ISSN No:-2456-2165

Identification and Classification of Tomato Leaf Diseases Utilizing a Convolutional Neural Network-Based System

Wijesekara JPD; Waruna Henarangoda; Pavithra Subhashini
School of Computing, NSBM Green University Town, Homagama, Sri Lanka

Abstract:- The integration of smart farming systems and requisite infrastructural developments represents a paradigm shift in agricultural technology, significantly augmenting both the quality and yield within the sector. Tomatoes, as one of the world's most vital crops, are frequently afflicted by leaf diseases, which critically impact harvest outcomes. Prompt detection and identification of these diseases are imperative to mitigate crop devastation and implement efficacious control measures, particularly in understanding the pathogen species composition. Delays in disease diagnosis and inadequate control responses can precipitate substantial crop losses and marked degradation in product quality. This study introduces an IT-based solution leveraging image processing and deep learning methodologies for the expedited detection of diseases in tomato plants. Utilizing a dataset of 22,930 images, encompassing nine distinct diseased-leaf categories and a healthy-leaf category, the research employs a Convolutional Neural Network (CNN) for disease classification and prediction. The model demonstrates notable efficacy, achieving an overall accuracy rate of 98.2% and maintaining a loss rate of 0.0532. This advancement in precision agriculture exemplifies the potential of integrating cutting-edge technology with traditional farming practices to enhance productivity and disease management.

***Keywords**:- Component, formatting, style, styling, insert.*

I. INTRODUCTION

Tomato (Solanum lycopersicum), recognized globally as a key vegetable crop, is valued both for its unique nutritional properties and its extensive cultivation. The Food and Agriculture Organization (FAO) reported that in 2020, global tomato production reached 186.821 million metric tons over an area of 5,051,983 hectares, yielding an average of 37.1 metric tons per hectare [1][2][3]. In Sri Lanka, between 2000 and 2010, tomato volume and production escalated by 25% and 71% respectively, with 7,261 hectares yielding 75,335 metric tons in 2010 [4].

A critical aspect of agriculture is the timely recognition and management of plant diseases. The FAO estimates that diseases and pests annually account for up to 40% of global crop losses [5]. The economic impact of plant diseases and invasive insects is substantial, costing over $220 billion and $70 billion respectively [6]. Plant diseases fall into two main categories: those caused by infectious microorganisms (nematodes, bacteria, viruses, fungi) which are contagious and can spread rapidly under favorable conditions [7], and non-contagious diseases stemming from physical or chemical factors like environmental stresses, nutritional deficiencies, or herbicide damage [8].

Tomato production is notably affected by pests and diseases, which can lead to reduced yields or even total crop failure [9]. Processed tomatoes are particularly vulnerable to arthropods, plant diseases, and nematodes, with estimated losses at 34.4% of potential yield under current cultivation practices. Without crop protection, these losses could escalate to 77.7% [7]. Traditional methods of disease and pest detection rely heavily on the farmer's experience or specialist consultation, a process that is slow, costly, subjective, and often impractical [11]. Moreover, the prevalent use of agricultural chemicals and pesticides, while necessary, poses environmental risks and can result in excessive pesticide residues [10].

The advancement of information technology has introduced innovative methods for detecting crop diseases and insect pests, addressing many of the limitations of traditional approaches. These technological solutions offer faster, more accurate, and cost-effective alternatives for disease and pest management in agriculture [12].

II. RELATED WORKS

At the Gokongwei College of Engineering, De La Salle University in Manila, Philippines, researchers developed a motor-controlled image capturing box designed to capture images of all four sides of tomato plants for leaf disease detection and recognition. Utilizing a dataset of 4,923 images of diseased and healthy tomato plant leaves obtained under controlled conditions, the automated system demonstrated a 91.67% accuracy in recognizing tomato plant leaf diseases [13].

Chitkara University Institute of Engineering and Technology at Chitkara University in Rajpura developed a Convolutional Neural Network (CNN)-based model to assist farmers in early-stage disease identification. The research, conducted using Google Colab, leveraged a dataset of 3,000 images representing nine different tomato leaf diseases and a healthy leaf category, achieving a prediction accuracy of 98.49% [14].

The Facility Horticulture Laboratory at Universities in Shandong, Weifang University of Science and Technology, China, focused on an improved object detection algorithm using YOLOv3. This research aimed to facilitate early, real-time detection of tomato diseases and pests amidst the complex backgrounds typically found in natural environments. The proposed algorithm yielded a 94.77% F1 score, a 91.81% Average Precision (AP) value, a low false detection rate of 2.1%, and a rapid detection time of 55 milliseconds [15].

The School of Computing at Telkom University in Bandung, Indonesia, devised a system to classify three different diseases affecting tomato plant leaves. Employing an Augmentation technique on a dataset of 4,400 leaf images, the data was trained using a Convolutional Neural Network (CNN). This approach resulted in an average accuracy of 97.8%, with a peak accuracy of 99.5%, as determined by 5-fold cross-validation [16].

Finally, the Department of Telecommunication Engineering at the University of Engineering and Technology in Taxila, Pakistan, proposed an advanced classification model for detecting and classifying tomato leaf diseases. Using a training dataset of 450 images, various models, including AlexNet, VGG16, and VGG19, were employed to extract visual attributes before applying the K-Nearest Neighbor (KNN) classification. AlexNet achieved the highest accuracy at 76.1%, surpassing the accuracy rates of VGG16 (55.0%) and VGG19 (55.6%) [17].

III. MATERIALS AND METHODS

A. Image Acquisition

For this research, a comprehensive dataset comprising 22,930 images of tomato leaves was compiled. This dataset was sourced from the "Tomato New Plant Diseases Dataset" available on the Kaggle platform, accessible via the link: https://www.kaggle.com/datasets/noulam/tomato. The dataset encompasses images representing nine distinct classes of diseased tomato leaves, in addition to a class of healthy leaves. Each image within the dataset is standardized with a resolution of 256x256 pixels, ensuring uniformity for effective analysis.

The dataset was meticulously organized into three primary categories: training, validation, and testing. This organization was executed following a proportionate distribution ratio of 8:1:1. Specifically, the training directory contained 18,345 images, which formed the bulk of the dataset for model training. The validation directory, essential for tuning and optimizing the model, included 2,290 images. Lastly, the testing directory, used for evaluating the model's performance, comprised 2,295 images. This structured approach to dataset segmentation is crucial in machine learning for developing robust and accurate models.

B. Pre-Processing

In this study, the preprocessing of tomato leaf images involved three key techniques: Resizing/Rescaling, Normalization, and Data Augmentation [18]. These processes are critical for preparing the images for effective analysis by the machine learning model.

- **Resizing/Rescaling**: For images that deviate from the required dimensions of 256x256 pixels, a resizing layer is employed to adjust them to the necessary size. In addition, a rescaling layer is used to rescale the images, typically by a factor of 1/256. This step ensures uniformity in image size across the dataset, which is crucial for consistent processing and analysis by the model.
- **Normalization:** Normalizing the pixel values of the images is a vital step aimed at enhancing the model's performance. This process involves scaling the pixel values to a range between 0 and 1, typically achieved by dividing each pixel value by 256. Normalization is important as it helps in reducing the computational complexity and improving the convergence speed during training. It can be integrated as a layer in the Sequential model, ensuring its application during both training and inference phases [19].
- **Data Augmentation:** This technique involves modifying the original images to generate a more diverse set of training data. Augmentation helps the model to generalize better by exposing it to a wider range of variations in the data. In this study, data augmentation was performed using specific methods, including:
- **Flip:** The RandomFlip layer is utilized to flip each image horizontally and vertically. This augmentation adds variability to the dataset by showing the model different orientations of the same leaf [21].
- **Rotation:** The RandomRotation layer is applied to randomly rotate the images during training. Each image undergoes a random rotation, introducing more variability and aiding the model in learning to recognize diseases regardless of the leaf's orientation [22]. These preprocessing techniques are integral to the preparation of the dataset, ensuring that the machine learning model receives consistently formatted and diversified data, which is essential for accurate disease detection and classification.

ISSN No:-2456-2165

No.	Class	Number of datasets	Sample Image
1.	Tomato Mosaic Virus	2238	
2.	Tomato Yellow Leaf Curl Virus	2451	
3.	Bacterial Spot	2127	
4.	Target Spot	2284	
5.	Early Blight	2400	
6.	Septoria Leaf Spot	2181	
7.	Leaf Mold	2352	
8.	Late Blight	2314	
9.	Spider Mites/ Two-spotted Spider Mite	2176	
10.	Healthy Leaves	2407	

Fig. 1: Image Acquisition

C. CNN Implementation

In the development of the tomato leaf disease detection model, the image input is a crucial first step. The model accepts images in the format of 256x256 pixels with 3 color channels (RGB), and these images are processed in batches of 32. This batch size is selected based on the architecture and capabilities of the proposed Convolutional Neural Network (CNN) model.

The construction of the CNN model includes several layers, each contributing to the feature extraction and learning process. One of the key components is the 2D Convolutional Layer, described as follows:

- **2D Convolutional Layer:** This layer serves as the foundation of the CNN model. It involves a convolutional operation where a convolution kernel (a small matrix of weights) is applied to the input image. The primary function of this layer is to extract features from the input image by sliding the kernel over the image spatially (both horizontally and vertically). During this process, the kernel is convolved with the portion of the image it covers, effectively filtering the image to extract specific features like edges, textures, or other relevant patterns [23].

The output of this convolution operation is a tensor of feature maps. These feature maps represent the presence of specific features at different locations in the image. By stacking multiple convolutional layers, each with their own kernels, the model can learn to identify increasingly complex features, which is essential for accurate disease detection in tomato leaves.

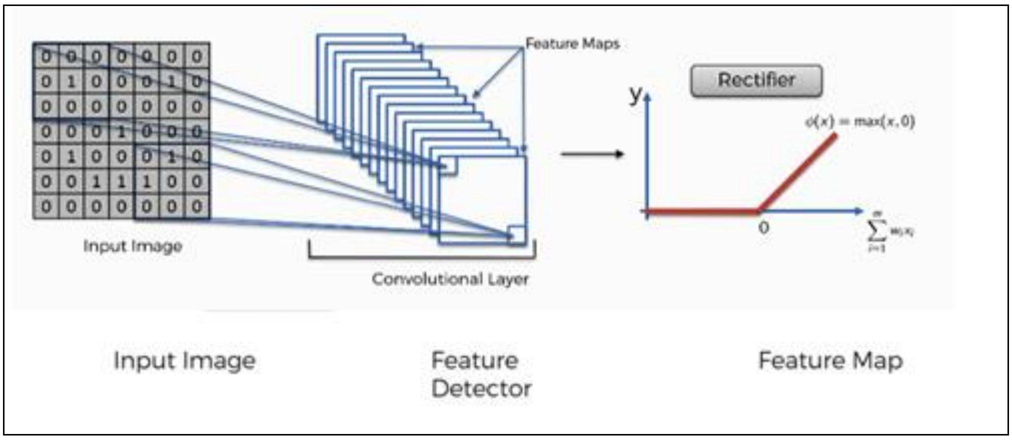

Fig. 2: 2D Convolutional Layer

ISSN No:-2456-2165

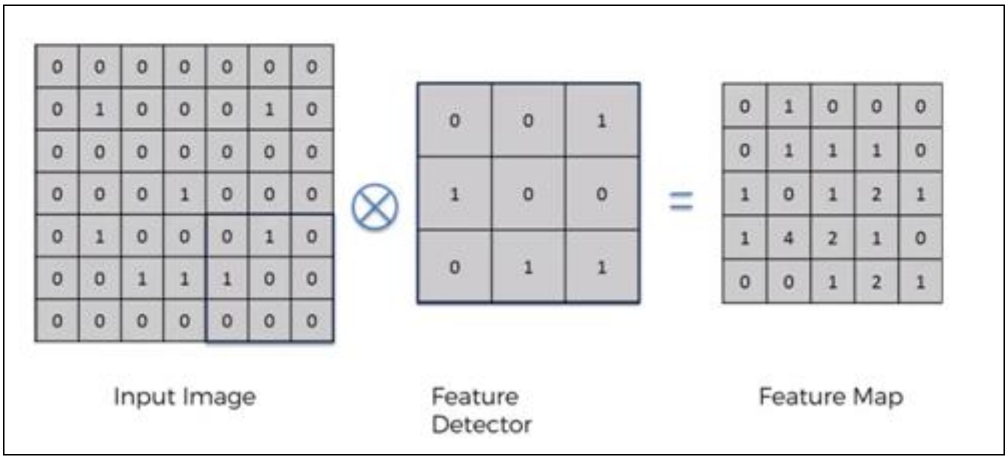

Fig. 3: Rectified Linear Unit (ReLU) Function

In summary, the 2D Convolutional Layer is a critical component in the CNN model for tomato leaf disease detection, enabling the model to learn from the visual data by extracting and processing key features from the input images.

The Rectified Linear Unit (ReLU) function plays a pivotal role in the architecture of the Convolutional Neural Network (CNN) used for tomato leaf disease detection. This function is applied following the 2D convolutional layers and serves a crucial purpose in the neural network's operation.

- **Rectified Linear Unit (ReLU) Function**: The primary function of ReLU is to introduce nonlinearity into the network's operation. It operates by replacing every negative value in the output of the convolutional layer (the filtered image) with zero. Mathematically, the ReLU function is defined as $f(x) = max(0, x)$ where x is the input to the neuron. This means if x is positive, $f(x)$ will be x; if x is negative, $f(x)$ will be zero.

The rationale behind using the ReLU function is rooted in the nature of image data. Images inherently possess non-linear properties due to variations in lighting, shadows, and object edges. The ReLU function helps in preserving these essential non-linear characteristics by eliminating negative values, which simplifies the model without losing key information contained in the image data.

Additionally, ReLU is known for its computational efficiency. Since it involves simple thresholding at zero, it is less computationally intensive compared to other non-linear functions like sigmoid or tanh. This attribute makes ReLU particularly advantageous in deep learning models where computational efficiency is vital.

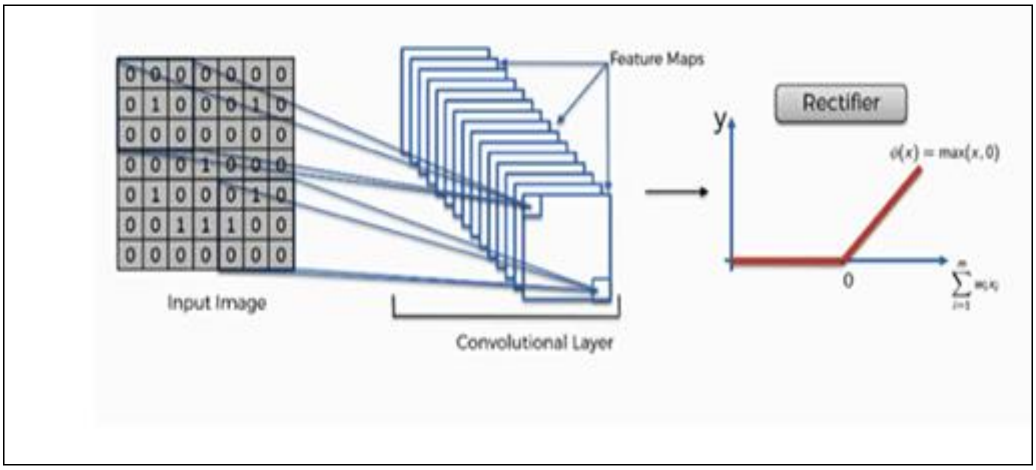

Fig. 4: Rectified Linear Unit (ReLU) Function

In summary, the ReLU function in the CNN model enhances the network's ability to process and learn from the complex, non-linear patterns present in tomato leaf images, contributing to the effectiveness of disease detection and classification [24].

The MaxPooling2D layer is a crucial component in the architecture of the Convolutional Neural Network (CNN) used for tomato leaf disease detection, particularly following one or more convolutional layers.

- **MaxPooling2D Layer**: This layer performs a 2-Dimensional max pooling operation, which is a form of down-sampling or sub-sampling. The primary function of the MaxPooling2D layer is to reduce the spatial dimensions (height and width) of the input feature maps.

The way max pooling works is by sliding a window (usually of size 2x2 or 3x3) over the input feature map and at each position, selecting the maximum value within that window. This operation is performed separately for each feature map generated by the previous convolutional layers. The result is a pooled feature map that has reduced dimensions.

The benefits of using Max Pooling2D in a CNN are multi-fold:

- **Dimensionality Reduction**: By reducing the size of the feature maps, max pooling decreases the number of parameters and computations in the network, which helps in controlling overfitting and reduces computational load.
- **Retaining Dominant Features**: Max pooling preserves the strongest features (i.e., the maximum values), ensuring that these prominent traits are carried forward in the network for further processing.
- **Invariance to Minor Changes**: The process of max pooling provides the model with a degree of invariance to small shifts and distortions in the input image, enhancing the robustness of the model.
- **Efficient Processing**: Since max pooling reduces the dimensionality of the feature maps, subsequent layers in the network have fewer inputs to process, making the network more efficient.

In the context of detecting diseases in tomato leaves, the MaxPooling2D layer aids in abstracting the essential features from the leaf images, enabling the CNN to focus on the most relevant patterns for disease classification, while reducing the computational complexity of the model [25].

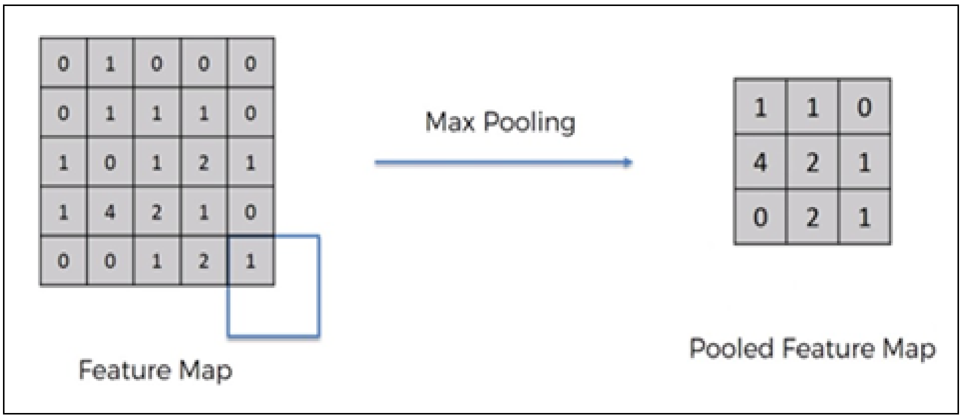

Fig. 5: MaxPooling2D Layer

The Flatten layer is an essential component in the architecture of the Convolutional Neural Network (CNN), particularly when transitioning from convolutional layers to fully connected layers in the context of tomato leaf disease detection.

- **Flatten Layer**: The primary role of the Flatten layer is to convert the pooled feature maps obtained from the preceding MaxPooling2D layers into a single, long vector. This transformation is necessary because fully connected layers (which often follow the Flatten layer in a CNN architecture) require input in the form of a one-dimensional vector.

During the flattening process, the 2D (or potentially 3D if considering depth) pooled feature maps are restructured. Each value from the feature maps is laid out in a single line, effectively transforming the multi-dimensional format into a 1D vector. This vector then serves as the input for the subsequent fully connected layers in the network.

The Flatten layer does not involve any learning or transformation of values; it merely reorganizes the data. Despite its simplicity, this layer is crucial for the following reasons:

- **Transition to Fully Connected Layers**: It bridges the convolutional part of the CNN, which is adept at feature extraction and spatial analysis, with the fully connected part, which is responsible for classification or regression tasks.
- **Maintaining Feature Information**: While reorganizing the data, the Flatten layer preserves all the feature information extracted by the convolutional and pooling layers, ensuring that this information is available for the final decision-making layers of the network.
- **Simplifying Network Design**: By converting the data into a 1D format, the Flatten layer simplifies the design and implementation of the subsequent layers in the network.

In the context of tomato leaf disease detection, the Flatten layer ensures that the spatially extracted features from the leaf images are properly channeled into the classification part of the network, maintaining the integrity of the information necessary for accurate disease identification [26].

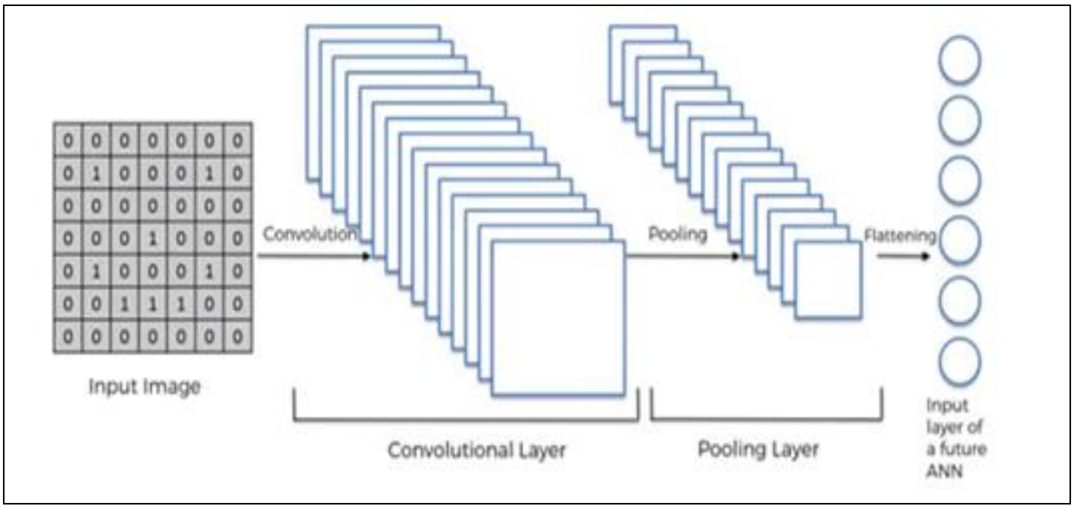

Fig. 6: Flatten Layer

The Dense layer, also known as a Fully Connected (FC) layer, is a fundamental component in the architecture of a Convolutional Neural Network (CNN), particularly in the latter stages of the network, following feature extraction and flattening processes.

- **Dense Layer (Fully Connected Layer)**: This layer plays a critical role in the interpretation and classification phase of a CNN. In a Dense layer, every neuron is connected to every neuron in the previous layer. This interconnectedness allows the network to combine the features extracted in earlier layers (by convolutional and pooling layers) to make predictions or classifications.

The primary functions of the Dense layer in a CNN, especially in the context of tomato leaf disease detection, are as follows:

- **Feature Integration**: The Dense layer integrates the features extracted by convolutional and pooling layers. It combines these features in various ways to identify patterns that are more abstract and complex than what the initial layers could detect.
- **Detection of Specific Features**: Each neuron in a Dense layer can be considered as a detector of a specific, complex feature in the input image. These features might be particular combinations of edges, textures, colors, or shapes that are indicative of certain diseases in tomato leaves.
- **Classification Decision**: In the final Dense layer, often followed by a softmax activation function, the network makes its final decision, classifying the input image into one of the classes (e.g., different types of diseases or healthy leaves).
- **Learning High-Level Abstractions**: The Dense layer's capacity to learn high-level abstractions makes it suitable for tasks like disease detection in plants, where the distinguishing features might be subtle or highly specific.

In summary, the Dense layer in a CNN for tomato leaf disease detection serves as the decision-making component of the network. It synthesizes the lower-level features extracted by previous layers into higher-level abstractions, enabling the network to perform complex classification tasks such as distinguishing between different types of diseases or identifying healthy leaves [27].

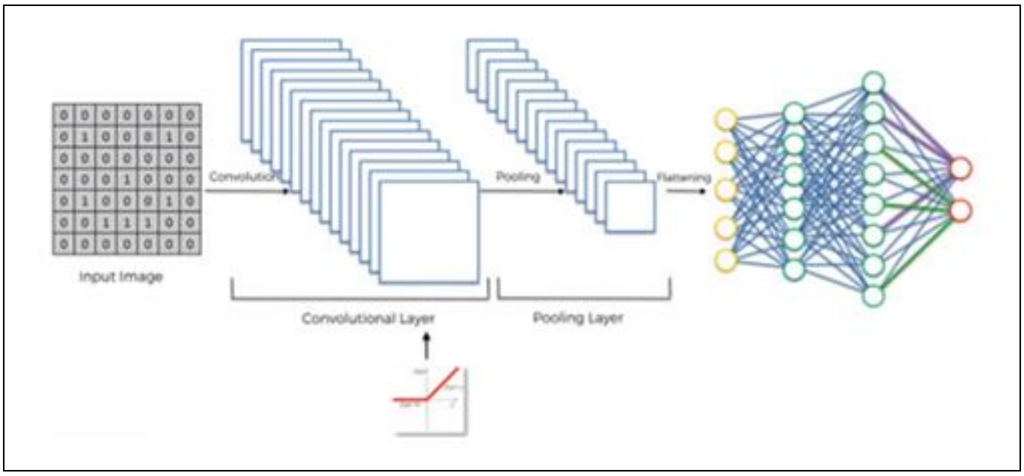

Fig. 7: Dense Layer (Fully Connected Layer)

The Softmax function, often used in conjunction with the Cross Entropy function, plays a crucial role in the output layer of a Convolutional Neural Network (CNN), particularly in multi-class classification tasks like tomato leaf disease detection.

- **Softmax Function**: This function is applied in the final layer of a CNN to convert the raw output values (logits) from the previous Dense layer into probabilities. The Softmax function is defined as follows:

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

Where x_i is the output of the last Dense layer for class *i*, and the denominator is the sum of exponential values of outputs for all classes in the problem.

The key characteristics and functions of the Softmax layer in a CNN are:

- **Probability Distribution**: The Softmax function outputs a probability distribution, where each value represents the probability of the input image belonging to a particular class. The probabilities sum up to 1, ensuring a valid probabilistic interpretation.
- **Multi-Class Classification**: It is particularly useful in scenarios where each input is to be classified into one of several classes, such as differentiating between various diseases in tomato leaves.
- **Decision Making**: The class with the highest probability is typically chosen as the model's prediction, making the Softmax function integral to the decision-making process in classification tasks.
- **Cross Entropy Function as Loss Function**: Following the application of the Softmax function, the Cross Entropy function is commonly used as the loss function to measure the performance of the CNN. It calculates the difference between the predicted probability distribution (from the Softmax function) and the actual distribution (the true labels). The formula for Cross Entropy loss in a multi-class classification is:

$$\text{Cross Entropy Loss} = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

Where *M* is the number of classes, *y* is a binary indicator (0 or 1) if class label *c* is the correct classification for observation *o*, and *p* is the predicted probability observation *o* is of class *c*.

The objective in training the CNN is to minimize this Cross Entropy loss, which effectively means aligning the model's predictions as closely as possible with the true labels, thereby optimizing the model's accuracy and functionality [28].

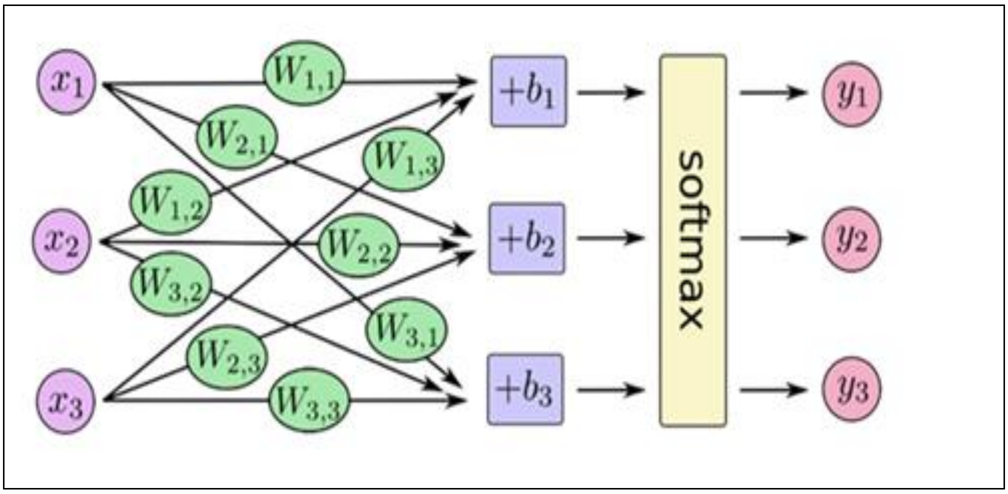

Fig. 8: Softmax Function

IV. ANALYSIS OF THE TEST RESULTS

A. Training and Validation Accuracy

The performance of the Convolutional Neural Network (CNN) model for tomato leaf disease detection was rigorously evaluated using a specific training and validation protocol. This protocol involved setting the number of epochs and batch size, both of which are critical parameters in the training process.

- **Epochs**: An epoch in machine learning is one complete pass of the training dataset through the algorithm. In this study, 100 epochs were used for the final training of the CNN model. This decision was based on experimental data showing that 100 epochs yielded higher accuracy. Initially, 50 epochs were used in preliminary training experiments, but this resulted in less accurate outcomes with significant loss values for the tomato image samples. Increasing the number of epochs to 100 allowed the model to learn more effectively from the data, improving its ability to accurately classify the images.
- **Batch Size**: The batch size refers to the number of training examples utilized in one iteration. In this study, a batch size of 32 was chosen. This size is widely recognized as a standard and effective choice in many machine learning applications. The batch size is typically less than the total number of datasets, and 32 is often considered an optimal balance between computational efficiency and the ability to generalize from the training data.

ISSN No:-2456-2165

The performance of the CNN model was quantified in terms of accuracy and loss:

- **Training Accuracy and Loss**: The highest training accuracy achieved was 98.86% with a corresponding loss value of 0.0367. These metrics indicate the model's effectiveness in learning from the training dataset.
- **Validation Accuracy and Loss**: The highest validation accuracy recorded was 99.03% with a loss value of 0.0355. Validation accuracy and loss are crucial indicators of the model's ability to generalize and perform well on unseen data.
- **Overall Model Performance**: The overall accuracy of the model was reported to be 98.2%, with a loss of 0.0532. These figures demonstrate the model's robustness and reliability in detecting tomato leaf diseases.

In summary, the chosen epoch count and batch size were pivotal in optimizing the CNN model's performance, resulting in high accuracy and low loss, thereby validating the model's effectiveness in classifying diseases in tomato leaves.

B. Model Testing

The evaluation of the Convolutional Neural Network (CNN) model's performance on the test dataset is a critical step in validating its effectiveness in detecting diseases in tomato leaves. This process involves comparing the model's predictions with the actual labels to assess accuracy and reliability.

- **Batch Selection for Testing**: Initially, a batch of images from the test dataset is selected. Testing the model in batches aligns with how the model was trained (using a batch size of 32), ensuring consistency in the evaluation process.
- **Obtaining Probabilities with Softmax Function**: Each image in the test batch is passed through the trained CNN model, which outputs a probability distribution for each image using the Softmax function. These probabilities indicate the likelihood of the image belonging to each of the possible classes (different types of diseases or healthy).
- **Determining Predicted Class with argmax**: To identify the model’s predicted class for each image, the **numpy.argmax** function is applied to the probability distribution. The **argmax** function identifies the index of the highest probability in the array, which corresponds to the model's prediction for the most likely disease class.
- **Comparing Predicted and Actual Labels**: The model’s predictions are then compared with the actual labels of the images. This comparison is crucial for evaluating the model's performance.

For example, in testing the model’s ability to identify Tomato Septoria Leaf Spot, the predicted label provided by the model for each image in the test batch is compared against the actual label of the disease. The effectiveness of the model is assessed based on how accurately it can identify this specific disease, amongst others.

The performance is often visualized or presented in a figure, showing side-by-side comparisons of predicted labels versus actual labels for a set of test images. This visual representation allows for a clear and intuitive assessment of the model's accuracy and reliability in real-world scenarios.

In summary, testing the CNN model on the test dataset and evaluating its performance based on actual predictions is essential to confirm the model's capability in accurately detecting and classifying tomato leaf diseases [29].

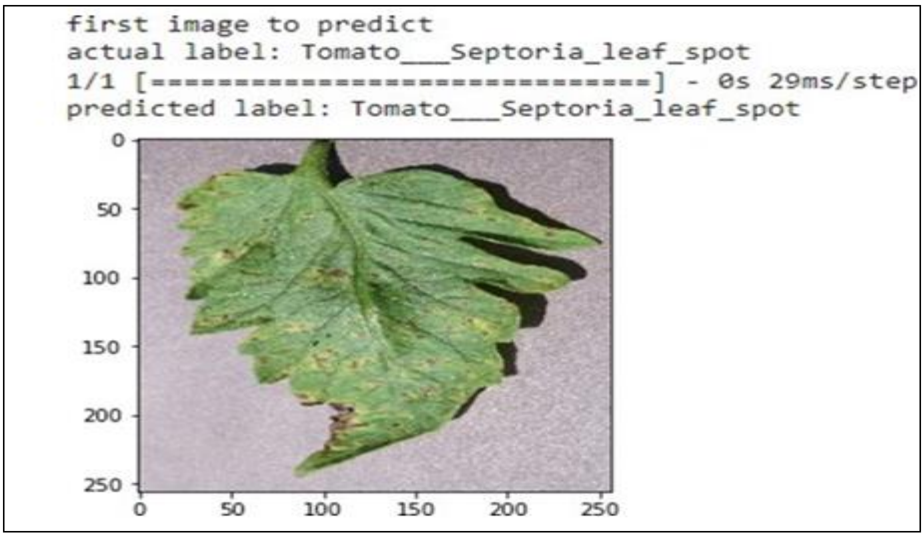

Fig. 9: Model Testing in One Class

Evaluating the confidence of predictions made by a Convolutional Neural Network (CNN) model is a crucial aspect of assessing its performance, particularly in tasks such as disease detection in tomato leaves. This process involves not only identifying the predicted class for each image but also quantifying the model's certainty in its predictions.

ISSN No:-2456-2165

- **Function Design**: A specialized function is created, which takes an image and the trained model as inputs. This function processes the image through the model to obtain predictions.
- **Outputting Predicted Class and Confidence Score**: For each image, the function outputs the predicted class (e.g., a specific type of tomato leaf disease or healthy leaf) along with a confidence score. This score is typically derived from the output probabilities generated by the model's Softmax layer.
- **Confidence Score as a Percentage**: The confidence score is usually presented as a percentage. It represents the probability assigned by the model to the predicted class, indicating how confident the model is about its prediction. A higher confidence score suggests greater certainty by the model in its classification.
- **Visual Representation in Figures**: The output of this function, including the predicted class and confidence score for each image, can be visualized in a figure. Such a figure typically shows the image, the model's predicted class, and the corresponding confidence score. Additionally, it may also include the actual class for comparison.

For example, in a figure illustrating the model's performance, each image might be accompanied by labels such as "Predicted: Tomato Septoria Leaf Spot (Confidence: 92%)" and "Actual: Tomato Septoria Leaf Spot". This format allows for a clear and concise presentation of the model's predictions and their accuracy, alongside the level of confidence the model has in each prediction.

In summary, evaluating the confidence scores of predictions made by the CNN model provides an additional layer of insight into the model's performance. It not only shows how often the model is correct but also how certain it is about its predictions, which is a vital aspect of model evaluation in practical applications like tomato leaf disease detection [30].

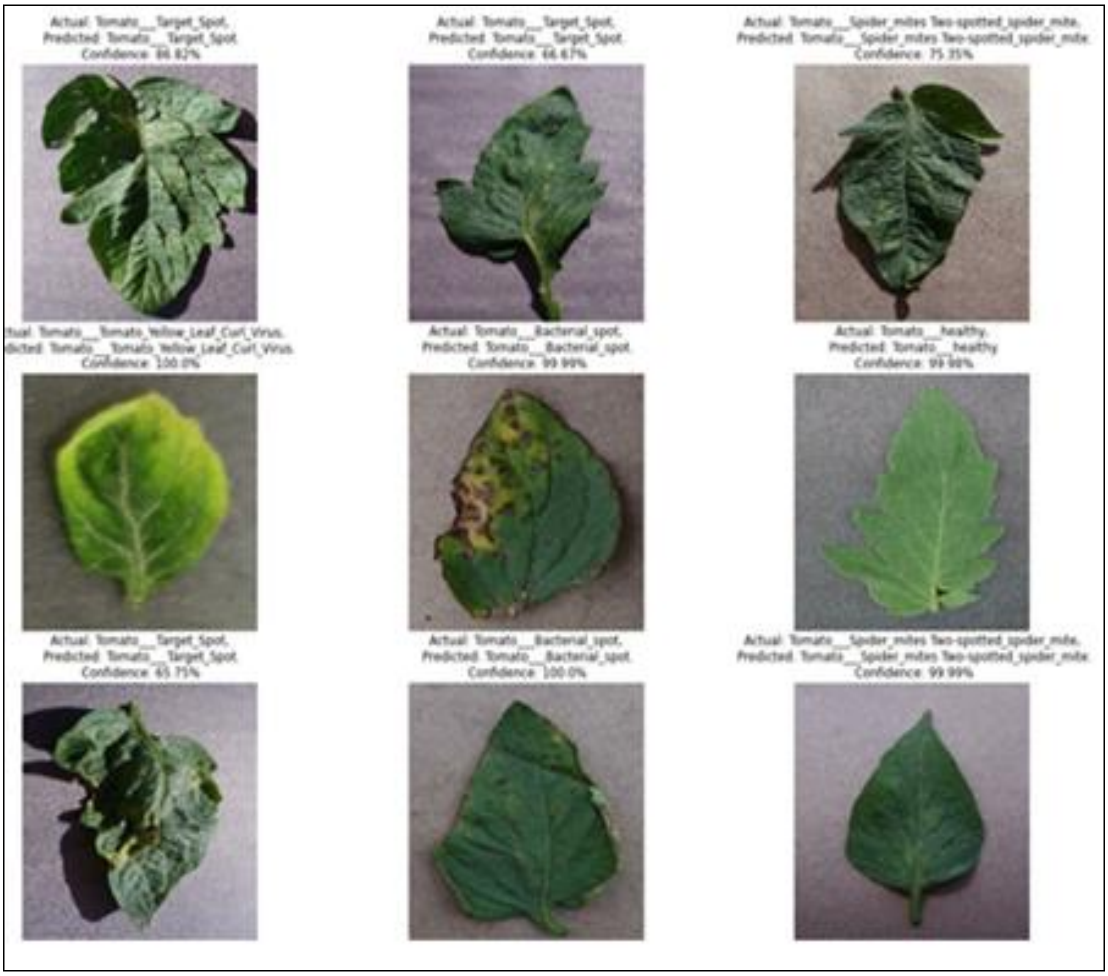

Fig. 10: Batch Testing with Confidence Level

C. Front End UI Development

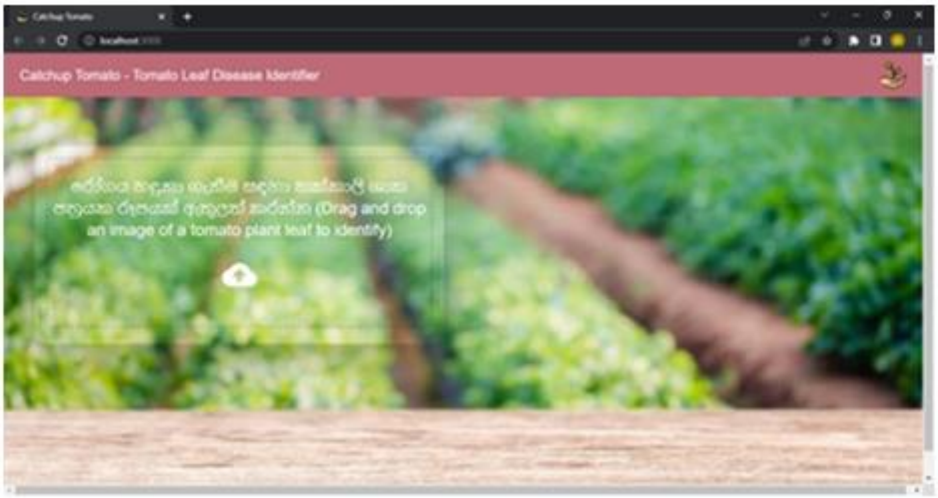

Fig. 11: Image Uploading Section

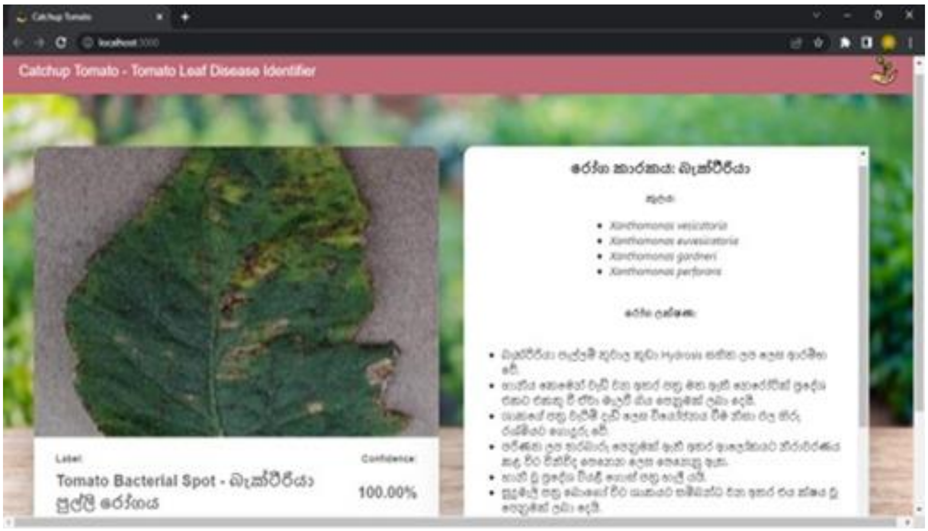

Fig. 12: Disease Prediction and Overview Section I

Fig. 13: Disease Prediction and Overview Section II

ISSN No:-2456-2165

V. DISCUSSION AND CONCLUSION

A. Significance of the obtained results

This study conducted a comprehensive evaluation of the Convolutional Neural Network (CNN) model for tomato leaf disease detection by comparing its performance across different training durations, specifically using 50 and 100 epochs.

➢ *Training and Testing with Different Epoch Counts:*

- **Training with 50 Epochs**: Initially, the CNN model was trained using 50 epochs. An epoch, in the context of machine learning, is one complete pass of the entire training dataset through the algorithm. The test results following this training phase revealed a highest accuracy of 90.75% and a corresponding loss value of 0.39. While this level of accuracy is commendable, the relatively higher loss value suggests that there was still room for improvement in the model's ability to generalize and accurately classify the images.
- **Training with 100 Epochs**: The model was then trained for a longer duration, using 100 epochs. This extended training was hypothesized to allow the model to better learn the nuances of the dataset and improve its predictive accuracy. The results from testing the model post this extended training phase confirmed this hypothesis, with the model achieving a significantly higher accuracy of 98.2% and a substantially lower loss value of 0.05. The improved accuracy and reduced loss indicate a more robust model, capable of effectively identifying diseases in tomato leaves with greater precision.

➢ *Comparative Analysis:*

- The increase in accuracy from 90.75% to 98.2% and the decrease in loss from 0.39 to 0.05 between the 50-epoch and 100-epoch training regimes underscore the importance of adequate training duration in machine learning.
- Longer training periods can lead to better model performance, as seen in the enhanced ability of the CNN model to accurately classify the leaf images with reduced error rates.
- However, it's also essential to monitor for overfitting, especially with increased epochs, where the model may start to learn noise and details specific to the training set that are not generally applicable.

In conclusion, this study demonstrates that extending the training duration of a CNN model can significantly enhance its performance, as evidenced by the higher accuracy and lower loss observed when using 100 epochs compared to 50 epochs in the context of tomato leaf disease detection.

	Pervious Systems					
	Research[1]	Research[2]	Research[3]	Research[4]	Research[5]	Proposed System
Dataset Weight	4923	3000	10696	4400	450	22930
Accuracy	91.67%	98.49%	91.81%	97.8%	76.1%, 55.0%, 55.6% (3 models)	98.2%
All Visible Features: Disease Confidence Level Pathogen and Genus Symptoms Disease Management	✗	✗	✗	✗	✗	✓

Fig. 14: Proposed System and Model Comparison with Previous Systems

B. Future Development

The recommendation to develop a mobile application for tomato leaf disease detection, in addition to a web application, reflects a strategic response to the evolving needs of modern farmers. This approach aligns with the increasing adoption of mobile technology in agriculture and offers several advantages and future development prospects:

➢ *Mobile Application Advantages:*

- **Accessibility and Convenience**: Mobile applications provide farmers with on-the-go access to disease detection tools, allowing for immediate and convenient use in the field.
- **Real-time Analysis**: With the integration of camera functionality, farmers can capture images of tomato leaves and receive instant disease diagnoses, facilitating timely intervention.

➢ *Extension to Other Crops:*

- **Advanced CNN Algorithms**: The current model, while effective for tomato leaf disease detection, can be expanded to identify diseases in other essential crops. This would involve training the CNN model on diverse datasets comprising various crop diseases.
- **Algorithmic Improvements**: Implementing advanced algorithmic techniques and deep learning mechanisms can further enhance the model's accuracy and versatility, making it a more comprehensive tool for agricultural disease management.

➢ *User Interface (UI) Enhancements:*

- **Crop Dropdown Feature**: Incorporating a dropdown menu in the UI where farmers can select different crops (in addition to tomato leaves) will broaden the application’s scope. This feature allows the model to be used for a wider range of plants, increasing its utility in the agricultural sector.

- **Compatibility Checks**: The application can be designed to analyze the uploaded image to confirm its compatibility with the selected crop or leaf disease. This feature ensures that the input data aligns with the model's capabilities, enhancing the accuracy of disease detection.
- ✓ **User-Centric Design**: In designing the mobile and web applications, a user-centric approach is essential. The interface should be intuitive and easy to navigate for farmers, who may have varying levels of technological proficiency.
- ✓ **Educational Component**: The application could also include an educational section with information about common diseases, prevention strategies, and best practices for crop management. This would not only aid in disease detection but also in promoting overall plant health.
- ✓ **Community and Support**: Integrating a community support feature where farmers can share insights, ask questions, and learn from each other can foster a collaborative environment.

In summary, the development of a mobile-friendly platform for disease detection in agriculture, coupled with advancements in CNN algorithms for broader crop applicability and user-friendly UI enhancements, represents a forward-thinking approach to leveraging technology in agricultural practices. This aligns well with the trend of digitalization in agriculture and addresses the practical needs of farmers in disease management and crop care.

C. Conclusion

In this research, a deep learning model utilizing a Convolutional Neural Network (CNN) was developed to identify diseases in tomato leaves with high accuracy. The key aspects of this research are outlined below:

- **Objective**: The primary goal was to create a reliable deep learning model capable of accurately identifying various diseases affecting tomato leaves. This was accomplished by leveraging the capabilities of CNNs, which are particularly well-suited for image recognition tasks.
- **CNN Architecture and Method**: The CNN method was chosen for its proficiency in handling image data, extracting features, and performing classifications. The intrinsic architecture of CNNs, including layers like Convolutional layers, ReLU, Max Pooling, Flatten, and Dense layers, plays a critical role in processing and learning from image data.
- **Dataset Acquisition**: The dataset, consisting of 22,900 images, was meticulously compiled. These images, all captured under controlled laboratory conditions, include nine different types of diseased-leaf images and one type representing healthy leaves. The variety in the dataset ensures a comprehensive training and testing of the model.
- **Data Splitting for Training, Validation, and Testing**: The dataset was divided following an 80-10-10 split ratio, where 80% of the images were used for training the model, 10% for validation, and the remaining 10% for testing. This distribution is a common practice in machine learning to train models effectively while also assessing their performance on unseen data.
- **Achieved Accuracy**: The model achieved a remarkable accuracy rate of over 98.2%. This high level of accuracy indicates the model's effectiveness in correctly identifying and classifying various diseases in tomato leaves. Achieving such a high accuracy rate is significant in the context of agricultural technology, where reliable disease detection can lead to timely and effective interventions.

In summary, this research successfully demonstrates the application of a CNN-based deep learning model for the identification of tomato leaf diseases. The model's high accuracy underscores its potential utility in the agricultural sector, particularly for farmers and agronomists who require reliable tools for plant health monitoring and disease management.

REFERENCES

[1]. Tomato processing in sl: Is it a viable remedy to manage surplus? daily news. https://www.dailynews.lk/2018/06/20/features/154441 / tomato-processing-sl-it-viable-remedy-manage-surplus, 2018. (Ac- cessed on 27/06/2022).

[2]. Scientific review of the impact of climate change on plant pests. *Scientific review of the impact of climate change on plant pests*, jun 2021.

[3]. FAO - News Article: Climate change fans spread of pests and threatens plants and crops, new FAO study.

[4]. Frank G. Zalom. Pests, endangered pesticides and processing tomatoes.

[5]. *Acta Horticulturae*, 613:223–233, 2003.

[6]. Lovepreet Kaur and Shiwani Guleria Sharma. Identification of plant diseases and distinct approaches for their management. *Bulletin of the National Research Centre*, 45(1), 2021.

[7]. Jun Liu and Xuewei Wang. Tomato Diseases and Pests Detection Based on Improved Yolo V3 Convolutional Neural Network. *Frontiers in Plant Science*, 11(June):1–12, 2020.

[8]. Muyesaier Tudi, Huada Daniel Ruan, Li Wang, Jia Lyu, Ross Sadler, Des Connell, and Cordia Chu. Agriculture Development , Pesticide Application and Its Impact on the Environment. pages 1–23, 2021.

[9]. Tiago Domingues, Toma´s Branda˜o, and Joa˜o C. Ferreira. Machine Learning for Detection and Prediction of Crop Diseases and Pests: A Comprehensive Survey. *Agriculture (Switzerland)*, 12(9):1–23, 2022.

[10]. Punitha Kartikeyan and Gyanesh Shrivastava. Review on emerging trends in detection of plant diseases using image processing with machine learning. *International Journal of Computer Applications*, 174(11):39–48, Jan 2021.

[11]. Automated image capturing system ... preview & related info https://www.mendeley.com/catalogue/ bd3baba4-bf30-3df8-9a5d-b487fecbf9a2/. (Accessed on 28/06/2022).

[12]. Vijaypal Singh Dhaka, Sangeeta Vaibhav Meena, Geeta Rani, Deepak Sinwar, Kavita, Muhammad Fazal Ijaz, and Marcin Woz´niak. A survey of deep convolutional neural networks applied for prediction of plant leaf diseases. *Sensors*, 21(14), 2021.
[13]. Aseem Patil and Milind Rane. Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition. Smart Innovation, Systems and Technologies, 195:21–30, 2021.
[14]. M. M. Chiampi and L. L. Zilberti, “Induction of electric field in human bodies moving near MRI: An efficient BEM computational procedure,” IEEE Trans. Biomed. Eng., vol. 58, pp. 2787–2793, Oct. 2011, doi: 10.1109/TBME.2011.2158315
[15]. Amador-Perez and R. A. Rodriguez-Solis, “Analysis of a CPW-fed annular slot ring antenna using DOE,” in Proc. IEEE Antennas Propag. Soc. Int. Symp., Jul. 2006, pp. 4301–4304
[16]. Cheemaladnne Vengaiah, M. Priyadharshini, "CNN Model Suitability Analysis for Prediction of Tomato Leaf Diseases", 2023 6th International Conference on Information Systems and Computer Networks (ISCON), pp.1-4, 2023.
[17]. Sameer Dev Sharma, Sonal Sharma, Abhishek Kumar Pathak, Nachaat Mohamed, "Real-time Skin Disease Prediction System using Deep Learning Approach", 2023 2nd Edition of IEEE Delhi Section Flagship Conference (DELCON), pp.1-6, 2023.
[18]. Saurabh Sharma, Gajanand Sharma, Ekta Menghani, Anupama Sharma, "A Comprehensive Review on Automatic Detection and Early Prediction of Tomato Diseases and Pests Control Based on Leaf/Fruit Images", International Conference on Cyber Security, Privacy and Networking (ICSPN 2022), vol.599, pp.276, 2023.
[19]. Prajwala Tm, Alla Pranathi, Kandiraju SaiAshritha, B. Chittaragi Nagaratna and G. Koolagudi Shashidhar, "Tomato Leaf Disease Detection Using Convolutional Neural Networks", 2018 Eleventh International Conference on Contemporary Computing (IC3), 2018.
[20]. R Swathika, S. Srinidhi, N Radha and K. Sowmya, "Disease Identification in paddy leaves using CNN based Deep Learning", theProceedings of the Third International Conference of ICICV, 2021.
[21]. Garima Shrestha, Deepsikha, Majolica Das and Naiwrita Dey, "Plant Disease Detection Using CNN", 2020 IEEE Applied Signal Processing Conference (ASPCON), 2020.
[22]. Tahmina Tashrif Mim, Md. Helal Sheikh, Roksana Akter Shampa, Md. Shamim Reza and Md. Sanzidul Islam, "Leaves Diseases Detection of Tomato Using Image Processing", 2019 8th International Conference System Modeling and Advancement in Research Trends (SMART), 2019.
[23]. Melike Sardogan, Adem Tuncer and Yunus Ozen, "Plant Leaf Disease Detection and Classification based on CNN with LVQ Algorithm", (UMBK'18) 3rd international conference on Computer Science and Engineering.
[24]. Garima Shrestha, Deepsikha, Majolica Das and Naiwrita Dey, "Plant Disease Detection Using CNN", 2020 IEEE Applied Signal Processing Conference (ASPCON), 2020.
[25]. G. Madhulatha and O. Ramadevi, "Recognition of Plant Diseases using Convolutional Neural Network", 2020 Fourth International Conference on I-SMAC (IoT in Social Mobile Analytics and Cloud) (I-SMAC), 2020.