-154193.
18. S. Abdelhay, S. Ghani , S. Gowid, Haggag M,A. Mecheter and F. Musharavati(2019), "An Intelligent Hybrid Experimental-Based Deep Learning Algorithm for Tomato-Sorting Controllers," in IEEE Access, vol. 7, pp. 106890-106898.
19. K. Roy et al. (2023), "Detection of Tomato Leaf Diseases for Agro-Based Industries Using Novel PCA DeepNet," in IEEE Access, vol. 11, pp. 14983-15001.
20. T. Hiraguri et al(2023),"Autonomous Drone-Based Pollination System Using AI Classifier to Replace Bees for Greenhouse Tomato Cultivation," in IEEE Access, vol. 11, pp. 99352-99364.