RESEARCH ARTICLE OPEN ACCESS

Development of Vision Based Crop Health Monitoring Systems

Balambika, Mahizhini, Swetha
Sri Ramakrishna Engineering College

ABSTRACT
Crop disease detection using CNN algorithms is an innovative approach that harnesses the power of convolutional neural networks (CNNs) to distinguish between healthy and diseased crops. This method offers a rapid and accurate means of identifying crop ailments, facilitating timely interventions.
The methodology involves training a CNN model using a dataset comprising images of both healthy and diseased crops. Within this framework, the CNN model extract features from the input images and learns to classify them into distinct categories.
In conclusion, CNN-based crop disease detection holds significant promise for the early and precise identification of crop maladies. Its potential impact extends to improving crop yields and reducing economic losses stemming from diseases. Notably, this technique has been implemented in MATLAB, complemented by a mobile app for user-friendly disease detection.
KEYWORDS: convolutional neural networks (CNN), healthy and diseased crops, accuracy.

Date of Submission: 22-09-2023 Date of acceptance: 05-10-2023

I. INTRODUCTION

Remote sensing is an alternative approach for fast and unbiased disease scouting and measurement. Here, the common information carrier is electromagnetic (EM) radiation. The range of all types of EM radiation is known as the EM spectrum, which consists of a range of spectra from shorter wavelengths (e.g., gamma-rays) to longer wavelengths (e.g., radio waves). Various sensors such as RGB (or visible), multispectral and hyperspectral sensors are used to capture the different portions of the EM spectrum is referred in [1].

Today, convolutional neural networks (CNNs) are more capable than standard feature extraction methods. CNN is a deep learning network that performs at a high level and employs an end-to-end architecture and abandons the complicated procedures of image preprocessing and feature extraction, simplifying the identification process compared with its learning model counterparts [8].

Precision agriculture has emerged as a promising strategy to revolutionize traditional farming methods by leveraging technology and data-driven techniques. Its primary goals are to enhance crop productivity, reduce resource wastage, and encourage sustainable farming practices. A critical component of precision agriculture is the monitoring and management of crop health, which directly impacts yield and quality.

Deep Convolutional Neural Network is utilized in this study to identify infected and healthy leaves, as well as to detect illness in afflicted plants. The CNN model is designed to suit both healthy and sick leaves; photos are used to train the model, and the output is determined by the input leaf [2].

The development of this system involves several stages, including image acquisition, preprocessing, feature extraction, and analysis. Advanced computer vision algorithms and machine learning models are employed to extract meaningful information from images, enabling accurate crop health assessments. Training the system on a diverse dataset of healthy and diseased plants allows it to classify and diagnose various health conditions.

The implications of this system for precision agriculture are substantial. By providing real-time, objective, and actionable information about crop health, it empowers farmers to implement targeted interventions and optimize resource allocation. This can lead to improved crop yields, reduced use of fertilizers and pesticides, and ultimately, a more sustainable and economically viable farming practice.

II. CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Networks (CNNs) have emerged as a transformative breakthrough in the realm of deep learning, reshaping various fields such as speech recognition, computer vision, and natural language processing. Their remarkable ability to extract intricate patterns and features from complex input data, with a particular emphasis on images, has catapulted CNNs into the spotlight of machine learning in recent years. CNNs have proved to be not just proficient but revolutionary in a plethora of tasks including image segmentation, object detection, and image classification.

Let's delve deeper into the architecture of CNNs, unpacking their constituent elements and elucidating their roles. At the heart of CNNs are convolutional layers, pooling layers, and fully connected layers. Convolutional layers engage in the core operation of convolutions using filters, which are small matrices designed to detect specific features such as edges, corners, or textures. The result of these convolutions is a set of feature maps, each one highlighting the presence of a particular feature in the input data. To ensure that spatial information is retained during convolutions, techniques like padding and strides are employed. Padding adds extra pixels around the input data, while strides determine how much the filter moves across the data, influencing the size of the resulting feature maps.

Pooling layers, another integral component, step in to downsample the feature maps from convolutional layers. This downsampling process reduces spatial dimensions while preserving critical features. Common pooling techniques encompass max pooling, which selects the maximum value within a predefined window, and average pooling, which calculates the average value within a window, offering a smoothed representation of the feature maps. Strided pooling serves as an alternative to standard pooling with a larger stride, directly diminishing feature map dimensions.

Activation functions infuse non-linearity into the network, allowing it to learn intricate relationships between inputs and outputs. Frequently employed activation functions in CNNs include Rectified Linear Unit (ReLU), known for setting all negative values to zero, which accelerates convergence and mitigates the vanishing gradient problem. Sigmoid activation maps inputs to a range between 0 and 1, commonly suitable for binary classification tasks, while Tanh activation maps inputs to a range between -1 and 1, presenting a centered activation function symmetric around zero.

Fully connected layers typically appear at the end of the CNN architecture. These layers establish connections between every neuron from the previous layer to the subsequent one, ultimately enabling classification or regression tasks. They utilize activation functions and weight matrices to transform extracted features into the desired output format.

Training a CNN entails two pivotal steps: forward propagation and backpropagation. Forward propagation orchestrates the flow of input data through the network, layer by layer, yielding predicted outputs. Backpropagation, on the other hand, calculates the gradient of the loss function with respect to the network's weights. This gradient is then harnessed to update the weights through optimization algorithms such as stochastic gradient descent (SGD) and its variants.

Loss functions occupy a critical role in quantifying the dissimilarity between predicted outputs and actual labels. The choice of loss function hinges on the specific task, encompassing categorical cross-entropy for multi-class classification or mean squared error for regression.

Notwithstanding their remarkable successes, CNNs grapple with a set of challenges. These encompass issues related to interpretability, dataset biases, vulnerabilities to adversarial attacks, and the substantial demand for copious labeled data for effective training. In a forward-looking perspective, the future of CNN research holds promising prospects. Researchers are increasingly exploring the integration of CNNs with other architectural paradigms to further enhance their capabilities. Ethical considerations surrounding the use of CNNs are also emerging as a significant area of focus.

In conclusion, the evolution and profound impact of Convolutional Neural Networks on the landscape of deep learning and machine learning are undeniable. Their architecture, encompassing convolutional layers, pooling layers, activation functions, and fully connected layers, forms the backbone of their success. Training CNNs involves forward and backward propagation, with loss functions guiding the learning process. Regularization techniques combat overfitting, while ongoing research is dedicated to addressing challenges such as interpretability and dataset biases. The future of CNNs holds exciting possibilities, with their continued integration into other architectural frameworks and increased attention to ethical considerations. In the realm of image recognition and detection, CNNs have emerged as indispensable tools, driving progress across numerous applications and leveraging benchmark datasets to enhance performance and expand their utility.

III. DETECTION PART

3.1 CONVOLUTION:

The convolutional layer is a central component of a CNN, responsible for the majority of computation. When processing a color image as input, it is represented as a 3D pixel matrix with dimensions of height, width, and depth, analogous to the RGB channels of the image.

In this layer, a feature detector, often referred to as a kernel or filter, moves across receptive fields of the image to identify relevant features. This process is known as convolution. A feature detector is represented by a two-dimensional (2D) array of weights, capturing a portion of the image. The filter size, which determines the receptive field size, is typically a 3x3 matrix, but it can vary.

During convolution, the filter is applied to a portion of the image, and the dot product between the input pixels and the filter is calculated. The resulting dot product is then fed into an output array. This process is repeated as the filter sweeps across the entire image, shifting by a specified stride.

The output of this series of dot products between the input and the filter is referred to as a feature map, activation map, or convolved feature. It represents the learned features extracted from the input image by the convolutional layer.

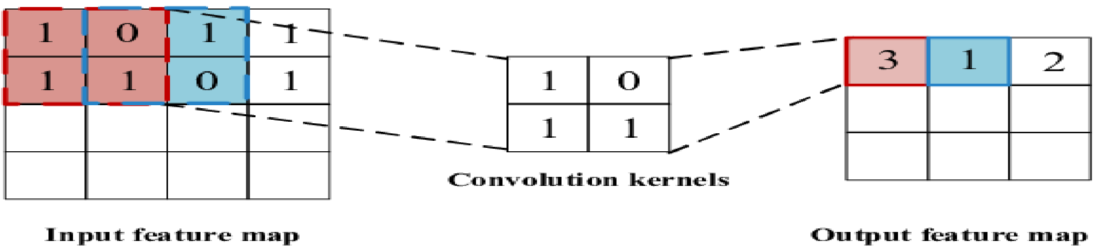

FIGURE 1. Convolution layer [2]

3.2 POLLING:

Convolutional neural networks' building blocks include pooling layers. Pooling layers combine the features discovered by CNNs, whereas convolutional layers retrieve features from images. Its goal is to gradually reduce the spatial dimension of the representation in order to reduce the quantity of parameters and calculations in the network. The filters of the convolutional layers provide a location-dependent feature map.

In certain cases, the Convolutional layer may struggle to identify an object in an image if it has undergone slight movements or translations. This limitation arises because the feature map produced by the Convolutional layer encodes the precise locations of features in the input image.

To address this issue, pooling layers provide a property known as "Translational Invariance." This means that even if the input image is translated or shifted, the CNN remains capable of recognizing the inherent characteristics within the image. Pooling layers help achieve this by reducing the spatial dimensionality of the feature maps, effectively capturing the essential features while discarding precise positional information. As a result, the CNN can still recognize and distinguish features even when they have been translated or slightly shifted within the image.

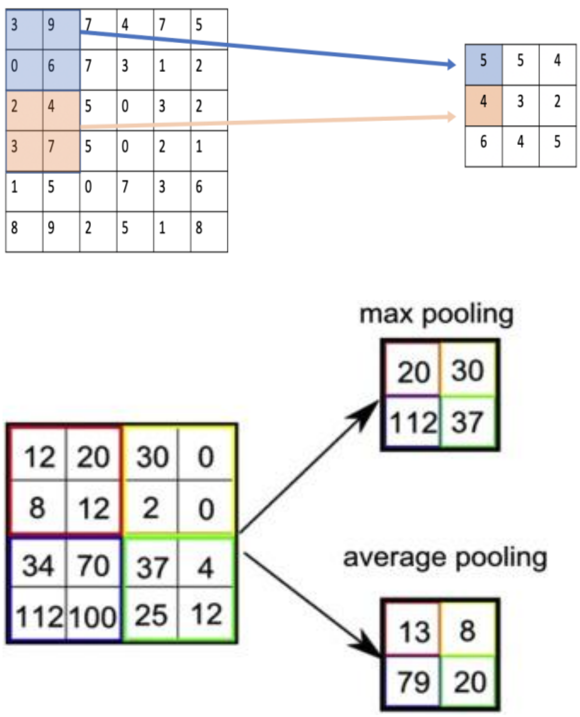

FIGURE 2. Pooling layer [2]

Only a small percentage of the outputs from the preceding layer are sent to each neuron in the convolutional layer after they have been convolved with a "kernel." The "receptive field" of a neuron

ISSN: 2248-9622, Vol. 13, Issue 10, October 2023, pp 14-20

refers to the set of output values that neuron may perceive. The "pooling layer" is the second primary structure. It creates a single neuron from each group of the outputs from the preceding layer. The average pooling and max pooling versions of pooling techniques are frequently used. An average pooling layer takes the mean of its input data and averages it. Max pooling, however, captures the greatest benefit.

3.3 FLATTERING:

The flattening stage is an essential step in building a convolutional neural network (CNN) and is remarkably straightforward. Its purpose is to transform the pooled feature map, generated during the pooling process, into a one-dimensional vector. This can be visualized as follows:

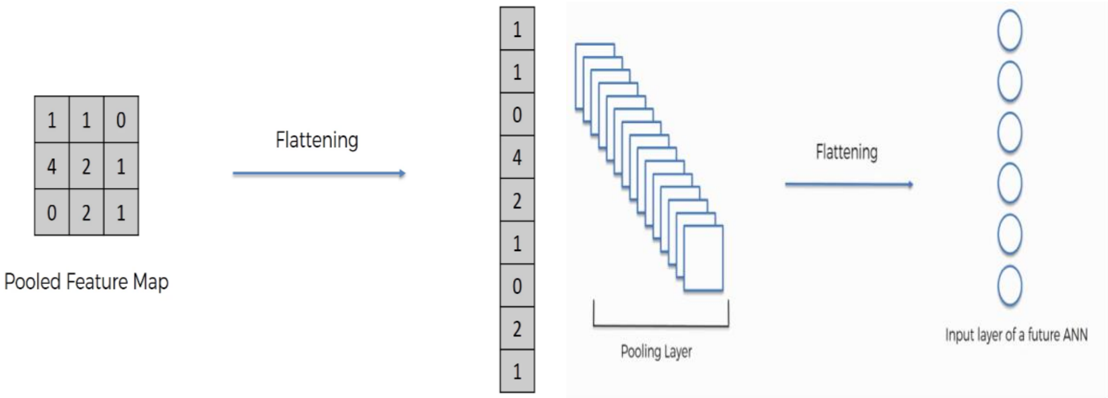

FIGURE 3. Flattering process

We convert the pooled feature map into a one-dimensional vector because an artificial neural network will now use this vector as input. To put it another way, the convolutional neural network we've been developing throughout this course will now be chained onto this vector, which will serve as the input layer of an artificial neural network.

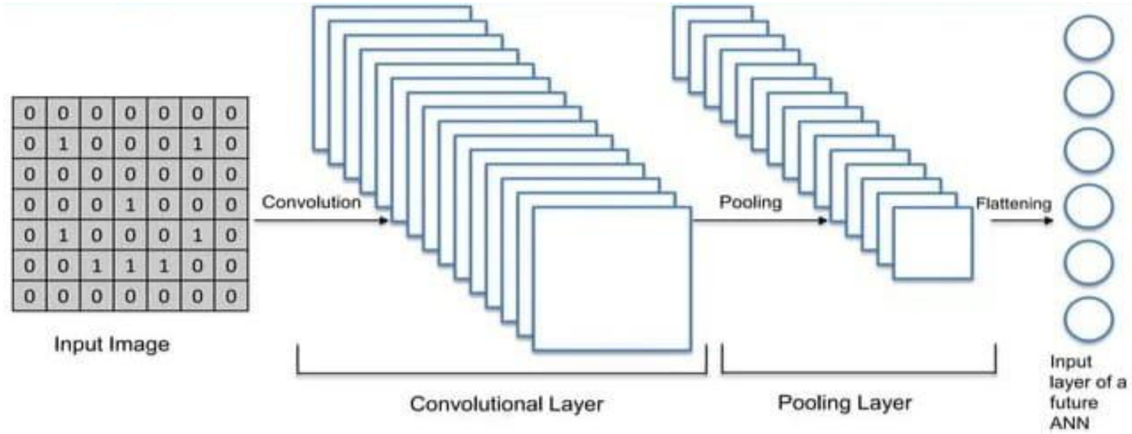

FIGURE 4. Diagram of convolution and pooling layer

3.4 FULLY CONNECTED:

The fully connected stage involves integrating an artificial neural network with our existing convolutional neural network, as indicated in the previous section. In this stage, the hidden layer of the artificial neural network is substituted with a specific type of hidden layer called a fully connected layer.

In a convolutional neural network, the primary task of the fully connected layer is to recognize specific features within an image. Each neuron in the fully connected layer is associated with a particular feature that could potentially be present in the image. The value transmitted by each neuron to the subsequent layer represents the probability or confidence that the associated feature is indeed present in the image.

3.5 OUTPUT LAYER:

We would require the result in the form of a class following numerous layers of convolution and padding. Only features and fewer parameters could be extracted from the source images by the convolution and pooling layers. However, in order to get the required number of classes, we must apply a completely linked layer to the final output. Reaching that number using only the convolution layers gets challenging. While we only require the output to indicate whether or not a picture belongs to a specific class, convolution layers produce 3D activation maps. To calculate the error in prediction, the output layer uses a loss function similar to categorical cross-entropy.

Here we can see the final process of how the following procedures proceed in the form of image of the leaf (FIGURE 5)

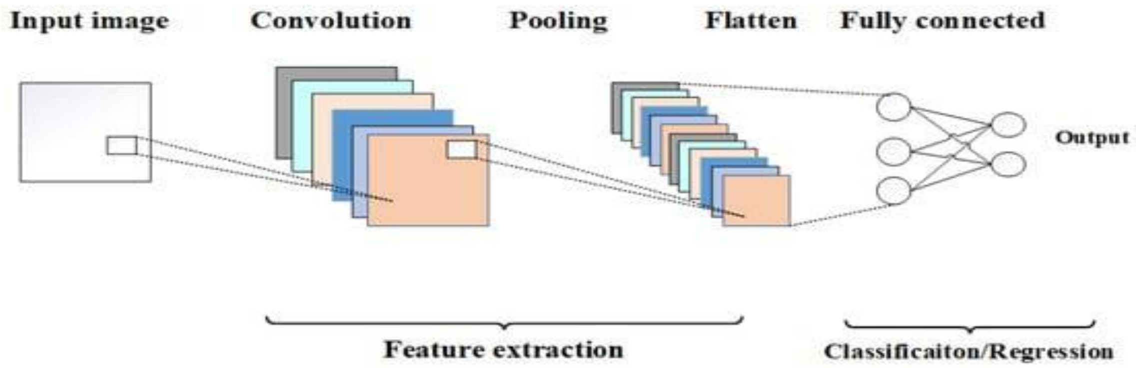

FIGURE 5. Illustration of a typical convolutional neural network (CNN)[1].

IV. METHODOLOGY

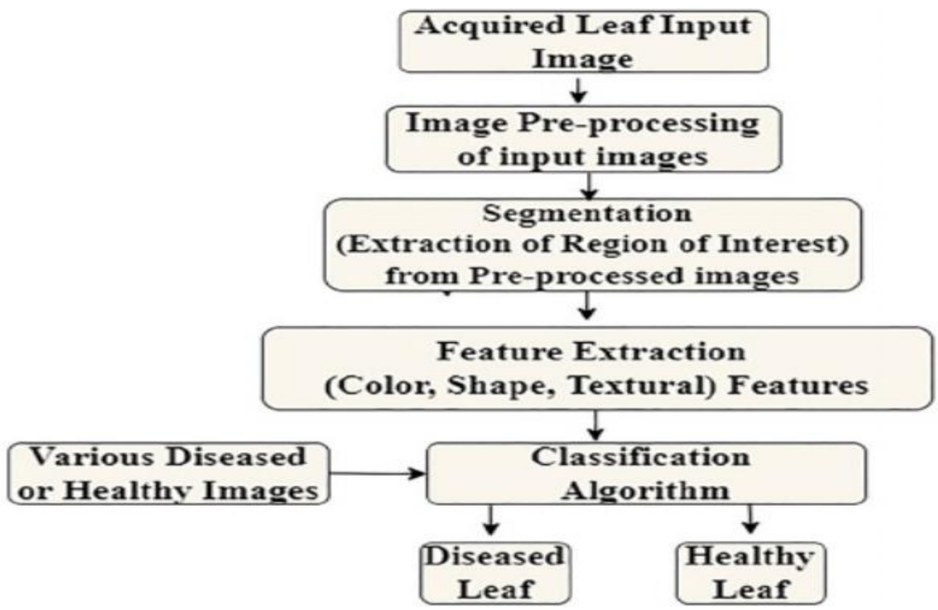

FIGURE 6. Flow chart of disease classification systems [4].

The plant disease detection process begins with data collection and preprocessing, involving gathering a diverse dataset of plant images, standardizing their dimensions, and normalizing pixel values. The dataset is then split into training, validation, and testing sets. A Convolutional Neural Network (CNN) architecture is designed with multiple convolutional and pooling layers, and hyperparameters are fine-tuned during training using backpropagation and SGD. Model performance is assessed with metrics like accuracy and F1 score, followed by hyperparameter adjustments if needed. Finally, the well-trained model can be deployed with a user-friendly interface for practical use in diagnosing plant diseases.

Further split the images for
TRAINING: 80%
TESTING: 20%
VALIDATION: 10%

V. RESULTS AND DISCUSSION

In our assessment of works focused on the automated categorization of healthy and unhealthy crop leaves using Convolutional Neural Networks (CNNs), we identified several critical challenges and shortcomings. Furthermore, we offered practical recommendations and guidelines to harness the full potential of CNNs in real-world applications. It has come to our attention that many previously published CNN-based solutions may not be readily deployable in agricultural settings due to their failure to adhere to fundamental machine learning principles.

One of the primary concerns stemming from this non-compliance is the potential weakness in the generalization skills of these models when faced with unknown data samples or different imaging conditions. This limitation significantly restricts the practical utility of these trained models.

Nevertheless, our research underscores the significant promise held by deep learning techniques in the domain of crop disease identification. While there are now more advanced solutions for disease detection and categorization, these innovations have notably improved accuracy levels. Our efforts have yielded predictions with accuracies ranging from 85% to 95%, which establishes a solid foundation for crop leaf assessment.

While our work represents a promising starting point for predicting crop leaf health, there remains ample room for further development and refinement. By enhancing the model's predictive capabilities to accurately identify the specific type of disease afflicting the plant, we can provide farmers with invaluable insights to maximize crop production. This timely and precise diagnostic tool has the potential to positively impact agricultural productivity, ensuring that farmers receive the correct solution at the right moment, minimizing disruptions to their operations.

The encouraging outcomes of our research underscore the potential for the creation of innovative agricultural technologies that can enhance crop management practices and contribute to the sustainable growth of the agricultural sector.

This implementation spans both MATLAB and python platforms, ensuring accuracy surpassing 80% with IoT integration, we create a farmer-friendly application, enabling easy disease detection. This fusion of technologies enhances agricultural practices by delivering effective crop disease management solutions.

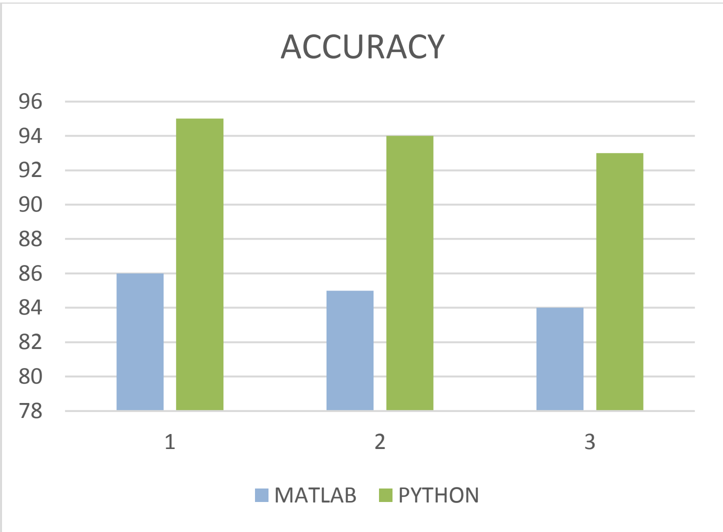

FIGURE 7. Chart representation of accuracy for MATLAB and python

ACKNOWLEDGEMENT

"We would like to express our heartfelt gratitude to Mr. C. Dinesh, Assistant Professor of the Department of Aeronautical Engineering at Sri Ramakrishna Engineering College, for his invaluable guidance, unwavering support, and encouragement in helping us transform our project idea into a published paper." We would also like to like to express our deepest gratitude to Dr. R. Rangaraj, Assistant Professor of the Department of Aeronautical Engineering at Sri Ramakrishna Engineering College, whose inspiring guidance and valuable opinion supported us all through the way up to completion of our project.

REFERENCES

[1]. Recent Advances **in Crop Disease Detection Using** UAV and Deep Learning TechniquesTB Shahi, CY Xu, A Neupane, W Guo - Remote Sensing, 2023 - mdpi.com
[2]. Plant Disease Detection Using Cnn Nishant Shelar1 , Suraj Shinde2 , Shubham Sawant3 ,

Shreyash Dhumal4 , and Kausar Fakir5,2022 - ICACC

[3]. A survey on **using** deep learning techniques for **plant disease** diagnosis and recommendations for development of appropriate tools A Ahmad, D Saraswat, A El Gamal - Smart Agricultural Technology, 2023 - Elsevier

[4]. A Systematic review of recent machine learning techniques for **plant disease identification** and classification L Goel, J Nagpal - IETE Technical Review, 2023 - Taylor & Francis

[5]. Deep learning system for paddy **plant disease detection** and classification A Haridasan, J Thomas, ED Raj - Environmental Monitoring and …, 2023 - Springer

[6]. AgriDet: **Plant** Leaf **Disease** severity classification **using** agriculture **detection** fra mework A Pal, V Kumar - Engineering Applications of Artificial Intelligence, 2023 - Elsevier

[7]. Peanut leaf **disease identification with** deep learning algorithms L Xu, B Cao, S Ning, W Zhang, F Zhao - Molecular Breeding, 2023 – Springer

[8]. An Efficient Hybrid CNN Classification Model for Tomato **Crop Disease** MV Sanida, T Sanida, A Sideris, M Dasygenis - Technologies, 2023 - mdpi.com

[9]. **Image**-based assessment of **plant disease** progression identifies new genetic loci for resistance to Ralstonia solanacearum **in** tomato V Meline, DL Caldwell, BS Kim, RS Khangura… - The **Plant** …, 2023 - Wiley Online Library

[10]. An optimal hybrid multiclass SVM for **plant** leaf **disease detection using** spatial Fuzzy C-Means model
SK Sahu, M Pandey - Expert Systems **with** Applications, 2023 – Elsevier