

Article

Detecting Plant Disease in Corn Leaf Using EfficientNet Architecture—An Analytical Approach

Fathimathul Rajeeena P. P.^{1,*}, Aswathy S. U.², Mohamed A. Moustafa^{3,4} and Mona A. S. Ali^{1,5,*}

¹ Computer Science Department, College of Computer Science and Information Technology, King Faisal University, Al-Ahsa 31982, Saudi Arabia

² Department of Computer Science and Engineering, Marian Engineering College, Kazhakkootam, Trivandrum 695582, India; aswathy.su@gmail.com

³ Management Department, College of Business Administration, King Faisal University, Al-Ahsa 31982, Saudi Arabia; mamoustafa@kfu.edu.sa

⁴ Faculty of Tourism and Hotel Management, Helwan University, Helwan 11795, Egypt

⁵ Computer Science Department, Faculty of Computers and Artificial Intelligence, Benha University, Benha 12311, Egypt

* Correspondence: fatimah.rajeeena@kfu.edu.sa (F.R.P.P.); m.ali@kfu.edu.sa (M.A.S.A.)

Abstract: The various corn diseases that affect agriculture go unnoticed by farmers. Each day, more crops fail due to diseases as there is no effective treatment or a way to identify the illness. Common rust, blight, and the northern leaf grey spot are the most prevalent corn diseases. The presence of a disease cannot be accurately detected by simply looking at the plant. This will lead to improper pesticide use, which harms people by bringing on chronic diseases. Therefore, maintaining food security depends on accurate and automatic disease detection. It might be possible to save time and stop crop degradation before it takes place by utilising digital technologies. Hence, applying modern digital technologies to identify the disease in the damaged corn fields automatically will be more advantageous to the farmers. Many academics have recently become interested in deep learning, which has aided in creating an exact and autonomous picture classification scheme. The use of deep learning techniques and their adjustments for detecting corn illnesses can greatly assist contemporary agriculture. To find plant leaf diseases, we employ image acquisition, preprocessing, and classification processes. Preprocessing includes procedures such as reading images, resizing images, and data augmentation. The suggested project is based on EfficientNet and improves the precision of the database of corn leaf diseases by tweaking the variables. Tests are run using DenseNet and Resnet on the test dataset to confirm the precision and robustness of this approach. The recognition accuracy of 98.85% that can be achieved using this method, according to experimental results, is significantly higher than those of other cutting-edge techniques.

Keywords: artificial intelligence; deep learning; machine learning; convolutional neural network; transfer learning; plant disease



Citation: Rajeeena P. P., F.; S. U., A.; Moustafa, M.A.; Ali, M.A.S. Detecting Plant Disease in Corn Leaf Using EfficientNet Architecture—An Analytical Approach. *Electronics* **2023**, *12*, 1938. <https://doi.org/10.3390/electronics12081938>

Academic Editor: Oscar Deniz Suarez

Received: 4 February 2023

Revised: 7 March 2023

Accepted: 24 March 2023

Published: 20 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Artificial intelligence and machine learning are being used for various agricultural operations, such as detecting diseases, soil monitoring, weed control, insect recognition, crop evaluation using drones and machine vision, and weather prediction. Agriculture is the world's largest business, contributing 18% of the GDP [1]. Agriculture is the primary source of income for a sizeable portion of the world. Therefore, crop productivity is important in the world.

Corn is a widely grown crop. Corn agriculture plays an important role because of its export potential and the fact that it supplies a lot of farmers with food. Several businesses use corn, including the food and beverage, poultry, and animal feed industries. Its productivity is more than that of wheat and rice put together, and it is a staple food in

numerous regions worldwide. However, Saudi Arabia's grain yields are less than half the world's average. The fact that corn is prone to several illnesses, and its considerably lower crop output overall, is one factor in the low corn yield. Here are some of the most prevalent corn diseases:

- **Gray Leaf Spot:** This disease is caused by the fungus *Cercospora zea-maydis* and results in small, grayish lesions on leaves that can coalesce and lead to leaf blight. Gray leaf spots can reduce yield if left unchecked.
- **Northern Corn Leaf Blight:** This disease, caused by the fungus *Exserohilum turcicum*, results in cigar-shaped leaf lesions and can cause leaf blight. Severe infections can reduce yield and quality.
- **Southern Corn Leaf Blight:** Also caused by a fungus, this disease (*Cochliobolus heterostrophus*) causes elliptical lesions on leaves that can expand and merge. It can also cause ear rot and a reduction in yield.
- **Fusarium Ear Rot:** This disease is caused by the fungus *Fusarium verticillioides* and can cause a range of symptoms, including ear rot and reduced yield. It can also produce mycotoxins, harming humans and livestock if ingested.
- **Goss's Wilt:** This bacterial disease (*Clavibacter michiganensis* subsp. *nebraskensis*) causes "freckles" on leaves and a characteristic "water-soaked" appearance. Severe infections can lead to blight and yield loss.

Existing methods for disease detection include visual inspection, DNA-based tests (such as PCR), and hyperspectral imaging. While these methods can be effective, they each have their limitations. Visual inspection can be time-consuming and subjective, DNA-based tests can be expensive and require specialised equipment, and lighting conditions can limit hyperspectral imaging.

One approach gaining traction is using machine learning algorithms to automatically analyse images of crops and detect disease symptoms. By training these algorithms on large datasets of images, researchers can develop highly accurate tools for detecting diseases in the field. This approach can potentially be faster, more objective, and more scalable than traditional disease detection methods. However, it also requires high-quality image data and careful calibration to ensure accurate results.

In addition to diminishing crop productivity and nutritional quality, some leaf infections remain an issue for agricultural production. Common leaf ailments include leaf rust, grey leaf spot, and leaf blight. Many leaf diseases may look remarkably similar in their early stages, making it challenging to identify them visually. A team of experts and ongoing crop monitoring are required to observe leaf diseases visually [2]. It is, therefore, exceedingly costly, time-consuming, and unreliable. Detecting and categorising leaf diseases can be performed automatically, quickly, and accurately using deep learning algorithms. Researchers have developed a model predicting whether corn plants will develop leaf diseases.

Plant leaf disease detection is an active area of research, and various methods have been proposed to address this problem; we take a closer look at some of this research in the following section. Inception, VGG16, Resnet50, Resnet101, Resnet152, and Densenet 121 are some of the most widely used deep learning architectures for plant leaf disease detection. Inception is a deep convolutional neural network architecture that uses inception modules to improve the model's performance. VGG16 is another deep learning architecture that uses a 16-layer convolutional neural network to classify images. Resnet50, Resnet101, and Resnet152 are residual networks that use skip connections to address the vanishing gradient problem in deep neural networks. Densenet 121 is a densely connected convolutional neural network that connects each layer to every other layer in a feed-forward fashion.

Despite the effectiveness of these methods, there is a need for new approaches that can improve the accuracy and efficiency of plant leaf disease detection. This is where the new approach using EfficientNet B0 comes in. EfficientNet B0 is a novel neural network architecture that has achieved state-of-the-art performance on various image classification tasks while being computationally efficient. It uses efficient scaling and compound coeffi-

cient optimisation techniques to achieve high accuracy while maintaining a small model size and low computational cost.

Compared to the existing methods, EfficientNet B0 offers a more efficient and accurate solution for plant leaf disease detection, which is essential in developing effective agricultural solutions. Its efficiency in processing large amounts of data and classifying them accurately can help in the early detection and prevention of plant diseases, ultimately leading to increased crop yield and economic growth in the agricultural sector.

This study proposes an architecture for deep learning to classify plant diseases based on EfficientNet [3]. Several state-of-the-art CNN structures are compared with the proposed model, including Inception V3, VGG16, Resnet50, Resnet101, and Resnet152 as shown in Figure 1.

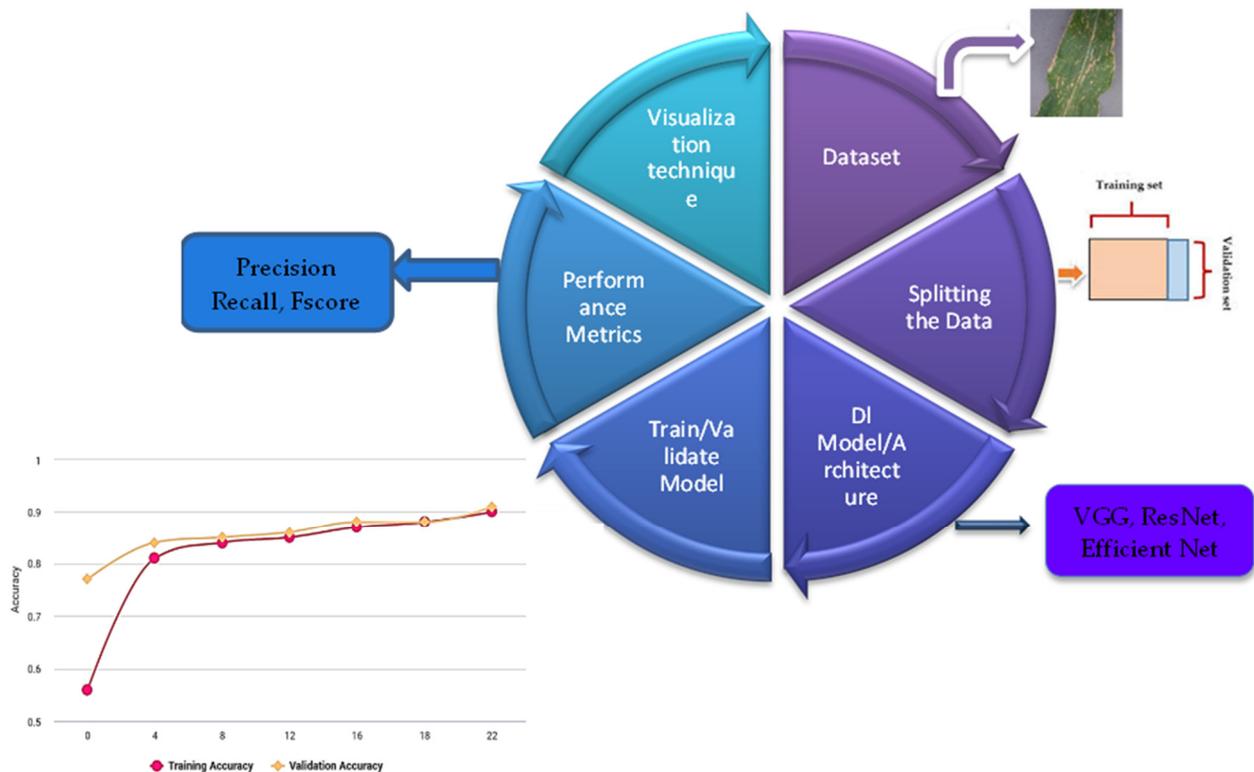


Figure 1. Overall conceptual view of prediction of leaf-based disease.

Key Highlights

The study highlights several key contributions to developing an efficient deep-learning model for identifying leaf images in corn plants. First, the study suggests using the EfficientNet structure as a suitable approach for this task. Secondly, the study employs two datasets, PlantVillage and PlantDoc, to train and validate the model. Thirdly, the study compares the performance of the proposed model against other state-of-the-art deep learning techniques, including Inception V3, VGG16, Resnet50, Resnet101, Resnet152, and Densnet121. Fourthly, transfer learning is implemented to train every single model component. Lastly, the study demonstrates that the efficiency of the EfficientNet network is higher than those of the other deep learning methods tested. Overall, the study presents an effective and efficient deep-learning model for identifying leaf images in corn plants, with potential applications in improving crop management and reducing the use of pesticides.

2. Literature Review

Zhang et al. [4] suggested a new SVM utilising a genetics approach (GA-SVM). Six different types of illnesses affecting corn leaves were acquired by the researchers and classified. The following steps were taken to classify the diseases: JPEG images were converted

to BMP format for image processing. To obtain different qualities, the images were also changed from RGB to HSI designs (with average and standard changes of R, G, and B and shape features such as area, circumference, circularity, height, and width). The creation of binary images was then accomplished by division. Lastly, the orthogonal rotation method was employed to find the right genetic optimisation algorithms. Twenty feature values were input into the model. Four kernels were used to evaluate the classification performance of SVM and GA-SVM. After contrasting the two techniques' performances, the researcher concluded that GA-SVM had a greater classification result (within 88.72% and 92.59% for GA-SVM and between 69.63% and 90.09% for SVM).

Alehegn [5] classified three distinct forms of corn leaf diseases in plants using the KNN (K-nearest neighbour) and ANN (Artificial Neural Network) approaches of identification (usual rust, leaf blight, and leaf spot). In Ethiopian farming areas, pictures of both healthy and sick corn leaves were taken. For training and testing, at least 800 images were taken into consideration. Textured, morphological, and colour characteristics were extracted from the images. The KNN and ANN classification models each received 22 variables in total. With an accuracy percentage of 94.4% versus 82.5%, the researcher concluded that ANN outperformed KNN.

Zhang et al. [6] upgraded GoogLeNet, and Cifar 10 algorithms were offered for the identification of eight corn plant pathogens (southern leaf blight, brown spot, rust, round spot, dwarf mosaic, curvularia leaf spot, grey leaf spot, northern leaf blight). Five hundred images were gathered for nine distinct types (8 classes of diseased corn leaves and one for healthy leaves). The classification process was used to incorporate the images. Compared to the VGG and Alexnet frameworks, the suggested GoogLeNet topology has 22 levels and attributes. ReLU was used, and more tiers were added to enhance the Cifar 10 designs. The dataset of corn leaves was used to assess the models' efficiency.

Kanaka Durga and Anuradha [7] utilised SVM and ANN algorithms to categorise leaf diseases in tomato and corn plants. The dataset contained 200 images of healthy and sick leaves, including those with tomato mosaic virus, common rust, bacterial spot, and northern leaf blight. They employed the subsequent procedures to determine the diseases: The RGB image was transformed into grayscale, and then the image was divided by figuring out the intensity gradient at each pixel. HOG was the method used for extracting the features (histogram of oriented gradients). The SVM and ANN classifier models received the characteristics that were extracted. SVM provided a 55–65% precision for the corn crop compared to ANN.

To discriminate between diseases that appear to be identical, Bhatt et al. [8] employed CNN architectures to construct a system for diagnosing corn leaf diseases. This framework incorporates decision-tree-based classifiers and adaptable enhancing techniques. The four visual data categories were frequent rust, regular leaf, leaf blight, and leaf spot. Each class's images were from the PlantVillage collection. The images were scaled according to the CNN model's requirements for image processing techniques. The CNN models provided the classifiers with characteristics. It was shown that inception-v2 offered the maximum degree of accuracy for randomised woodlands. The authors acknowledged that it was challenging to distinguish between the leaf blight and leaf spot classifications depending on the extracted features of each categorisation.

Chen et al. [9] proposed a lightweight network architecture called MobInc-Net for performing crop disease recognition and detection. The proposed architecture enhances the Inception module by replacing the original convolutions with depth-wise and point-wise convolutions. The modified Inception (M-Inception) module is paired with a pre-trained MobileNet to extract high-quality image features. A completely linked Softmax layer is added with the actual number of categories, followed by an SSD block, to classify and detect crop disease types. This architecture seems to be designed specifically for crop disease recognition and detection, and lightweight modules may make it suitable for deployment on resource-constrained devices.

Alatawi et al. [10] presented a convolutional neural network VGG-16 model to detect plant diseases, allowing farmers to take timely actions regarding treatment without further delay. To carry this out, 19 different classes of plant diseases were chosen, where 15,915 plant leaf images (both diseased and healthy leaves) were acquired from the Plant Village dataset for training and testing.

Mukti et al. [11] discussed a Transfer-Learning-based CNN model for the identification of plant diseases precisely. Our dataset consists of 70,295 training images and 17,572 validation images holding 38 different plant leaf images. They focused mainly on the ResNet50 network, a popular CNN architecture, as the pre-trained model in Transfer Learning.

Wu et al. [12] proposed an improved model based on ResNet101 to identify woody fruit plant leaf diseases. The leaf disease dataset of six kinds of fruits was divided into 25 categories based on species, disease type, and severity. The authors used a global average pooling layer to reduce model training parameters and employed layer normalisation, dropout, and L2 regularisation to prevent model overfitting. They also incorporated the Squeeze-and-Excitation Network (SENet) attention mechanism to enhance the model's feature extraction ability. Transfer learning was utilised to reduce training time and parameters. Experimental results revealed that the proposed model achieved an overall accuracy of 85.90% in recognising woody fruit plant leaf diseases, which is 1.20% higher than the classic ResNet network, while the number of model parameters was reduced by 98.14%. The authors concluded that their proposed model provides a better solution for identifying leaf diseases of woody fruit plants and has a higher accuracy rate. The study demonstrates the effectiveness of incorporating advanced techniques such as SENet attention and transfer learning to improve the accuracy and efficiency of plant disease detection models.

Swaminathan et al. [13] proposed using a pre-trained neural network model, specifically Densenet-121, which was imported from the Keras library for training. They explained the application of convolution as a simple filter application that leads to activation and a feature map indicating the location and strength of detected features in an input image. The authors employed 35,779 images from the Huges DP Plant-Village dataset available on Kaggle to train the model to classify 29 different diseases for seven plants: potato, tomato, corn, bell pepper, grape, apple, and cherry. The Densenet-121 model achieved an impressive accuracy rate of 97.2% in detecting plant diseases, providing an efficient solution to address issues related to low crop yield and reduced crop quality caused by plant diseases. The authors highlighted the significance of their study in agriculture, emphasising the importance of using advanced technological tools to improve the agricultural sector and ensure food security.

Table 1 summarises the state-of-the-art works in plant leaf disease identification.

Table 1. Summary of state-of-the-art works.

Author	Methodology	Advantage	Disadvantage
[4] (2016)	Used SVM with genetic algorithm and HSI design	Achieved a classification result of 88.72% to 92.59% for GA-SVM	Only six types of illnesses were classified
[5] (2019)	Used KNN and ANN for classification of three corn leaf diseases	Achieved an accuracy percentage of 94.4% for ANN	Only three types of corn leaf diseases were classified
[6] (2017)	Upgraded GoogLeNet and Cifar 10 algorithms for eight corn diseases	Used GoogLeNet topology with 22 levels and attributes	The dataset size was relatively small
[7] (2019)	Utilised SVM and ANN algorithms for tomato and corn leaves	HOG was the method used for extracting features	SVM provided a precision of 55–65% for the corn crop
[8] (2019)	Employed CNN architectures for diagnosing corn leaf diseases	Offered a high degree of accuracy with inception-v2 for image processing	Difficulty in distinguishing between leaf blight and leaf spot

Table 1. Cont.

Author	Methodology	Advantage	Disadvantage
[9] (2022)	Proposed a lightweight network architecture for crop disease recognition and detection	Designed specifically for the task of crop disease recognition and detection	Only discussed the proposed architecture and did not compare it with other methods
[10] (2022)	Presented a VGG-16 model for detecting plant diseases	Chose 19 different classes of plant diseases for training and testing	No comparison with other state-of-the-art methods
[11] (2019)	Discussed a hybrid technique for crop disease detection	Combined computer vision and machine learning techniques for detecting crop diseases	No comparison with other state-of-the-art methods
[12] (2022)	Proposed a method based on ResNet101 with tweaked variables	Model parameters were reduced by 98.14% with the proposed model, making it more efficient and requiring less storage space.	Dataset and generalisability limitations
[13] (2021)	Employed Densenet-121	Densenet-121 achieved 97.2% accuracy in detecting plant diseases	The proposed model was not evaluated on other datasets or in real-world settings, making it unclear how well it would perform in different environments.

3. Methodology

The methods employed in the current investigation are described in this section. The suggested method for identifying corn plant diseases is shown in Figure 2. An input image from the PlantVillage and PlantDoc dataset is preprocessed using greyscale conversion, Otsu thresholding, and other morphological operations such as smoothing. The preprocessed image is augmented, and the features are extracted using the GLCM feature extraction method. The extracted feature trains the EfficientNet for better classification accuracy. Moreover, EfficientNet helps us to reduce the parameter size and FLOPS by an order of magnitude, thereby providing better precision. Our proposed method is validated by comparing it with other deep learning methods such as VGG and Resnet. The flow diagram is displayed in Figure 2.

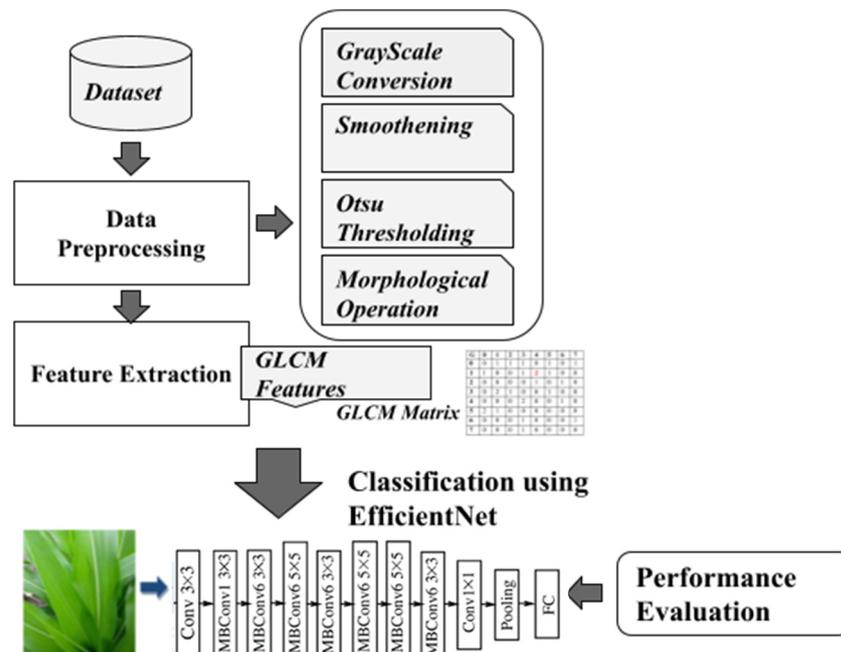


Figure 2. Flow Diagram of the proposed framework.

3.1. Dataset

In this investigation, the empirical information is obtained from the well-known PlantVillage and PlantDoc databases [14,15], which include 1306 Common Rust images, 574 Gray Leaf Spot images, 1146 Blight images, and 162 Healthy images and are presented in Figure 3.



Figure 3. Images from the dataset. (Common Rust, Gray Leaf Spot, Blight, Healthy).

3.2. Data Preprocessing and Augmentation

All leaf disease images are resized to 224×224 dimensions so that the length and width of the image are the same. These images are preprocessed by suppressing unwanted distortions and enhancing certain image features to improve the data for further processing and analysis. We create grayscale versions of the RGB images of plants and apply a Gaussian filter to calm them out. The image is then divided into blocks with Otsu's thresholding technique. We finally use a topological change to fill in the minor gaps in the image's foreground. The segmentation leaf's RGB image is bitwise, as the binary and original colour images. Shape, structure, and colour attributes are retrieved after segmenting an image. A leaf's area and perimeter are shown by its contour, which is a line connecting all areas along an object's edges of the same colour or brightness. An RGB image's average and standard deviations for every channel are also included. A contour is a line that connects all pixels along an element's edges. By dividing the total number of pixels by the hue (H) channel intensity between 30 and 70, we calculate the pixel intensity of the image using the HSV colour space. By subtracting one from the image's green colour part, non-green parts can be calculated.

3.2.1. Greyscale Conversion

For grayscale image preprocessing, the diseased leaves' image must be converted to grayscale. This matches the grayscale image luminance with the colour image luminance. This linear intensity value is encoded using the gamma expansion, which takes the three primary colours (Red, Green, and Blue) into account [16]. The gamma expansion is:

$$C_{\text{linear}} = \begin{cases} \frac{C_{\text{rgb}}}{12.92} & C_{\text{rgb}} \leq 0.04045 \\ \frac{(C_{\text{rgb}} + 0.065)}{1.065} & C_{\text{rgb}} > 0.04045 \end{cases} \quad (1)$$

An output luminance value is calculated based on a weighted sum of the three linear intensity values, with C_{rgb} representing RGB primaries and C_{linear} representing linear intensity values, ranging from 0 to 1. The conversion is performed using the function:

$$y = f(x) \quad (2)$$

The $f(x)$ function uses the weighted sum of the R, G, and B elements of RGB values to convert them to grayscale values.

$$f(x) = 0.2989 \times R + 0.5870 \times G + 0.1140 \times B \quad (3)$$

3.2.2. Smoothing

There is a 2-D convolution operator called Gaussian smoothing that uses a kernel representing the shape of a Gaussian to smooth images by removing details (higher frequencies) and noise. As shown below, the Gaussian filter has a general representation [17]:

$$G(x, y) = 1/2\pi\sigma^2 \left(e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \right) \tag{4}$$

3.2.3. Otsu Thresholding

Segmenting images into functional areas or objects is an important technique in computer vision. In computer vision, image segmentation is a critical and challenging issue. Two characteristics, discontinuity and similarity, are used to classify various image segmentation techniques. An image is divided into comparable groups of connected pixels using a region-based segmentation technique. Thresholding is a crucial technique in image segmentation as it can be applied to many kinds of image segmentation [18]. The background and object’s gray-level distributions calculate the grey-level threshold value. If $G(x, y)$ is the threshold type of $f(x, y)$, then T is the overall threshold value:

$$G(x, y) = \begin{cases} 1 & \text{if } f(x, y) \geq T \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The definition of a threshold operation is

$$T = M[x, y, p(x, y), f(x, y)] \tag{6}$$

The equation has three variables: T , which stands for the threshold; $f(x, y)$, which denotes the point’s grey value; $p(x, y)$, which denotes a local feature of the point (x, y) .

3.2.4. Morphological Operation

After segmenting the affected region, top hat and bottom hat transformations are carried out to isolate the precise location of the disease using the element disks with size 12. Foreground pixels are removed from the edges of the foreground pixels by erosion followed by dilation [19].

$$I \circ P = (I - P) + P \tag{7}$$

where I = Original Image and P = Structuring element

Closed pixels enlarge a pixel’s boundary in the image and fill in the blanks in the background region; this is called morphological closing. In the image, closing is just the opposite of opening. It consists of a dilation followed by erosion.

The mathematical function of closing is

$$I \bullet P = (I + P) - P \tag{8}$$

where I = Original Image and P = Structuring element. Table 2 shows the morphological features for leaf disease detection.

Table 2. Overall morphological features analysed.

Morphological Features	Formula
Rectangularity	$\frac{\text{area}}{\text{major axis} \times \text{minor axis}}$
Elongation	$1 - \frac{\text{minor axis}}{\text{major axis}}$
Roundness	$\frac{4\pi \times \text{area}}{\text{convex perimeter}^2}$
Solidity	$\frac{\text{area}}{\text{convex area}}$
Convexity	$\frac{\text{convex area}}{\text{convex perimeter}}$
Compactness	$\frac{4\pi \times \text{area}}{\text{perimeter}^2}$
Eccentricity	$\frac{\sqrt{\text{major axis}^2 - \text{minor axis}^2}}{\text{major axis}}$

3.3. Feature Extraction

An image can be analysed using the GLCM feature, which utilises the distance between two pixels in the image to calculate its value. A fine texture region changes rapidly, whereas a coarse texture region changes more slowly [20]. Its value is computed as follows:

$$G(m, n) = \frac{\#\{[(m_1, n_1), (m_2, n_2)] \in J | (m_1, n_1) = g_1 \& (m_2, n_2) = g_2\}}{\#S} \tag{9}$$

We build a GLCM matrix as an illustration, with the levels of intensity 1 and 4 being marked in red and based on distance $d = 1$ and angular orientation 0. This shows that distance $d = 1$ and intensity values 1 and 4 are at an angle of 0 degrees. The GLCM matrix, which has a value of 2 for row 1 and column 4, is created using the same method. Table 3 displays the GLCM matrix produced using a similar technique. Using this technique, we build the GLCM matrix for the full image and divide it into non-overlapping blocks of size $n \times n$.

Table 3. GLCM matrix for feature extraction.

G	0	1	2	3	4	5	6	7
0	0	1	1	1	0	1	0	1
1	1	0	0	1	2	1	0	0
2	0	0	0	0	1	0	1	0
3	0	2	1	0	0	1	0	0
4	0	0	0	2	0	0	1	0
5	2	1	0	0	0	0	0	0
6	0	0	0	1	0	0	0	1
7	0	0	0	1	0	0	0	0

Each block is examined using statistics showing the regular distribution, including the angular second scene, entropy, correlations, energies, contrasting, and homogeneous.

The angular second scene is calculated using Equation (10):

$$A_{sm} = \sum_{m=1}^M \sum_{n=1}^N G(m, n)^2 \tag{10}$$

Correlation: An image’s correlation value measures the similarity of its textures across horizontally and vertically perpendicular directions [20] and can be given as

$$C_{or} = \frac{\sum_{m=1}^M \sum_{n=1}^N (m - \bar{x})(n - \bar{y})G(m, n)}{\sigma_x \sigma_y} \tag{11}$$

Contrast: The contrast value is calculated according to the following formula (Equation (12)) and indicates the changes in depth and smoothness of the image:

$$C_{on} = \sum_{m=1}^M \sum_{n=1}^N (m - n)^2 G(m, n) \tag{12}$$

Entropy: Information content is measured by entropy, which is computed as follows:

$$E_{nt} = - \sum_{m=1}^M \sum_{n=1}^N G(m, n) I_g G(m, n) \tag{13}$$

Energy: A square of an element equals its energy and is given by

$$E_{ne} = \sum_{m=1}^m \sum_{n=1}^n G^2(m, n) \tag{14}$$

Homogeneity: An image’s homogeneity is determined by its structural similarity [20]. It is computed as:

$$H_{om}(m, n) = \frac{e^{-k(G(x,y)-G(m,n))^2}}{\sum_{m=1}^M \sum_{n=1}^N e^{-k(G(x,y)-G(m,n))^2}} \tag{15}$$

There are p non-overlapping blocks and q non-overlapping blocks in each block, so we can create a feature matrix by arranging the extracted features.

3.4. Classification

Since 2012, success has grown in tandem with the complexity of the models used in the ImageNet dataset, yet many are computationally inefficient. When used to solve the ImageNet classification issue, the EfficientNet model, which has 66M characteristics, obtains an 84.4% accuracy [21]. An RGB picture’s average and standard deviations for every channel are also included. A contour is a line that connects all pixels along an element’s edges.

Through smaller models, deep learning systems aim to find more effective methods. EfficientNet properly adjusts depth, width, and resolution when the network is shrunk, outperforming other cutting-edge algorithms in effectiveness [22]. Its first stage when applying the compounded scaling method is to choose a grid to investigate how the different scaling parameters of the baseline channels interact with one another while observing specified resource shortages. It thus becomes possible to determine the right transformation function for the sharpness, thickness, and width characteristics. The targeted market networks are then scaled using the parameters to match the initial structure.

The inversion bottleneck MBConv, initially introduced in MobileNetV2, is the main part of EfficientNet [23]. However, because of its greater FLOPS (Floating Point Calculations Per Second) budget, it is used slightly more frequently than MobileNetV2. The layers that make-up pieces in MBConv enlarge connections before compression. This results in direct connections between bottlenecks, which connect fewer routes than expansion stages. The calculated duration for this design’s in-depth separated convolutions is significantly decreased compared to traditional levels (approximately by a factor of k 2), where k is the kernel size, establishing the parameters for the 2D convolution window. Figure 4 depicts the EfficientNet B0 model schematic.

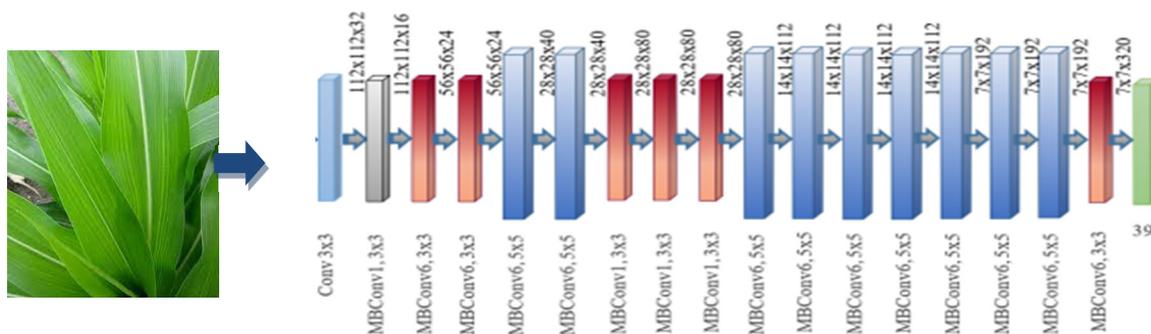


Figure 4. Schematic representation of EfficientNet.

Compound scaling uses the compounding coefficients and the recommendations in Equation (16) to appropriately scale depths, width, and resolution.

$$\begin{aligned} \text{depth } d &= \alpha^\varphi \\ \text{width } w &= \beta^\varphi \\ \text{resolution } r &= \gamma^\varphi \\ \alpha &\geq 1, \beta \geq 1, \gamma \geq 1 \end{aligned} \tag{16}$$

where α , β , and γ are the grid search used to determine the constants. The amount of resources that can be used for model scaling is controlled by the φ user-defined coefficients,

and the networks' width, depth, and resolution are allocated to α , β , and γ , respectively. FLOPS are proportional to d , w^2 , and r^2 in a typical convolution process. As the cost of computing in convolution networks is largely due to convolution operations, scaling the convolution network as given in Equation (16) increases the FLOPS of the network by approximately $(\alpha, \beta^2, \gamma^2) \varphi$ in total.

Based on Baseline EfficientNet-B0, the model is scaled using the compound scaling method in two steps.

1. The optimum values for α , β , and γ are discovered using a grid search with $\varphi = 1$ under the assumption that there are twice as many benefits available.
2. Using the calculated values, α , β , and γ will be considered constants, and the baseline networks are scaled up to produce EfficientNet-B1 through B7 by altering the parameters in Equation (16).

4. Experimental Result

The framework was created in Python and ran on a system with an Nvidia GTX 1070 GPU. We employed the provided method to transfer learning to enhance the recognition and tracking of corn plant leaf disease on the datasets rather than building the model from scratch. We used MS-COCO pre-trained weights for this. A variety of evaluation metrics were applied. Table 4 displays the overall evaluation of the suggested model.

Table 4. Overall evaluation of the proposed model.

Epoch	Accuracy	Loss	Val_Accuracy	Val_Loss
0	0.565102	1.033861	0.776978	0.574386
1	0.778308	0.550707	0.822542	0.387065
2	0.803235	0.474993	0.834532	0.35298
3	0.815434	0.451608	0.846523	0.340599
4	0.817555	0.42162	0.846523	0.35767
5	0.826836	0.39928	0.82494	0.375728
6	0.830019	0.393824	0.858513	0.309951
7	0.828958	0.385043	0.858513	0.3032
8	0.840361	0.363005	0.853717	0.304749
9	0.849642	0.347106	0.863309	0.289048
10	0.854415	0.338511	0.858513	0.276987
11	0.852559	0.333878	0.848921	0.323801
12	0.858128	0.331062	0.863309	0.291771
13	0.863697	0.320002	0.877698	0.262441
14	0.865288	0.315084	0.877698	0.257888
15	0.869	0.314482	0.889688	0.261933
16	0.87563	0.307013	0.88729	0.254111
17	0.882259	0.281178	0.894484	0.239494
18	0.869265	0.311961	0.896882	0.244813
19	0.880933	0.281312	0.88729	0.250468
20	0.884646	0.294803	0.904077	0.258856
21	0.899231	0.256305	0.911271	0.245372
22	0.894193	0.254476	0.870504	0.29428
23	0.892601	0.259995	0.892086	0.250815
24	0.903474	0.238936	0.88729	0.258726

The evaluation of the suggested model is shown in Figure 5 for various epoch values, and the graphs for the training and validation losses exhibit varied intervals. After period level 2, the training loss does rise, but it later progressively falls to 0.22. Figure 6 shows that as epoch durations are increased, the training and validation of the performance of the proposed model also rise exponentially.

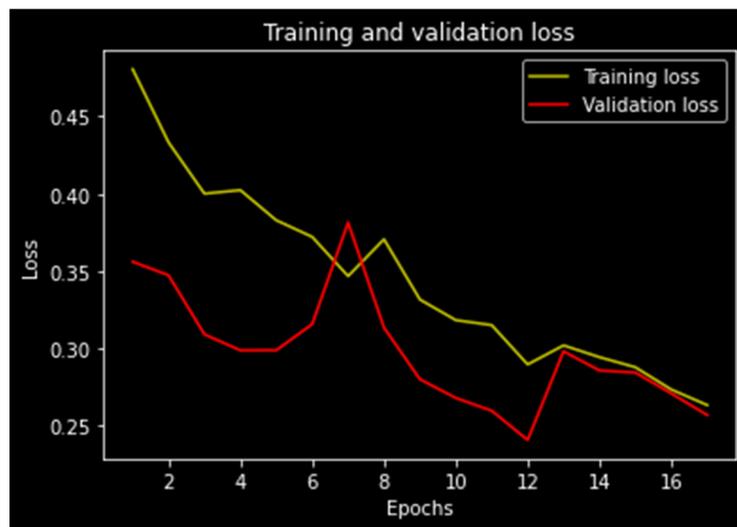


Figure 5. Overall training and validation loss of the proposed model.

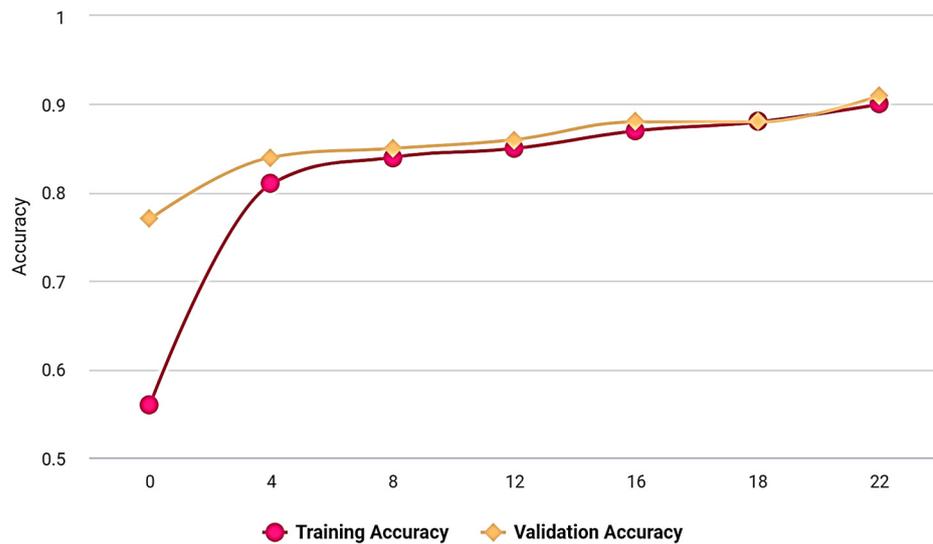


Figure 6. Training and Validation accuracy vs. Epoch. A comparison is made between the proposed model that uses EfficientNet and other classification methods commonly used in the discussed literature. Tables 5–7 represent the same against different evaluation metrics. Figure 7a–c illustrate a graphical representation of various designs across precision, sensitivity, and specificity. The recommended model is more effective due to improved dense and transition block effectiveness.

Table 5. Comparison analysis of the proposed model under Accuracy, Sensitivity, and Specificity.

Models	Accuracy	Sensitivity	Specificity
Inception	89	92	95
VGG16	81	86	92
Resnet50	85	89	91
Resnet101	88	92	95
Resnet152	92	95	96
Densnet 121	93	97	96
EfficientNet B0	98.85	98	98

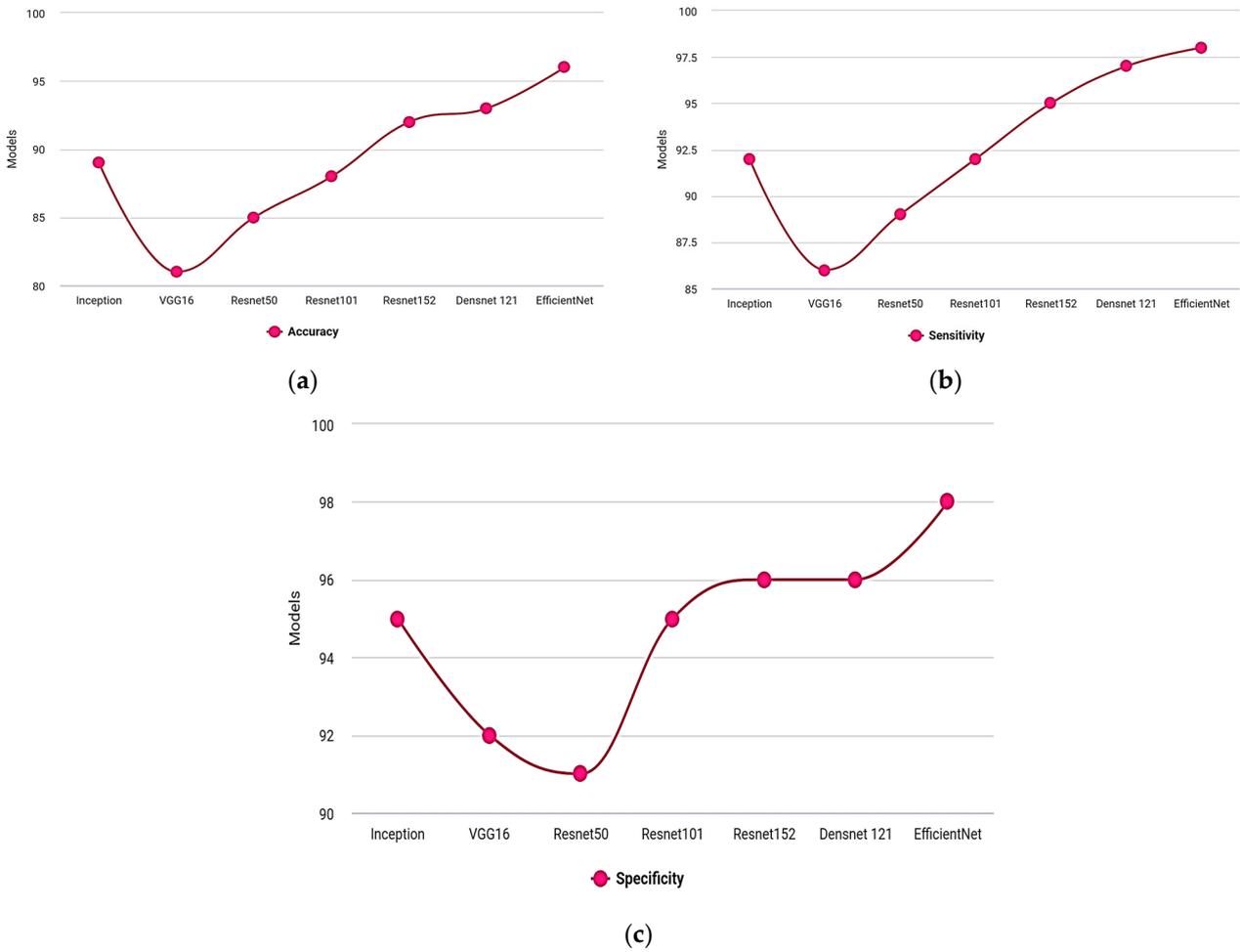


Figure 7. Models vs. (a) Accuracy, (b) Sensitivity, (c) Specificity.

Table 6 shows the evaluation under precision, recall, and F1-score. Figure 8a–c show the graphical representation in which the proposed model outperforms due to the image’s greater preprocessing efficiency. The feature extraction feature eventually creates an additional dimension for the classifier to analyse the features.

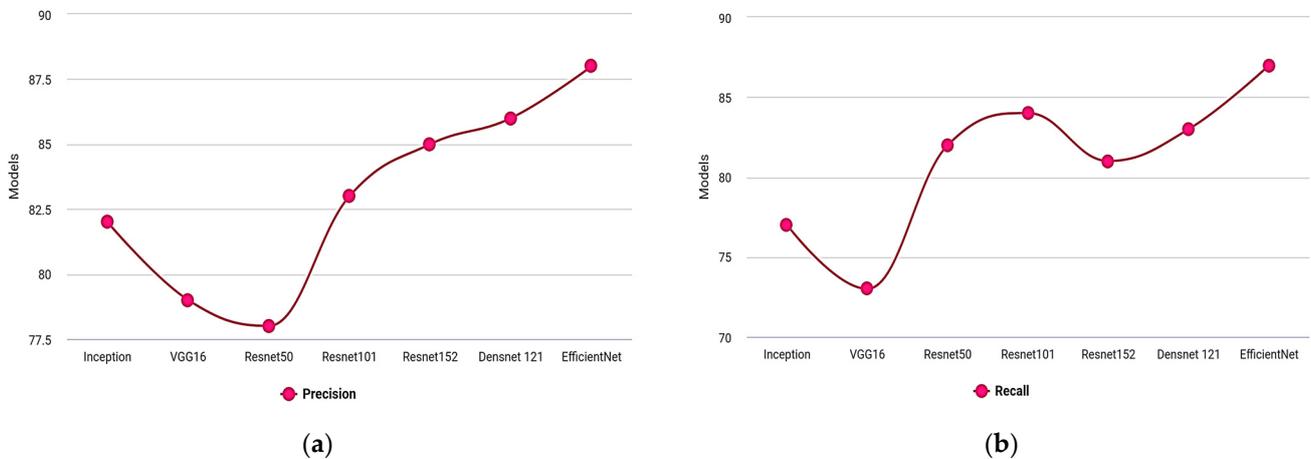


Figure 8. Cont.

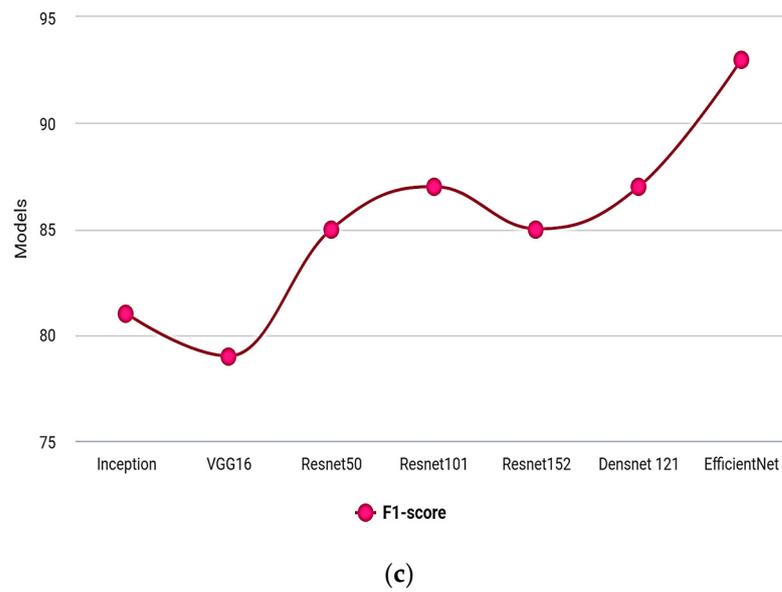


Figure 8. Models vs. (a) Precision, (b) Recall, (c) F1-score.

Table 6. Comparison analysis of the proposed model under Precision, Recall, and F1-score.

Models	Precision	Recall	F1-Score
Inception	82	77	81
VGG16	79	73	79
Resnet50	78	82	85
Resnet101	83	84	87
Resnet152	85	81	85
Densnet 121	86	83	87
EfficientNet B0	88	87	93

Table 7 shows the evaluation under IoU, mAP, and computation time. Figure 9a–c show the graphical representation in which the proposed model outperforms due to greater training and testing efficiency and the models’ capability to learn based on the larger datasets.

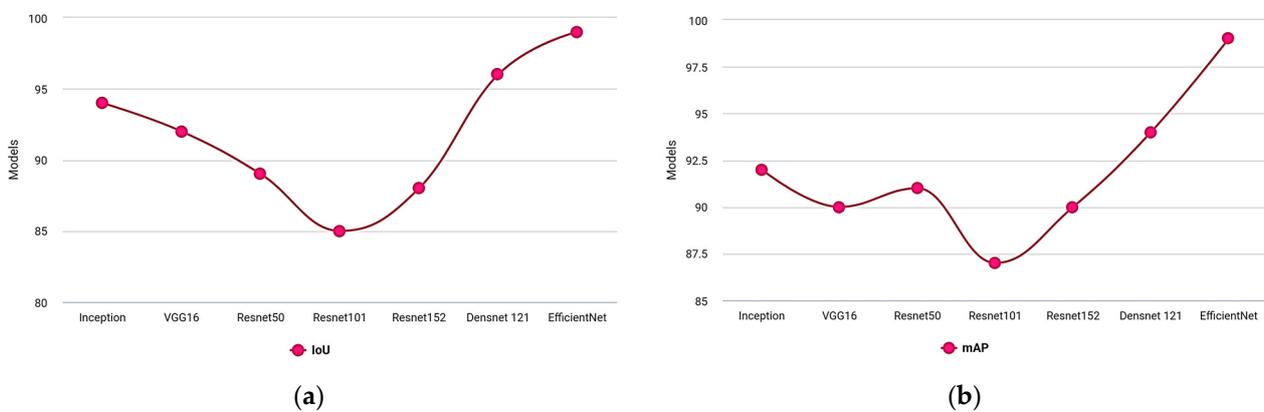


Figure 9. Cont.

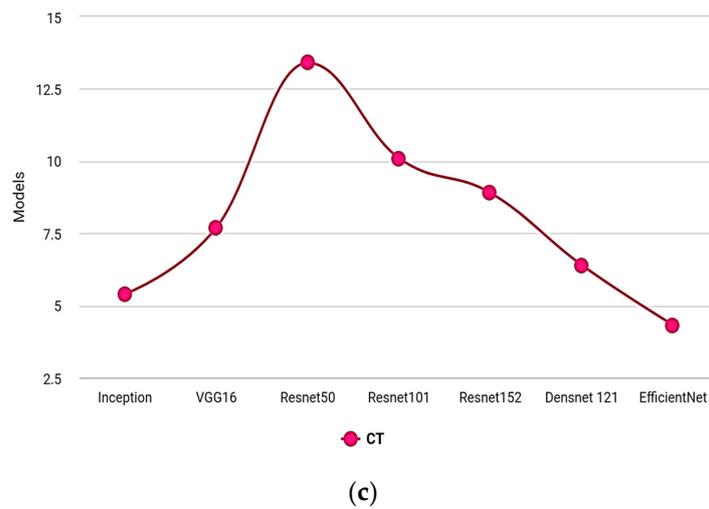


Figure 9. Models vs. (a) IoU, (b) mAP, (c) CT.

Table 7. Comparison analysis of the proposed model under IoU, mAP, and CT.

Models	IoU	mAP	Computation Time
Inception	94	92	5.4
VGG16	92	90	7.7
Resnet50	89	91	13.4
Resnet101	85	87	10.1
Resnet152	88	90	8.9
Densnet 121	96	94	6.4
EfficientNet B0	99	99	4.32

We compare the performance of our approach in terms of classification and network parameters with many base models in Table 7. The values unmistakably demonstrate that the EfficientNet B0 is more resilient than its peer approaches. Furthermore, Table 6 shows the ResNet-152 [24] approach, which has the longest processing time, while the VGG-16 [25] approach has the most parameters. Our method, in contrast, is computationally the most reliable because it processes a suspected sample in just 1067 s. The confusion matrix is depicted in Figure 10.

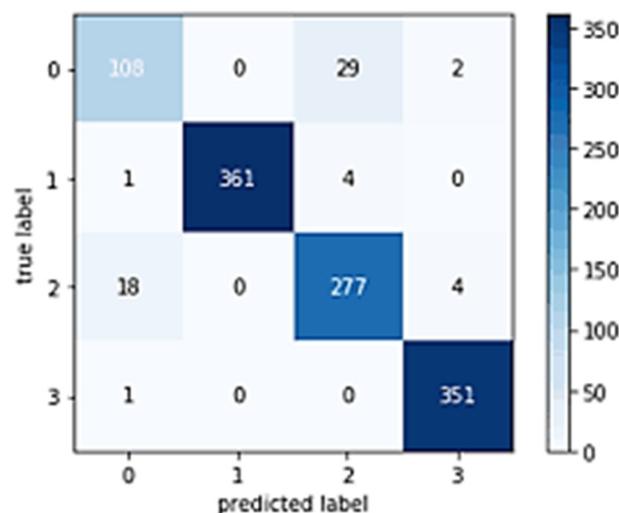


Figure 10. Confusion Matrix.

Compared to comparative base networks, those with higher computational costs are more likely to have sample post-processing problems such as noise, blurring, and light variations. This method offers a more accurate framework for computing key points under varying image distortions and accurately displays complex sample transformations. The results discussed above support the conclusion that our proposed model performs better both at classification performance and execution speed than conventional Deep Learning approaches. As the compared approaches are built based on very deep architectures, which can easily lead to overfitting, the proposed solution outperforms them. This means that, in terms of the classification of plant diseases, our methodology is more efficient and effective than the methods.

5. Conclusions

In recent years, the use of deep learning algorithms for pattern categorisation and image analysis has gained significant attention. Deep learning can help identify plant leaf disease by enabling the development of accurate and efficient automated systems that can classify and diagnose plant diseases. This can help reduce the reliance on manual inspection, which can be time-consuming and prone to errors.

One potential implication of using deep learning for plant disease identification is that it can facilitate early detection and intervention, allowing farmers to take appropriate action to control the spread of diseases. This can help reduce crop losses and minimise the need for excessive pesticide use, thereby improving the sustainability of agriculture and reducing the potential risks associated with pesticide exposure. In this study, we proposed an EfficientNet-based deep learning framework for identifying illnesses in corn leaves. Our approach outperformed cutting-edge deep learning techniques for plant leaf damage identification, achieving an accuracy of 98.85% and a precision of 88% while also being computationally efficient. However, it is important to acknowledge the limited diversity of plant species in our dataset. We aim to expand our dataset to include a broader range of plant leaf diseases to enhance the generalisation capability of our model. In the future, the proposed approach could be further developed for mobile settings, allowing plant pathologists and farmers to quickly and accurately recognise plant pathogens and take necessary precautions to mitigate the impact of plant diseases.

6. Limitations and Future Directions

The limitations of this approach include the limited dataset used in this study, which only focused on corn leaves, and the need to expand to other plant species to validate its effectiveness in identifying other plant diseases. The dataset used was relatively small, which may affect the model's generalisation capabilities and limit its use in the real world when presented with a diverse range of environmental and plant conditions. Additionally, the proposed method may not detect new or unknown diseases not included in the dataset.

In terms of future directions, the model can be improved by increasing the size of the dataset and introducing more plant species to the model to make it more versatile. Another area of improvement is the development of a mobile application that farmers and plant pathologists can use in the field to recognise and identify diseases in real time, making it more accessible and user-friendly. Finally, the model's performance can be further improved by incorporating additional features, such as environmental data, in the training process. This would enable the model to consider different factors that may affect the plant's health and provide a more comprehensive analysis of the plant's health status.

Author Contributions: Conceptualisation, F.R.P.P. and M.A.S.A.; data curation, F.R.P.P. and M.A.M.; formal analysis, F.R.P.P., A.S.U. and M.A.S.A.; funding acquisition, F.R.P.P. and M.A.S.A.; methodology, F.R.P.P., A.S.U. and M.A.M.; project administration, M.A.S.A.; resources, A.S.U.; supervision, F.R.P.P.; validation, A.S.U. and M.A.M.; writing—original draft, F.R.P.P., A.S.U., M.A.M. and M.A.S.A.; writing—review and editing, F.R.P.P., A.S.U., M.A.M. and M.A.S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been funded by The Deanship of Scientific Research, King Faisal University, Saudi Arabia, with grant Number (GRANT3155).

Data Availability Statement: The study did not report any data.

Acknowledgments: We deeply acknowledge The Deanship of Scientific Research, King Faisal University, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Waheed, A.; Goyal, M.; Gupta, D.; Khanna, A.; Hassaniien, A.E.; Pandey, H.M. An optimised dense convolutional neural network model for disease recognition and classification in corn leaf. *Comput. Electron. Agric.* **2020**, *175*, 105456. [\[CrossRef\]](#)
2. Singh, V.; Misra, A.K. Detection of plant leaf diseases using image segmentation and soft computing techniques. *Inf. Process. Agric.* **2017**, *4*, 41–49. [\[CrossRef\]](#)
3. Tan, M.; Le, Q. Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
4. Zhang, Z.Y.; He, X.Y.; Sun, X.H.; Guo, L.M.; Wang, J.H.; Wang, F.S. Image recognition of maize leaf disease based on GA-SVM. *Chem. Eng. Trans.* **2015**, *46*, 199–204.
5. Alehegn, E. Maize Leaf Diseases Recognition and Classification Based on Imaging and Machine Learning Techniques. Ph.D. Thesis, BDU Institutional Repository, Bahir Dar, Ethiopia, 2020.
6. Zhang, X.; Qiao, Y.; Meng, F.; Fan, C.; Zhang, M. Identification of maize leaf diseases using improved deep convolutional neural networks. *IEEE Access* **2018**, *6*, 30370–30377. [\[CrossRef\]](#)
7. Durga, N.K.; Anuradha, G. Plant disease identification using SVM and ANN algorithms. *Int. J. Recent Technol. Eng.* **2019**, *7*, 5S4.
8. Bhatt, P.; Sarangi, S.; Shivhare, A.; Singh, D.; Pappula, S. Identification of Diseases in Corn Leaves using Convolutional Neural Networks and Boosting. In Proceedings of the ICPRAM, Prague, Czech Republic, 19–21 February 2019; pp. 894–899.
9. Chen, J.; Chen, W.; Zeb, A.; Yang, S.; Zhang, D. Lightweight inception networks for the recognition and detection of rice plant diseases. *IEEE Sens. J.* **2022**, *22*, 14628–14638. [\[CrossRef\]](#)
10. Alatawi, A.A.; Alomani, S.M.; Alhawiti, N.I.; Ayaz, M. Plant Disease Detection using AI based VGG-16 Model. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 718–727. [\[CrossRef\]](#)
11. Mukti; Zahan, I.; Biswas, D. Transfer learning based plant diseases detection using ResNet50. In Proceedings of the 2019 4th International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 20–22 December 2019; IEEE: Piscataway, NJ, USA, 2019.
12. Wu, Z.; Jiang, F.; Cao, R. Research on recognition method of leaf diseases of woody fruit plants based on transfer learning. *Sci. Rep.* **2022**, *12*, 15385. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Swaminathan, A.; Varun, C.; Kalaivani, S. Multiple Plant Leaf Disease Classification using Densenet-121 Architecture. *Int. J. Electr. Eng. Technol.* **2021**, *12*, 38–57.
14. Singh, D.; Jain, N.; Jain, P.; Kayal, P.; Kumawat, S.; Batra, N. PlantDoc: A dataset for visual plant disease detection. In Proceedings of the 7th ACM IKDD CoDS and 25th COMAD, New York, NY, USA, 5–7 January 2020; pp. 249–253.
15. Pandian, J.A.; Geetharamani, G. Data for: Identification of Plant Leaf Diseases Using a 9-layer Deep Convolutional Neural Network. *Comput. Electr. Eng.* **2019**, *76*, 323–338.
16. Saravanan, C. Color image to grayscale image conversion. In Proceedings of the 2010 Second International Conference on Computer Engineering and Applications, Bali, Indonesia, 19–21 March 2010; IEEE: Piscataway, NJ, USA, 2010; Volume 2, pp. 196–199.
17. Li, P.; Wang, H.; Yu, M.; Li, Y. Overview of image smoothing algorithms. *J. Phys. Conf. Ser.* **2021**, *1883*, 012024. [\[CrossRef\]](#)
18. Soeleman, M.A.; Nurhindarto, A.; Muslih, M.; Karis, W.; Muljono, M.; Al Zami, F.; Pramunendar, R.A. Adaptive threshold for moving objects detection using gaussian mixture model. *Telecommun. Comput. Electron. Control.* **2020**, *18*, 1122–1129. [\[CrossRef\]](#)
19. Said, K.A.M.; Jambek, A.B.; Sulaiman, N. A study of image processing using morphological opening and closing processes. *Int. J. Control. Theory Appl.* **2016**, *9*, 15–21.
20. Panchal, P.; Raman, V.C.; Mantri, S. Plant diseases detection and classification using machine learning models. In Proceedings of the 2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), Bengaluru, India, 20–21 December 2019; IEEE: Piscataway, NJ, USA, 2019; Volume 4, pp. 1–6.
21. Chai, J.; Zeng, H.; Li, A.; Ngai, E.W. Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Mach. Learn. Appl.* **2021**, *6*, 100134. [\[CrossRef\]](#)
22. Atila, Ü.; Uçar, M.; Akyol, K.; Uçar, E. Plant leaf disease classification using EfficientNet deep learning model. *Ecological Informatics.* **2021**, *61*, 101182. [\[CrossRef\]](#)
23. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.

24. Rajasree, R.; Latha, C.; Paul, S. Application of Transfer Learning with a Fine-tuned ResNet-152 for Evaluation of Disease Severity in Tomato Plants. In *Mobile Computing and Sustainable Informatics*; Springer: Singapore, 2022; pp. 695–710.
25. Too, E.C.; Yujian, L.; Njuki, S.; Yingchun, L. A comparative study of fine-tuning deep learning models for plant disease identification. *Comput. Electron. Agric.* **2019**, *161*, 272–279. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.