An efficient tomato diseases detection and classification methodology using CNN Deep Learning Network

Uma metodologia eficiente de detecção e classificação de doenças do tomate usando CNN Deep Learning Network

Una metodología eficiente para la detección y clasificación de enfermedades del tomate utilizando la Red de Aprendizaje Profundo de CNN

DOI:10.38152/bjt7n2-002

Submitted: Jan 19th, 2024
Approved: Apr 20th, 2024

Sofiane Tchoketch Kebir
Graduated in Signal Processing
Institution: Medea University
Address: Medea, Algeria
E-mail: tchoketchkebir.sofiane@univ-medea.dz

Fouaz Berrhail
Graduated in Computer Science
Institution: University of SETIF 1
Address: Setif, Algeria
E-mail:berrhail.fouaz@univ-setif.dz

Faouzi Didi
Graduated in Renewables Energies
Institution: University Yahia Futures of Medea
Address: Medea, Algeria
E-mail: didifouzi19@gmail.com

ABSTRACT
In this paper, an efficient methodology for detecting and classifying tomato diseases using a Convolutional Neural Network (CNN) deep learning network is presented as an efficient aided tool to classify different tomato diseases based on their leaf appearance, where plant diseases and insects are considered as a main challenges for farmers to overcome. The proposed methodology structure is based on 20 layers using convolution, Maxpooling, Batch normalization and ReLU process as main operations in the adopted architecture. The obtained results using Plant Village database show that our proposed methodology outclasses the best recent methods of tomato diseases detection and classification with scores of 97.3%, 96.8%, 97.0% and 97.5% for precision, Recall, F1-Score and accuracy coefficient criteria respectively. Our methodology of tomato diseases detection and classification is found as effective, accurate and aided diagnostic tool that aims to aid farmers to make a precise plant treatments and enhancing productivity while promoting environmental sustainability.

Keywords: tomato plant disease detection, tomato plant disease classification, Deep Learning techniques DL, Convolutional Neural Networks CNN.
RESUMO
Neste artigo, uma metodologia eficiente para detectar e classificar doenças do tomate usando uma rede de aprendizagem profunda Convolutional Neural Network (CNN) é apresentada como uma ferramenta eficiente para classificar diferentes doenças do tomate com base na aparência das folhas, onde doenças de plantas e insetos são considerados como um dos principais desafios a serem superados pelos agricultores. A estrutura da metodologia proposta é baseada em 20 camadas utilizando convolução, Maxpooling, normalização em lote e processo ReLU como operações principais na arquitetura adotada. Os resultados obtidos usando o banco de dados Plant Village mostram que nossa metodologia proposta supera os melhores métodos recentes de detecção e classificação de doenças do tomate com pontuações de 97,3%, 96,8%, 97,0% e 97,5% para critérios de precisão, Recall, F1-Score e coeficiente de precisão, respectivamente. Nossa metodologia de detecção e classificação de doenças do tomate é considerada uma ferramenta de diagnóstico eficaz, precisa e auxiliada que visa auxiliar os agricultores a fazer tratamentos precisos das plantas e aumentar a produtividade, ao mesmo tempo que promove a sustentabilidade ambiental.

Palavras-chave: detecção de doenças do tomateiro, classificação de doenças do tomateiro, Técnicas de aprendizagem profunda DL, Redes Neurais Convolucionais CNN.

RESUMEN
En este trabajo se presenta una metodología eficiente para la detección y clasificación de enfermedades del tomate, utilizando una red de aprendizaje profundo de la Red Neuronal Convolucional (CNN), como una herramienta eficiente para clasificar diferentes enfermedades del tomate con base en la aparición de las hojas, donde las enfermedades de plantas e insectos son consideradas como uno de los principales retos a superar por los agricultores. La estructura de la metodología propuesta se basa en 20 capas utilizando la convolución, Maxpooling, normalización por lotes y el proceso ReLU como operaciones principales en la arquitectura adoptada. Los resultados obtenidos utilizando la base de datos de Aldea Vegetal muestran que nuestra metodología propuesta supera los mejores métodos recientes de detección y clasificación de enfermedades del tomate con puntuaciones de 97,3%, 96,8%, 97,0% y 97,5% para los criterios de precisión, Recall, F1-Score y coeficiente de precisión, respectivamente. Nuestra metodología para la detección y clasificación de enfermedades del tomate se considera una herramienta de diagnóstico eficaz, precisa y asistida que tiene como objetivo ayudar a los agricultores a realizar tratamientos de plantas precisos y aumentar la productividad al tiempo que se promueve la sostenibilidad ambiental.

Palabras clave: detección de enfermedades del tomate, clasificación de enfermedades del tomate, técnicas de aprendizaje profundo DL, redes neuronales convolucionales CNN.

1 INTRODUCTION

Due to the technological development that we have witnessed in recent years and the new technologies used in the agricultural sector, a new horizon has been opened in optimizing production, enhancing sustainability, and addressing the global challenge of food security. Innovations such as precision farming, which utilizes new technology systems and methods, enable farmers to monitor and manage their crops more efficiently[1] [2] [3]. Nevertheless, many challenges remain significant obstacles in the agricultural sectors,
where plant diseases and insects are considered as the main challenges for farmers to overcome. However, the integration of artificial intelligence techniques into agricultural practices offers promising solutions to these problems, where diagnostic tools based on artificial intelligence can correctly and quickly identify plant diseases. This type of plant diagnosis tool leads farmers to adopt the correct solutions, precise plant treatments, and keep plant diseases localized before they spread in agricultural fields. Additionally, the adoption of these new technologies not only helps to increase productivity but also promotes environmental sustainability by minimizing chemical treatments once diseases are well identified.

Artificial intelligence (AI) [24-25] is a set of techniques and methods that prepare machines to emulate human intelligence and equip computers with cognitive abilities similar to human brains. Automated training is characterized by the use of algorithms capable of automatically optimizing a volume of data without relying on predefined models or direct programming. This ability to train directly from data allows for continuous adaptation and improvement of machine performance, emphasizing the importance of the quantity and quality of data during the training phase [4]. All artificial intelligence approaches and methods can be categorized into supervised, unsupervised, and reinforcement methods, providing different strategies for data delivery and knowledge extraction [5] [6] [7].

- **Supervised learning:** is a machine learning method aimed at developing a model capable of predicting or classifying unknown data from a set of labeled examples and associating the inputs with the desired outputs. The process is divided into two main phases: the learning phase and the testing phase. During the training phase, the algorithm is trained with a set of labeled data, where each example is associated with a specific label. The testing phase then involves using this model to predict the labels of new, previously unseen data, thereby evaluating the generalization ability of the algorithm. Learning effectiveness is measured by the accuracy of the algorithm’s predictions compared to the actual labels provided by the experts.

- **Unsupervised Learning:** This machine learning modality is deployed in the absence of supervision by experts. It focuses on the structuring and categorization of information without prior knowledge of the labels or classes assigned to the data. The main objective is the identification of hidden structures or unknown patterns within the data, thus facilitating the understanding and exploitation of unlabeled information.

- **Reinforcement Learning:** This learning approach is characterized by the
interaction of an algorithmic agent with its environment, with the aim of optimizing a reward function through the selection of sequences of actions. The agent learns to execute actions that maximize cumulative rewards, based on the feedback (rewards or penalties) received from the environment following its actions. This method finds significant applications in various fields such as robotics, game theory, and autonomous vehicle navigation, where it allows the agent to adapt its strategies autonomously to achieve its goals.

Motivated by making a contribution to improving plant disease identification and classifications, this paper introduces an efficient methodology for tomato diseases detection and classification using a CNN deep learning network as a fast, effective, and accurate method as a tomato aided diagnostic tool. The paper is organized as follows: Section 2 presents a review of related works. Section 3 describes the different Deep Learning techniques and, more specifically, Convolutional Neural Networks approaches. Section 4 focuses on materials and the proposed methodology where the used database and the evaluation criteria for the proposed methodology are detailed. Section 5 presents experimental results and discussions. Finally, conclusions are given in Section 6.

2 RELATED WORKS

Over the last decade, the integration of artificial intelligence into information and data classification techniques has been found to be a promising approach. For this reason, many researchers have focused their ideas and techniques on using this technology to produce new systems and methodologies as accurate and efficient approaches for information and data classification. Authors in [8] proposed a hybrid approach that is based on object detection and pretrained CNN architectures technique for Tomato Disease Classification in a Natural Environment. The adapted techniques are based on three phases: tomato leaf detection in natural environments, the classification step for each leaf detected, and finally, tomato disease prediction. Experimental results using the Plant-Village database achieved accuracy, recall, and F1-score around 90.7%. In [9], authors present another method of tomato plant diseases detection based on CNN deep learning technique and other pretrained networks such as ResNet152 and EfficientNet-B3. The obtained results show accuracy rates of 88.32%, 93%, and 97% for 5 layers of CNN network, ResNet152, and EfficientNet-B3 architectures respectively. A multiple plants diseases classification using transfer learning based on the Densenet-121 architecture is introduced in [10] to classify 29 different kinds of diseases manifesting in potatoes, bell peppers, tomatoes, corn, cherries,
apples, and grapes. The proposed methodology achieved an accuracy of around 94.96% for plants diseases classification. Authors in [11] proposed hybrid techniques for tomato plant classification using transfer learning based on the DenseNet121 model and Conditional Generative Adversarial approach (C-GAN) to generate synthetic plant images. The adapted technique was trained and validated using the Plant-Village database, with the obtained results achieving an accuracy of 99.51% for 3 tomato disease classes, 98.65% for 5 tomato disease classes, and 97.11% for 10 tomato classes. A smart approach has been proposed in [12] using pre-trained Inception Version 3 and ResNet Version 2 networks to classify tomato diseases. The most important accuracy results obtained were 99.22% with specific configurations for the pre-trained models Inception V3 and ResNet V2 with a 50% and 15% dropout rate respectively. This important result is based on only 5225 images from the Plant-Village datasets, and the authors recommend validating their proposed methodology using all the content of the Plant-Village datasets. In [13], authors present an improved deep learning technique based on the CNN network model for tomato plant leaves diseases classification. The proposed model is considered as the simplest model among all the available models with four convolution layers followed by a max-pooling layer where the input layer is structured to reduce the size of images to 150x150 pixels. The proposed model achieves an accuracy value around 98.19% and 97.57% for the training and validation process respectively. In the literature, many pre-trained models and CNN architectures were adapted by researchers in their methodologies for classifying tomato plant diseases [14], such as the method depicted in [15] using the VGG16 network that achieved an accuracy around 96.19%. Another pre-trained network used in [16] is based on the DensNet121 Xception network as a transfer learning task, achieving an accuracy of 97.1% to classify the different tomato plant diseases.

3 DEEP LEARNING TECHNIQUES

Deep learning (DL) is a subcategory of machine learning characterized by the ability to enable computer systems to gain expertise from empirical data, closely imitate human experiential learning. Unlike traditional artificial intelligence methodologies, which depend on explicit mathematical models for data processing, DL relies on artificial neural networks with multiple levels of representation and transformation of information, carrying out these operations through non-linear processing structures. These networks leverage a layered architecture to perform feature extraction and manipulation, whether from supervised or unsupervised environments.
Deep learning techniques and methods encompasses multiple architectures and models, each designed to process different types of data and solve various problems. The main types are:

- **Convolutional Neural Networks (CNNs):** Specially designed to process grid-structured data, such as images, they use convolution operations to extract spatial features and reduce the dimensionality of the data.

- **Recurrent Neural Networks (RNNs):** Ideal for processing sequences of data, such as text or time series. RNNs have the particularity of keeping information from previous inputs in memory to influence the processing of future inputs.

- **Long Short-Term Memory (LSTM) Neural Networks:** An advanced variant of RNNs, specially designed to avoid the problem of gradient disappearance or explosion, making them more efficient at learning about long-term dependencies.

- **Gated Recurrent Units (GRUs):** Similar to LSTMs, but with a simplified structure. They combine multiple gates into one, reducing model complexity and computational resources required.

- **Generative Adversarial Networks (GANs):** They consist of two networks, a generator and a discriminator, which are trained simultaneously in an adversarial game. GANs are often used to generate realistic synthetic data, including images.

- **Autoencoders:** Used for unsupervised learning, they aim to learn a compressed representation (encoding) of the data, and then reconstruct the data from this representation.

- **Capsule Networks:** Offering an alternative to CNNs, capsule networks attempt to model the spatial hierarchy between the parts and the whole in images, aiming to improve the recognition and understanding of objects under various viewing conditions.

- **Transformers:** Introducing the concept of attention, these models revolutionized natural language processing (NLP). They enable more efficient modeling of long-range dependencies in textual data than RNNs and LSTMs.

Each type of deep learning model has specific advantages depending on the nature of the data and the problem to be solved, making deep learning extremely versatile and powerful in the field of artificial intelligence.

### 3.1 CONVOLUTIONAL NEURAL NETWORKS

Computer vision is experiencing rapid development, largely driven by advances in
deep learning. In this context, the concept of convolutional neural networks (CNN) emerges as an essential pillar, due to its extensive application. CNNs stand out in various computer vision applications, such as facial recognition and image classification, thanks to their ability to learn parameters such as weights and biases, similar to traditional neural networks[17]. This similarity highlights the adaptability of CNNs to complex model for computer vision problems.

The minimum necessary configuration of a CNN network contains six layers; beyond this configuration, the operation of a CNN network is impossible. This minimum configuration is based on:

- **Input layer of CNN**: The input layer in CNN must contain data describing the signal or image to be processed. These data must be represented by their original dimensions.

- **Convolution layer**: In the field of image processing and deep learning, the convolution layer plays a central role, often referred to as the feature extraction layer. This name highlights its ability to identify and isolate the salient attributes of an image, thus facilitating the image analysis and recognition processes. The operation of this layer is initiated by the selection of a specific region of the image, which is then processed by a convolution operation according to the following equation:

\[
 x(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n)
\]

(1)

Where:

- \( x(i, j) \) is the output pixel at position \((i, j)\) in the feature map.
- \( I \) is the input to the matrix or part of the image we are processing.
- \( K \) is the convolution kernel or filter applied to the input.
- \( \sum_m \sum_n \) represents the sum over all valid positions of \( m \) and \( n \) where the kernel \( K \) is applied on the input \( I \).
- \( I(m, n) \) is the value of the input pixel at position \((m, n)\).
- \( K(i - m, j - n) \) is the value of the convolution kernel at the adjusted position according to the applied offset from the output position \((i, j)\).

The result is a single value that helps form the output volume, representing a map of features extracted from the image. To ensure complete coverage of the image, the filter is moved across the entire image in regular increments, called steps or strides, allowing the convolution process to be repeatedly applied to new areas. This iterative process transforms the initial image into a set of feature maps that encapsulate essential information about the visual patterns present, such as edges, textures, or shapes. Through this transformation,
convolutional neural networks (CNNs) are able to perform image classification, detection, and segmentation tasks with remarkable accuracy, leveraging the extracted features to make decisions based on visual content.

- Pooling Layer: The pooling layer is used to reduce the spatial volume of the input image after convolution. It is used between two convolution layers. If we apply a fully connected (FC) layer after the convolution layer without applying pooling or max pooling, the computation will be expensive. Thus, maximum pooling is the only way to reduce the spatial volume of the input image by encoding the information. The main used functions are:
  - Max Pooling: This operation selects the maximum of values in a defined neighborhood region.

\[
MaxPooling(x_{i,j}) = \max_{(p,q) \in R_{i,j}} x_{p,q}
\]

- Average Pooling: This operation calculates the average of the values in a defined neighborhood region.

\[
AveragePooling(x_{i,j}) = \frac{1}{|R_{i,j}|} \sum_{(p,q) \in R_{i,j}} x_{p,q}
\]

- Sum Pooling: This operation calculates the sum of values in a defined neighborhood region.

\[
SumPooling(x_{i,j}) = \sum_{(p,q) \in R_{i,j}} x_{p,q}
\]

- Min Pooling: This operation selects the minimum of values in a defined neighborhood region.

\[
MinPooling(x_{i,j}) = \min_{(p,q) \in R_{i,j}} (x_{p,q})
\]

Where

\(x_{i,j}\) represents the element at position \((i,j)\) of the input matrix and \(R_{i,j}\) is the neighborhood region (window) around the element \((i,j)\).
• **Fully Connected layer:** A fully connected layer involves weights, biases, and neurons. It connects neurons in one layer to neurons in another layer. It is used to classify images into different categories by training. This process can be defined by the mathematical equation below:

\[ y = f(W.x + b) \] (6)

Where:

- \( y \): represents the output vector of the fully connected layer.
- \( W \): is the weight matrix of the layer.
- \( x \): is the input vector to the layer.
- \( b \): is the bias vector added to the weighted sum of inputs.
- \( f \): is the activation function applied element-wise.

• **Softmax layer:** The Softmax or logistic layer represents the final layer within a Convolutional Neural Network (CNN), situated at the terminus of the Fully Connected (FC) layer. The logistic layer is predominantly employed for binary classification tasks, while Softmax is utilized for multi-classification purposes.

\[ \text{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \] (7)

• **Output layer:** The output layer encapsulates the encoded label information.

4 MATERIALS AND THE PROPOSED METHODOLOGY

4.1 THE USED DATABASE

The Plant-Village database is a collection of images designed to support plant disease research and automatic diagnosis using deep learning techniques. For tomato plants, the PlantVillage database contains 16,011 images showcasing various plant diseases, alongside images of healthy tomato plants. These images are categorized into 10 classes based on the type of disease (Figure 1), facilitating the development and training of models capable of recognizing and distinguishing these diseases.

Tomato diseases are diverse and can impact various parts of the plant, with symptoms often manifesting first on the leaves. These symptoms may affect the shape,
color, or external appearance of the tomato plant’s leaves [18]. Common diseases include:

- **Bacterial Spot**: mainly affects tomato leaves and fruit, causing dark, sunken spots with a yellow halo.
- **Early Blight**: is a fungal disease that affects tomato leaves, causing dark concentric rings on lower leaves.
- **Late Blight**: is another fungal disease that affects both tomato leaves and fruit. It manifests as water-soaked lesions that turn brown, leading to rapid plant defoliation.
- **Leaf Mold**: is a fungal disease that affects tomato leaves, causing yellow patches on the upper leaf surface and a fuzzy white growth on the underside.
- **Septoria Leaf Spot**: is a common fungal disease that affects tomato leaves, leading to small, circular spots with dark borders. It can cause leaf yellowing and defoliation.
- **Spider Mites**: are tiny arachnids that feed on tomato plants, causing stippling on leaves, webbing, and overall decline in plant health.
- **Target Spot**: caused by fungi, affects tomato leaves with circular lesions that have a target-like appearance.
- **Tomato Yellow Leaf Curl Virus**: is a viral disease transmitted by whiteflies that affects tomato plants, causing yellowing and upward curling of leaves, stunted growth, and reduced fruit production.
- **Tomato Mosaic Virus**: is a viral disease that causes mottled patterns on tomato leaves, stunted growth, and reduced fruit quality.
4.2 EVALUATION CRITERIA

In deep learning classification techniques, many metrics are proposed to evaluate the performances of the proposed and used approaches [19] [20] [21]. In this work, we are interested in various performance metrics, mainly accuracy, precision, recall, and F1-score. These metrics are described below:

- **Accuracy**: This parameter is crucial to globally evaluate the performance of a classification model. It represents the proportion of correct predictions among all the predictions made by a classification model, measuring the ability of the model to correctly predict classes, regardless of their label. The formula for accuracy is:

\[
\text{Accuracy} = \frac{TP + TN}{Total\ Predictions} \tag{8}
\]

- **Precision**: This parameter is an essential in the classification process to evaluate the relevance of the positive predictions of the model, particularly in cases where false positives must be avoided. It indicates the proportion of correct positive predictions among all positive predictions made by a model. Precision is calculated using the following formula:

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{9}
\]
• **Recall:** In the classification process, recall is crucial to evaluate the sensitivity of the model in detecting positive instances. It measures the proportion of correct positive predictions among all truly positive instances. Recall is calculated as follows:

\[
Recall = \frac{TP}{TP + FN}
\]

(10)

• **F1-Score:** This parameter is a combined measure of precision and recall, providing a balanced assessment of a model’s performance in terms of positive and negative predictions. It is particularly useful when classes are unbalanced, offering a holistic view of the quality of the model’s predictions in classification, while considering trade-offs between precision and recall. The equation for the F1-Score is:

\[
F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}
\]

(11)

Where:

- **True Positives (TP):** represents correct predictions of the presence of a plant disease.
- **True Negatives (TN):** represents correct predictions that a plant is healthy.
- **False Positives (FP):** represents cases or predictions that a tomato is classified as diseased (presence of disease) when in reality the plant is healthy.
- **False Negatives (FN):** occurs when the predictions that a tomato is classified as healthy (absence of disease) when in reality the plant is infected with a disease.

4.3 THE PROPOSED METHODOLOGY

In this section, we introduce a description of all the necessary parts to achieve tomato disease detection based on Plant-Village database images presented in subsection 4.1. The proposed methodology consists of two main steps: The first focuses on the architecture to build a CNN deep learning network for tomato disease detection based on 70% of the content of the used database, and the second step focuses on the validation process of the proposed architecture using the remaining 30% of the database. The proposed architecture of the CNN deep learning network is based on 20 layers as below:

- **Image Input:** 256x256x3 images with ’zero center’ normalization.
- **2-D Convolution:** 3x3x3 convolutions with 32 filters and stride [1 1] with padding ’same’ option.
• **Batch Normalization**: Batch normalization with 32 channels.
• **ReLU**: Sets all negative values in the input to zero and leaves positive values unchanged.
• **2-D Max Pooling**: 3x3 max pooling with stride [3 3] and padding [0 0 0 0].
• **2-D Convolution**: 3x3x32 with 32 filters and stride [1 1] with padding 'same' option.
• **Batch Normalization**: Batch normalization with 32 channels.
• **ReLU**: Sets all negative values in the input to zero and leaves positive values unchanged.
• **2-D Max Pooling**: 3x3 max pooling with stride [2 2] and padding 'same'.
• **2-D Convolution**: 3x3x32 with 64 filters and stride [1 1] with padding 'same' option.
• **Batch Normalization**: Batch normalization with 64 channels.
• **ReLU**: Sets all negative values in the input to zero and leaves positive values unchanged.
• **2-D Max Pooling**: 3x3 max pooling with stride [2 2] and padding 'same'.
• **2-D Convolution**: 3x3x64 with 128 filters and stride [1 1] with padding 'same' option.
• **Batch Normalization**: Batch normalization with 128 channels.
• **ReLU**: Sets all negative values in the input to zero and leaves positive values unchanged.
• **2-D Max Pooling**: 3x3 max pooling with stride [2 2] and padding 'same'.
• **Fully Connected**: 10 fully connected layer.
• **Softmax**: Converts the raw output scores of the network into probabilities that sum up to 1, making it easier to interpret the final output as class probabilities.
• **Classification Output**: Cross-entropy with 'Healthy Tomatoes' and 9 other disease classes.

**5 EXPERIMENTAL RESULT AND DISCUSSION**

The proposed CNN architecture for tomato plant disease detection and classification, as presented in subsection 4.3, was trained and validated using 70% of the content of the used database and tested using the remaining 30% of the PlantVillage database. The main metrics evaluation criteria used to ensure correct and precise contemplation are presented in
section 4.2. The obtained results are presented using confusion matrices as shown in figures 2 and 3 for the training and testing processes, respectively.

According to the obtained results presented in figure 2, we note that the proposed CNN architecture leads to identify 1114 cases of healthy tomatoes and 10094 cases of diseased tomatoes distributed according to the 9 diseases classes, where all these predictions are completely correct without any failure for each class. These results show a high level of precision and recall with metrics of 100% according to the right column and the bottom line respectively. In addition, we deduce the accuracy of the proposed architecture by dividing the number of correct predictions, represented by the principal matrices confusion diagonal, by the total number of predictions, this training accuracy achieves a metric of 100%. With these results, we note that the proposed methodology successfully classifies all images for each condition or class of tomato and indicates the perfect degree of learning during the training process using the Plant-Village database.

The testing process is very important to validate the obtained results from the training process in CNN deep learning techniques. The obtained results of the testing process are presented in figure 3.
We note that the model achieves accuracy around 97.5% for the testing process, this accuracy is the sum of correct predictions divided by the total number of predictions. We can also extract other parameters and evaluation metrics from the confusion matrices presented by the figure 3 for a deep analysis and accurate discussions for each disease class, these parameters are presented in table 1.

Table 1: Classification Metrics results for Tomato Diseases using Plant Village database.

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision(%)</th>
<th>Recall(%)</th>
<th>F1-Score(%)</th>
<th>Accuracy(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Healthy_Tomatos</td>
<td>474</td>
<td>2</td>
<td>3</td>
<td>99.6</td>
<td>99.4</td>
<td>99.5</td>
<td>98.5</td>
</tr>
<tr>
<td>Bacterial_spot</td>
<td>631</td>
<td>12</td>
<td>7</td>
<td>98.1</td>
<td>98.9</td>
<td>98.5</td>
<td>99.4</td>
</tr>
<tr>
<td>Early_blight</td>
<td>269</td>
<td>22</td>
<td>31</td>
<td>92.4</td>
<td>89.7</td>
<td>91.0</td>
<td>89.4</td>
</tr>
<tr>
<td>Late_blight</td>
<td>549</td>
<td>27</td>
<td>24</td>
<td>95.3</td>
<td>95.8</td>
<td>95.6</td>
<td>96.0</td>
</tr>
<tr>
<td>Leaf_Mold</td>
<td>279</td>
<td>4</td>
<td>7</td>
<td>98.6</td>
<td>97.6</td>
<td>98.1</td>
<td>97.2</td>
</tr>
<tr>
<td>Septoria_leaf_spot</td>
<td>521</td>
<td>10</td>
<td>10</td>
<td>98.1</td>
<td>98.1</td>
<td>98.1</td>
<td>97.6</td>
</tr>
<tr>
<td>Spider_mites_ Two_spotted_spider_mite</td>
<td>494</td>
<td>17</td>
<td>9</td>
<td>96.7</td>
<td>98.2</td>
<td>97.4</td>
<td>97.8</td>
</tr>
<tr>
<td>Target_Spot</td>
<td>404</td>
<td>13</td>
<td>17</td>
<td>96.9</td>
<td>96.0</td>
<td>96.4</td>
<td>97.4</td>
</tr>
<tr>
<td>YellowLeaf_ Curl_Virus</td>
<td>958</td>
<td>9</td>
<td>4</td>
<td>99.1</td>
<td>99.6</td>
<td>99.3</td>
<td>99.4</td>
</tr>
<tr>
<td>mosaic_virus</td>
<td>106</td>
<td>2</td>
<td>6</td>
<td>98.2</td>
<td>94.6</td>
<td>96.4</td>
<td>98.2</td>
</tr>
<tr>
<td>Mean</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>97.3</td>
<td>96.8</td>
<td>97.0</td>
<td>97.1</td>
</tr>
</tbody>
</table>

Source: the authors 2024

According to the obtained results presented in table 1, we note that the proposed architecture shows an average accuracy around 97.1%, this average accuracy value is very close to the real value of the accuracy of our model, which is 97.5%. This small difference comes back to the variance cases number for each class in the Plant Village database.
Despite this small difference, the proposed methodology shows that the model is very competent in correctly classifying the various tomato diseases across all the instances tested. Furthermore, according to the precision and recall, we notice that the model showed an excellent precision and high recall coefficient with metric values around 99% for some classes such as healthy tomatoes, Bacterial Spot, and Yellow Leaf. This means that the model is very reliable in correctly identifying these tomato diseases or classes with few false predictions. All these good results, such as accuracy, precision, and recall, are confirmed by the high values obtained of the F1-Score in most tomato disease classes (around or more than 97%), indicating a good balance between precision and recall parameters. This particularly is important in agricultural applications where both the reduction of false predictions and the complete identification of positive cases are crucial for this type of classification. Despite the positive results obtained by our CNN architecture for the classification of different types of tomato disease, it is imperative to focus on the weaknesses identified for the Early blight disease compared to other classes with 92.4%, 89.7%, 91.0% and 89.3% for precision, recall, F1-Score, and accuracy respectively. Moreover, a comparison of the proposed methodology of tomato diseases classification with the most recent popular methods in this field of research is important, and the obtained results are presented in the table 1.

Table 2: Comparison of Tomato Disease Classification results of the proposed methodology and the most popular methods.

<table>
<thead>
<tr>
<th>Authors</th>
<th>Methods</th>
<th>Accuracy(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lingwal et al [22]</td>
<td>VGG-16 and CNN</td>
<td>90.0</td>
</tr>
<tr>
<td>Agarwal et al [23]</td>
<td>CNN Network</td>
<td>91.2</td>
</tr>
<tr>
<td>Abass et al [11]</td>
<td>DenseNet, C-GAN</td>
<td>97.1</td>
</tr>
<tr>
<td>Our Method</td>
<td>CNN Network</td>
<td>97.5</td>
</tr>
</tbody>
</table>

Source: the authors 2024

According to the obtained results presented in Table 2, a significant advancement in the field of tomato disease classification through the use of artificial intelligence techniques and methods is observed. We notice that our CNN architecture outperforms the best recent methods, such as those proposed by Lingwal et al[22], Agarwal et al[23], and Abass et al[11]. These recent methods offer an accuracy scores around 90.0%, 91.2%, and 97.1% respectively, our proposed method outclass them with an accuracy score around 97.5% for tomato diseases classification. In addition, both our proposed methodology and Baser et al.’s approach [13] report an identical accuracy of 97.5%. This parallel performance suggests that both methods are equally efficient at leveraging CNN architectures to differentiate among the various tomato diseases and healthy tomato plants. However, it’s
important to highlight that our proposed methodology outclasses the Baser et al.’s approach during the learning step by an accuracy score of 100%, while Baser et al.’s approach achieved an accuracy score of 98.1%. With this option, our approach is found more accurate than Baser et al.’s approach for more cases different from Plant Village database.

6 CONCLUSION

In this work, we have proposed an efficient methodology to achieve a correct and precise tomato diseases detection and classification using a CNN Deep Learning Network. The proposed architecture is based on 20 layers including convolution, max pooling, batch normalization, and the ReLU process. The proposed methodology proved a strong robustness and effectiveness according to the training step results, where all metric criteria scored 100%. These results were validated by the testing process, with scores of 97.3%, 96.8%, 97.0%, and 97.5% for precision, recall, F1-score, and accuracy coefficient criteria, respectively. Furthermore, a comparison with the best existing techniques, which aimed to achieve the same goals in the literature, has proven that our proposed method is more effective and more accurate than the best recent methods. Despite the positive results obtained by our proposed method, it is imperative to highlight an accepted weakness for the Early blight disease compared to other classes.
REFERENCES


