STUDIES PUBLICAÇÕES
Studies in Engineering and Exact Sciences
ISSN: 2764-0981

An intelligent agriculture monitoring framework for leaf disease detection using YOLOv7

Uma estrutura de monitoramento agrícola inteligente para detecção de doenças foliares usando o YOLOv7

Marco de vigilancia agrícola inteligente para la detección de enfermedades foliares mediante YOLOv7

DOI: 10.54021/seesv5n2-516

Originals received: 10/11/2024
Acceptance for publication: 11/01/2024

Thabet Righi
PhD in Computer Science
Institution: LIAP Laboratory, University of El Oued
Address: PO Box 789, El Oued, 39000, Algeria
E-mail: righi-thabet@univ-eloued.dz

Mohammed Charaf Eddine Meftah
PhD in Computer Science
Institution: LIAP Laboratory, University of El Oued
Address: PO Box 789, El Oued, 39000, Algeria
E-mail: mohammed-meftah@univ-eloued.dz

Abdelkader Laouid
PhD in Computer Science
Institution: LIAP Laboratory, University of El Oued
Address: PO Box 789, El Oued, 39000, Algeria
E-mail: abdelkader-laouid@univ-eloued.dz

Mohammed Al-Khalidi
PhD in Computer Science
Institution: Manchester Metropolitan University
Address: Department of Computing and Mathematics, Manchester Metropolitan University, Manchester, M1 5GD, UK
E-mail: m.al-khalidi@mmu.ac.uk

Mostefa Kara
PhD in Computer Science
Institution: King Fahd University of Petroleum and Minerals (KFUPM)
Address: Interdisciplinary Research Center for Intelligent Secure Systems (IRC-ISS), KFUPM, Dhahran, 31261, Saudi Arabia
E-mail: mostefa.kara@kfupm.edu.sa

ABSTRACT

Agriculture is one of the most important economic sectors on which societies have relied since ancient times. With the recent development of technology, agriculture has also been incorporating modern techniques such as the Internet of Things and Artificial Intelligence to improve productivity and monitor the farming process. One of agriculture's most prominent issues is the spread of plant diseases and the lack of real-time monitoring. Various systems and operations have recently been developed to predict and diagnose plant diseases. However, current operations have been selective, focusing on a specific aspect without addressing other important aspects, resulting in either partial or compound application of results, rendering the desired outcomes ineffective. To deal with such challenges, we propose an intelligent framework for real-time agriculture monitoring and disease detection, namely a system for monitoring plant diseases using YOLOv7. In the proposed framework, a rule-based policy has been designed for detecting plant diseases using online plant leaf monitoring, sensors, and surveillance cameras. Images of plant leaves captured by different cameras are sent in real-time to central cloud servers for disease detection. The improved YOLOv7 technology is utilized for plant disease detection, and the proposed system has been evaluated using a dataset of diseased tomato leaves, comparing it with different models based on various performance metrics to demonstrate its effectiveness, achieving an accuracy of 96%.

Keywords: Agriculture. Intelligent Agriculture. Monitoring Framework. YOLOv7. Leaf Disease Detection. Tomato Plant Disease Detection.

RESUMO

A agricultura é um dos setores econômicos mais importantes dos quais as sociedades dependem desde os tempos antigos. Com o recente desenvolvimento da tecnologia, a agricultura também vem incorporando técnicas modernas, como a Internet das Coisas e a Inteligência Artificial, para melhorar a produtividade e monitorar o processo agrícola. Um dos problemas mais proeminentes da agricultura é a disseminação de doenças nas plantas e a falta de monitoramento em tempo real. Recentemente, vários sistemas e operações foram desenvolvidos para prever e diagnosticar doenças de plantas. Entretanto, as operações atuais têm sido seletivas, concentrando-se em um aspecto específico sem abordar outros aspectos importantes, o que resulta na aplicação parcial ou composta dos resultados, tornando ineficazes os resultados desejados. Para lidar com esses desafios, propomos uma estrutura inteligente para o monitoramento agrícola em tempo real e a detecção de doenças, ou seja, um sistema para monitorar doenças de plantas usando o YOLOv7. Na estrutura proposta, uma política baseada em regras foi projetada para detectar doenças de plantas usando monitoramento on-line de folhas de plantas, sensores e câmeras de vigilância. As imagens das folhas das plantas capturadas por diferentes câmeras são enviadas em tempo real para servidores centrais em nuvem para detecção de doenças. A tecnologia YOLOv7 aprimorada é utilizada para a detecção de doenças em plantas, e o sistema proposto foi avaliado usando um conjunto de dados de folhas de tomate doentes, comparando-o com diferentes modelos com base em várias métricas de desempenho para demonstrar sua eficácia, alcançando uma precisão de 96%.

Palavras-chave: Agricultura. Agricultura Inteligente. Estrutura de Monitoramento.

YOLOv7. Detecção de Doenças nas Folhas. Detecção de Doenças em Plantas de Tomate.

RESUMEN

La agricultura es uno de los sectores económicos más importantes en los que se han basado las sociedades desde la antigüedad. Con el reciente desarrollo de la tecnología, la agricultura también ha ido incorporando técnicas modernas como el Internet de las Cosas y la Inteligencia Artificial para mejorar la productividad y monitorizar el proceso de cultivo. Uno de los problemas más destacados de la agricultura es la propagación de enfermedades de las plantas y la falta de supervisión en tiempo real. Recientemente se han desarrollado diversos sistemas y operaciones para predecir y diagnosticar las enfermedades de las plantas. Sin embargo, las operaciones actuales han sido selectivas, centrándose en un aspecto específico sin abordar otros aspectos importantes, lo que ha dado lugar a una aplicación parcial o compuesta de los resultados, haciendo que los resultados deseados sean ineficaces. Para hacer frente a estos retos, proponemos un marco inteligente para la monitorización de la agricultura en tiempo real y la detección de enfermedades, a saber, un sistema para la monitorización de enfermedades de las plantas utilizando YOLOv7. En el marco propuesto, se ha diseñado una política basada en reglas para la detección de enfermedades de las plantas utilizando la monitorización en línea de las hojas de las plantas, sensores y cámaras de vigilancia. Las imágenes de las hojas de las plantas captadas por diferentes cámaras se envían en tiempo real a servidores centrales en la nube para la detección de enfermedades. La tecnología mejorada YOLOv7 se utiliza para la detección de enfermedades de las plantas, y el sistema propuesto se ha evaluado utilizando un conjunto de datos de hojas de tomate enfermas, comparándolo con diferentes modelos basados en diversas métricas de rendimiento para demostrar su eficacia, alcanzando una precisión del 96%.

Palabras clave: Agricultura. Agricultura Inteligente. Marco de Monitorización. YOLOv7. Detección de Enfermedades de la Hoja. Detección de Enfermedades del Tomate.

1 INTRODUCTION

As the backbone of human civilization, agriculture holds immense importance for society, economies, and the environment. It is the primary source of food production, providing the essential sustenance required for the growing global population. It plays a crucial role in ensuring food security by producing diverse crops, fruits, vegetables, and livestock products [8]. Due to the importance of agriculture, it is essential to foster sustainable practices that address the challenges of a growing global population, changing climates, and evolving economic landscapes [14]. Therefore, continued development and innovation in agriculture are vital for ensuring a resilient, productive, and environmentally conscious food system for the future [20]. Digital farming, or precision agriculture, uses advanced technology and digital tools to optimize various aspects of agricultural practices. The goal is to enhance efficiency, productivity, and sustainability in farming operations, increasing crop yields, resource efficiency, cost savings, and reducing environmental impact [31]. By leveraging technology and data-driven insights, farmers can make more informed decisions, adapt to changing conditions, and contribute to sustainable and resilient agriculture practices [28]. As part of this, Artificial Intelligence (AI) has played a crucial role in advancing digital farming by providing sophisticated tools and capabilities to analyze data, make predictions, and optimize various aspects of agricultural operations. AI algorithms analyze data from various sources to monitor crop health, identify diseases, and assess overall plant conditions [30, 22]. AI-powered systems can identify and differentiate between crops, weeds, and pests. This information helps in the targeted application of pesticides and herbicides, detecting signs of pests or diseases early, allowing for timely intervention, and subsequently minimizing environmental impact [10,7]. One of the most important concerns in agriculture is the detection of plant diseases. Early detection of diseases helps prevent their spread among other plants, thus preventing significant economic losses. The consequences of plant diseases can vary depending on the type of pathogen, the affected crop, and the stage at which the disease is identified and managed. Diseases can affect the quality and quantity of harvested crops and the impact can range from mild manifestations to the destruction of entire plantations that severely affect the agricultural economy [1]. Tomatoes are known as one of the most widely consumed fruits globally [18]. They are one of the major crops in agriculture, the second largest crop globally, and a significant source of income for farmers and horticulturists [33, 35]. Tomatoes can grow in various dry soil types [21, 6]. Farmers and horticulturists cultivate tomatoes for cooking or commercial purposes. However, at times, they struggle to achieve suitable progress in plant growth, and tomatoes may not appear at the right harvest time or fully develop on the plant. In some cases, tomato plants are susceptible to various diseases, such as blight, rot, or other fungal and bacterial infections, which can cause black spots or a change in the fruit’s color [21]. One proposed solution to this problem is the implementation of an agriculture monitoring and plant disease detection system, which uses tomatoes as a model to evaluate the efficiency of the proposed system in detecting plant diseases. Deep learning techniques are considered the latest methods for computer vision tasks [2]. These techniques have been applied in various fields, including agriculture. Applying machine learning techniques in agriculture has significantly increased agricultural productivity, especially with recent advancements in deep learning [05, 25]. Among these advancements is the use of Convolutional Neural Networks (CNNs) to diagnose and detect plant diseases [27]. It involves training artificial neural networks to learn and recognize patterns within large datasets, enabling accurate identification of diseases based on visual symptoms [17, 16, 15]. CNNs are trained on extensive datasets containing images of healthy and diseased plants. The models learn to automatically extract relevant features from these images, allowing them to differentiate between healthy and infected plants based on visual cues [12, 11]. In this paper, we primarily focus on using the optimized YOLOv7 algorithm [09] in our proposed system for real-time disease detection in tomato plants based on the approach of deep learning before the spread of the disease in the plant and reducing the risk of plant damage. This paper has two main contributions:

1. We propose a system that detects and monitors plant diseases in realtime in agriculture using deep learning techniques.
2. The second contribution is improving the accuracy of the yolov7 methodfor detecting plants, where these improvements are an addition to the agricultural field in the process of improving the performance of detecting plant diseases, reducing their spread, and improving production yields for farmers.

The rest of the paper is structured as follows. Section 2 presents works similar to this project. Section 3 shows a comparative analysis of the selected Study. Section 4 explains how it works and handles training and test data. Section 5 presents the results obtained. Section 6 concludes the paper.

2 RELATED WORK

Nowadays, one of the crucial tasks in modern agriculture is to monitor and track plant growth in real time to minimize plant damage and improve productivity. Several works and different applications have been developed lately for monitoring and detecting plant diseases. Many research projects have contributed to the development of smart agriculture. Still, these projects focus on partial aspects without addressing other aspects, so the results of these projects become either partial applications or a synthesis of different works and methods that may be inconsistent, which makes the desired results ineffective. Therefore, we proposed in previous work [32] a comprehensive reference approach for an advanced agricultural information system that accommodates the agricultural sector within its interconnected levels, public and private, and its various aspects (Monitor, Prediction, Optimization, Control) within a modern information technology vision based on the Internet of Things (IoT), artificial intelligence and optimization technologies [24]. In this section, we shed some light on the most relevant previous work regarding the application of monitoring and disease detection that affect various types of plants. The results achieved by these applications are as follows: In 2018, researchers proposed developing convolutional neural network models for plant disease detection and diagnosis using simple leaf images of healthy and diseased plants through deep learning methodologies. Konstantinos P. Ferentinos et al performed model training using an open database of 87,848 images containing 25 different plants in a set of 58 distinct classes of ’plant, disease’ groups, including healthy plants. They trained model architectures, and the results yielded a 99.53% success rate in identifying a mixture of plant and disease or a healthy plant [19]. In 2021, Amreen Abbaset al proposed a deep learning-based method for detecting tomato diseases. The approach used a Conditional Generative Adversarial Network (C-GAN) to generate structural images of tomato leaves, followed by training a DenseNet121 model on artificial and real images using transfer learning to classify tomato leaf images into ten different disease classes. The model was extensively trained and tested on the publicly available PlantVillage dataset. The proposed method achieved impressive accuracy rates of 99.51%, 98.65%, and 97.11% for classifying tomato leaf images into 5 classes, 7 classes, and 10 classes, respectively [9]. In 2020, Anjanadevi B et al. developed an optimized and customized Deep Convolutional Neural Network (DCNN) model for plant disease classification. The model was trained using the PlantVillage dataset, primarily using images of tomato, maize, and potato plants for training and testing. The dataset included both healthy and diseased tomato leaves. The experimental results of the proposed model were compared with other architectures, such as MobileNet and DarkNet-19 ResNet-101. The proposed network was also trained on an object classification task for monitoring the working conditions of pin insulators. The overall accuracy achieved in the experiments was 85.3% [04]. In 2021, Abhishek Mohandas et al. proposed a system to detect and identify plant leaf diseases using object detection techniques in image processing. They used the YOLOv4 framework, based on convolutional neural networks, for real-time object detection. Their work focused on different leaf diseases of vegetables and fruits such as tomato, mango, strawberry, bean, and potato, focusing on real-time detection of leaf diseases [03]. In 2021, Midhun P. Mathew et al. utilized the YOLOv5 algorithm to detect bacterial spots in sweet pepper plants using a deep-learning approach. Their proposed method involved collecting a dataset from Kaggle and tagging it using the LabelImg tool. Using this data, they then trained a custom YOLOv5 model and tested it, achieving faster and more accurate results than previous versions of the same algorithm. In 2021, Mahnoor Khalid et al. proposed a method for detecting healthy and unhealthy money plant leaves using deep learning. The method involved creating a dataset of thousands of images of money plant leaves, categorized as healthy or unhealthy. These images were collected in a controlled environment, and a public dataset with accurate dimensions was used. A deep learning model was trained to identify healthy and unhealthy leaves. Subsequently, a YOLOv5 model was trained to detect the smallest disease spot on the exclusive and general datasets. The search conducted using YOLOv5 was able to accurately and quickly identify diseased spots with an accuracy of up to 93% in the test set [29].

3 COMPARATIVE ANALYSIS OF SELECTED STUDY

In this study, we aim to compare and analyze the performance of some existing projects related to detecting and monitoring plant diseases using deep learning. Our analysis will include studies that use Convolutional Neural Networks (CNN), transfer learning, and other deep learning methods. By identifying the strengths and weaknesses of these projects, we hope to provide insights for developing more effective and efficient solutions for detecting and monitoring plant diseases. Table 1 represents a comparative analysis of selected studies.

Table 1: A comparative analysis of selected studies.

Reference	Year	Author	Methodology	Project	Accuracy
[19]	2018	Konstantinos P. Ferentinos	CNN, deep learning	Plant disease detection	99.53%
[01]	2021	Amreen Abbas, Sweta Jain, Mahesh Gour, Swetha Vankudothu	C-GAN, DenseNet121, transfer learning	Tomato disease detection	99.51%, 98.65%, 97.11%
[04]	2020	Anjanadevi B, Charmila I, Akhil NS, Anusha R	DCNN, MobileNet, DarkNet-19, ResNet-101	Plant disease classification	85.3%
[29]	2016	Sharada P. Mohanty, David P. Hughes and Marcel Salathe	CNN, deep learning, GoogLeNet	Image-Based Plant, Disease Detection	99.35%

Source: Authors.

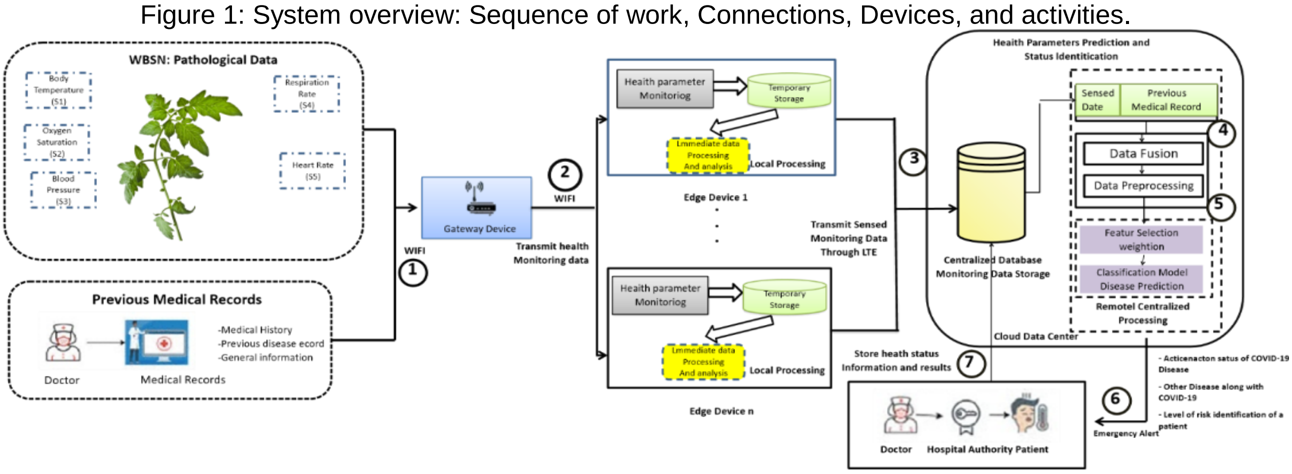
4 METHODOLOGY

As illustrated in Figure 1, the plant monitoring and growth tracking system is a three-stage system consisting of pre-, growth, planting, and postplanting stages. In the pre-planting stage, information is gathered about the plant and soil, and predictions are made about the plant product based on data collected from previous crops. The growth and planting stage involves monitoring the growth and health of plants using sensors and cameras to collect data on factors such as temperature, humidity, wind speed, lighting conditions, and plant leaves. This data is sent to a cloud-based system, which analyzes it using deep learning techniques to provide information about the plant growth status and any diseases present. In

Studies in Engineering and Exact Sciences
ISSN: 2764-0981

the post-planting stage, the system monitors the harvesting process. The second stage of the plant growth monitoring system focuses on monitoring plant growth and health using advanced techniques. This is achieved through the use of a network of sensors and devices that collect data on different parameters, including temperature, humidity, wind speed and direction, and lighting conditions. Additionally, plant leaf images are captured using cameras and processed using deep learning algorithms to detect any signs of disease or abnormalities in the plants.

Figure 1: System overview: Sequence of work, Connections, Devices, and activities.



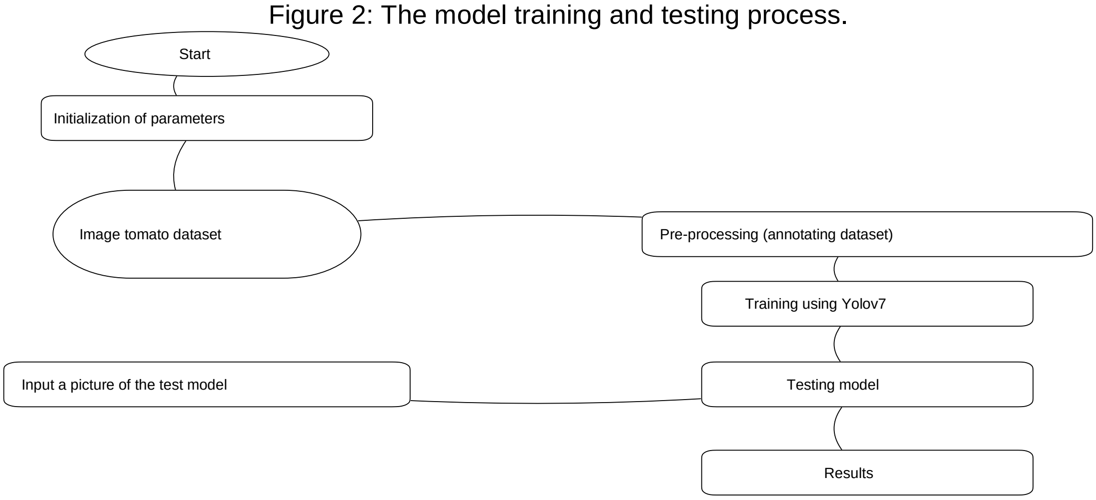
Source:Authors.

4.1 SYSTEM OVERVIEW

The proposed system uses an improved version of the YOLO (You Only Look Once) object detection model, known as YOLOv7. This model has been specifically trained to detect plant diseases by analyzing a large dataset of images of diseased plant leaves. A convolutional neural network (CNN) is used to extract meaningful features from the images and classify them into different disease categories. The image processing pipeline starts by capturing images of plant leaves using cameras placed in strategic locations within the farm. These images are then transferred to the cloud-based monitoring

system, which processes and analyzes the images using the YOLOv7 improved model to detect any signs of disease or abnormalities in the plants. The methodology is presented, and the proposed solution is discussed, including data collection, preprocessing, model selection, training, and evaluation. The proposed work relied on the YOLO (You Only Look Once) model, known as YOLOv7 [09], for object detection, which is based on deep learning approaches and is discussed in detail. In the preprocessing section, new improvements were added to achieve better results. The proposed model was evaluated by applying techniques and comparing results. Figure 2 shows a progressive procedure that was used in the process of detecting and classifying plant diseases. After collecting the data, they were divided into two sections, 80/20 for training and testing, respectively. Then, the optimized YOLOv7 model was trained, and its training plots were obtained to evaluate its suitability.

Figure 2: The model training and testing process.



Source:Authors.

4.2 DATASET PREPARATION AND PREPROCESSING

This research used a public dataset from Kaggle that focused on tomato leaf papers [26]. The images were downloaded from datasets of plant disease detection. The datasets categorized diseased tomato leaves into 2 varieties of plant diseases. Each category contained 1000 images of diseased tomato leaves for training (Table 2). Figure 3 represents images of different types of tomato plant diseases in the datasets.

Table 2: The classification of diseases in the utilized datasets.

class	total
Bacterial-spot	1000
Septoria leaf spot	1000

Source: Authors.

Figure 3: Pictures of different types of tomato plant diseases in the datasets.

(a) Bacterial Spot 1

(b) Bacterial Spot 2

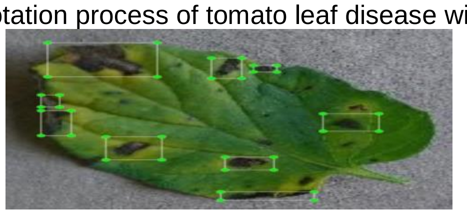
(c) Septoria Leaf Spot 1

(d) Septoria Leaf Spot 2

Source:Authors.

Image annotation is a critical step in accurately identifying leaf diseases. The process involves adding annotations to the images before they are used for training a machine-learning model. The labeling tool was used in this research for image annotation. To begin with, the labeling tool was installed using the pip3 command in Python. Once installed, the tool provided a user-friendly graphical interface to browse the images folder. Then, the annotation method was selected, and YOLO was chosen as the annotation method for this research. Next, the images were loaded into the tool individually, and a bounding box was accurately placed around the plant disease in each image, as shown in (Figure 4). The bounding box helped the machine-learning model identify the disease’s exact location on the leaf. After placing the bounding box, a text comment was entered to describe the specific plant disease within the selected box. This annotation process was repeated for all images in the dataset. Upon completion of the annotation process, a text file was created for each image. The text file contained explanatory data such as the bounding box’s location and size, the plant disease type, and any other relevant information. These annotations were used to train the machine learning model to identify and classify different types of tomato plant diseases accurately.

Figure 4: The annotation process of tomato leaf disease with the labImg tool.

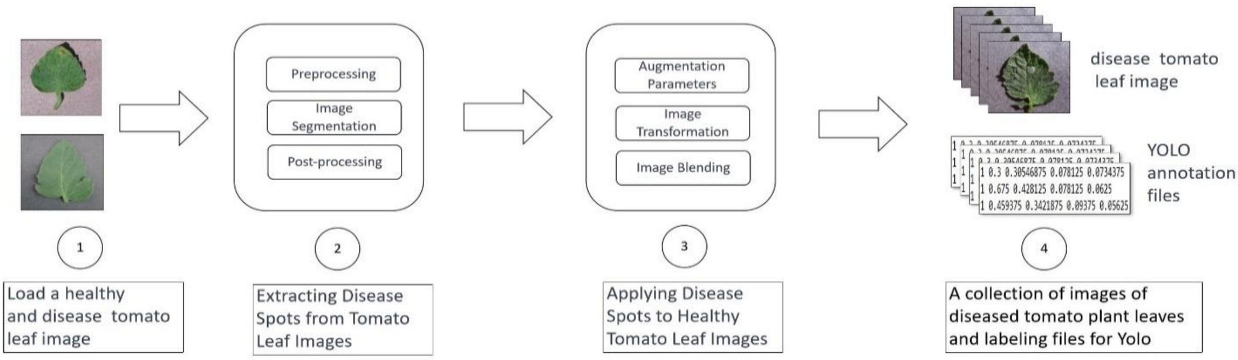


Source:Authors.

4.3 DATA AUGMENTATION FOR ENHANCING DATASET

To achieve better results and improve the training process, data augmentation techniques were employed to generate images of diseased tomato plant leaves based on images of healthy tomato leaves. This increased the number of images in the dataset, thus enhancing the training process. Using image processing operations, a method was proposed to generate images of diseased tomato leaves (Figure 5). The diseased spots were extracted from images of healthy tomato leaves using image processing techniques. Subsequently, an algorithm was developed to blend these disease spots into healthy tomato leaves and create YOLO annotation files to improve image descriptions' accuracy. This means connecting the disease spots with healthy tomato leaves, resulting in promising outcomes in enhancing the training process and increasing accuracy.

Figure 5: Data Augmentation for Tomato Disease Detection: An Overview and Step-byStep Approach.



Source:Authors.

4.3.1 Extracting disease spots from tomato leaf images

At the outset of this procedure, a collection of images portraying tomato leaves afflicted by diseases was curated. These images were sourced from a publicly available dataset on Kaggle. They were meticulously selected and categorized based on the clarity of the diseased regions, thereby enhancing the machine learning model's ability to discern leaf diseases accurately. Subsequently, a series of processing operations were applied to these images to enhance their quality and prepare them for model training. The process of extracting disease spots unfolds in three primary stages (Algorithm 1).

Algorithm 1 - Extracting Disease Spots from Tomato Leaf Images

```
procedure DatasetCompilation
    Let D be the dataset of tomato leaf images affected by diseases.
    D = {d_1, d_2, ..., d_n}, where d_i represents an image.
procedure ImageSelectionAndSorting
    Let C be the clarity measure of a diseased spot in an image.
    C = {c_1, c_2, ..., c_n}, where c_i represents the clarity score of d_i.
    Sort D based on C in descending order.
procedure PreprocessingImages
    for each image d_i in D do
        d_i = resize(d_i, 640, 640)
        d_i = denoise(d_i)
        d_i = color_correction(d_i)
procedure ImageSegmentation
    for each image d_i in D do
        d_i = grayscale_conversion(d_i)
        d_i = thresholding(d_i)
        Resulting binary image: d_i = {0, 1}, where 0 represents background
and 1 represents diseased spots.
procedure NoiseRemovalAndEnhancement
    for each image d_i in D do
        d_i = remove_noise(d_i)
        d_i = morphological_operations(d_i)
        d_i = contour_extraction(d_i)
        d_i = boundary_refinement(d_i)
procedure AccurateDiseaseSpotImages
    The result is a set of images that accurately represent the disease spots:
D' = {d'_1, d'_2, ..., d'_n}, where d'_i represents an image with refined disease
spot boundaries.
```

Source:Authors.

- Pre-processing Infected Leaf Images: In this initial stage, images of diseased tomato leaves are uploaded, and various processing operations are executed. These operations encompass resizing the images to a standardized 640×640-pixel dimension, followed by noise reduction to enhance image clarity and color correction to ensure color uniformity while eliminating distortions (Figure 6).

Figure 6: Picture of diseased tomato leaves after pre-processing.

Source:Authors.

- Image Segmentation Using Thresholding: During this stage, grayscale conversion of the images facilitates a clearer differentiation between diseased spots and the background. The images are subsequently partitioned into distinct regions utilizing thresholding techniques, yielding binary images represented in black and white to isolate the diseased spots effectively (Figure 7).

- Noise Reduction and Image Enhancement: Any remaining noise is eliminated at this juncture, and image quality is further improved using morphological ellipses. Individual disease spot regions are then extracted from segmented images by applying contour detection. Nonspot areas are meticulously filtered out from the spot area to finely tune and sharpen the boundaries of the disease spots, ensuring precise representation. As a result, a set of images is acquired that faithfully portrays the disease spots (Figure 8).

Figure 7: Binary image of diseased tomato leaves after segmentation.

Source:Authors.

Figure 8: A sample extracted from a tomato leaf showing a diseased spot.

Source:Authors.

4.3.2 Applying disease spots to healthy tomato leaf images

After the extraction of diseased spots from tomato leaves in the first stage, these spots are utilized in the second stage, which involves generating images of healthy tomato leaves based on the images of the diseased spots (Algorithm 2). This process includes creating diseased images by conducting a series of sequential operations in three steps. In the first step, images of healthy tomato leaves and diseased spots are uploaded. The intact images are then resized to dimensions of 640×640 pixels. Next, a set of enhancement parameters will be used in the diseased image generation process. These parameters include positioning, measurements, and rotation.

In the second step, these enhancement parameters are applied to each

Studies in Engineering and Exact Sciences
ISSN: 2764-0981

diseased spot with each healthy image. The enhancement parameters are changed randomly with each image. This means that each diseased spot's location, size, and angle are mapped based on these parameters. The modified spots are then combined with the intact images. In the third step, the coordinates of the diseased spots are taken from the image and added to a text file associated with each image. This file contains the coordinates of the diseased spots in the image "YOLO annotation file." Step two and three are repeated for all healthy images, creating a set of diseased images containing the altered spots with YOLO annotation files (.txt). This approach ensures that the coordinates of the diseased spots are aligned with the healthy images, thereby enhancing the accuracy of the training process and the model's capability to effectively identify diseases in tomato leaves.

Algorithm 2 - Generating Images of Healthy Tomato Leaves

```
procedure GenerateHealthyImages
    Let D be the set of diseased spots extracted in the first stage.
    Let H be the set of images of healthy tomato leaves.
    Let E be a set of enhancement parameters.
    E = {e_1, e_2, ..., e_n}, where e_i includes positioning, measurements, and
rotation.
    for each healthy image h_i in H do
        h_i = resize(h_i, 640, 640)
        for each enhancement parameter e_j in E do
            Apply e_j to each diseased spot in D on h_i randomly.
        Save the modified image with diseased spots.
        Create a YOLO annotation file with coordinates of diseased spots.
```

Source:Authors.

4.4 YOLOV7 MODEL OPTIMIZATION

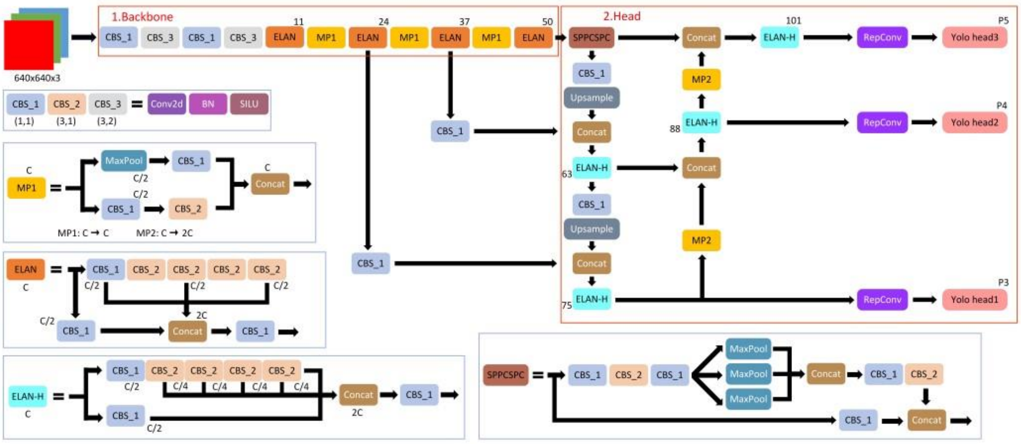
YOLOv7 is the latest addition to the YOLO (You Only Look Once) family of object detection models. As a single-stage object detector, YOLOv7 processes entire image frames in a single forward pass, making it computationally efficient and reducing the need for multiple running operations [30].

The YOLOv7 model uses a backbone network to extract features from input image frames. These features are then combined and mixed in the network neck before being passed along to the head, where YOLOv7 predicts the locations and classes of objects in the image frames. Specifically, YOLOv7 predicts the bounding boxes for each object, along with the confidence score indicating the probability of the object being present in the predicted bounding box [13].

Thanks to its advanced performance in terms of speed and accuracy, YOLOv7 has become a popular choice in many real-world applications for object detection. By leveraging the efficient single-stage structure, YOLOv7 provides a powerful and scalable solution for real-time object detection tasks.

The YOLOv7 model, the newest addition to the YOLO series, boasts unparalleled speed and accuracy in object detection, ranging from 5 fps to 160 fps, with a remarkable accuracy of 56.8% AP, surpassing all other realtime object detectors at 30 fps or higher on the GPU [09]. During the training phase, the YOLOv7 algorithm was chosen for its exceptional accuracy and rapid object detection capabilities. In this paper, experiments are conducted to propose a method for improving the detection efficiency of the YOLOv7 model. The YOLOv7 architecture is shown in Figure 9, and to maintain the integrity of the model's backbone, experiments are carried out on the head and backbone adjustments, taking into account the experimental results and comparisons to obtain the best results.

Figure 9: The structure of YOLOv7



Source : Y. Mu, T.-S. Chen, S. Ninomiya, W. Guo.2020 [35].

The modifications are divided into two steps: part one modifies the convolutional layers, and part two experiments with pooling layers. Firstly, a series of convolutional layers are added to the model's structure. The convolutional layers are used to extract features from the input images and identify objects in the images, which can be expressed as shown in equation 1.

$fconu = Conu(f(i,j,height))(i,j,height \in N)$ (1)

The pooling layers are adjusted after the convolutional layers. The pooling layers are used to reduce the spatial size of the feature maps and make the model more computationally efficient, which can be expressed as shown in equation 2.

$fconu = Conu(fconcat)$ (2)

5 EXPERIMENTS AND RESULTS

5.1 PREPARING PROJECT FOR TRAINING

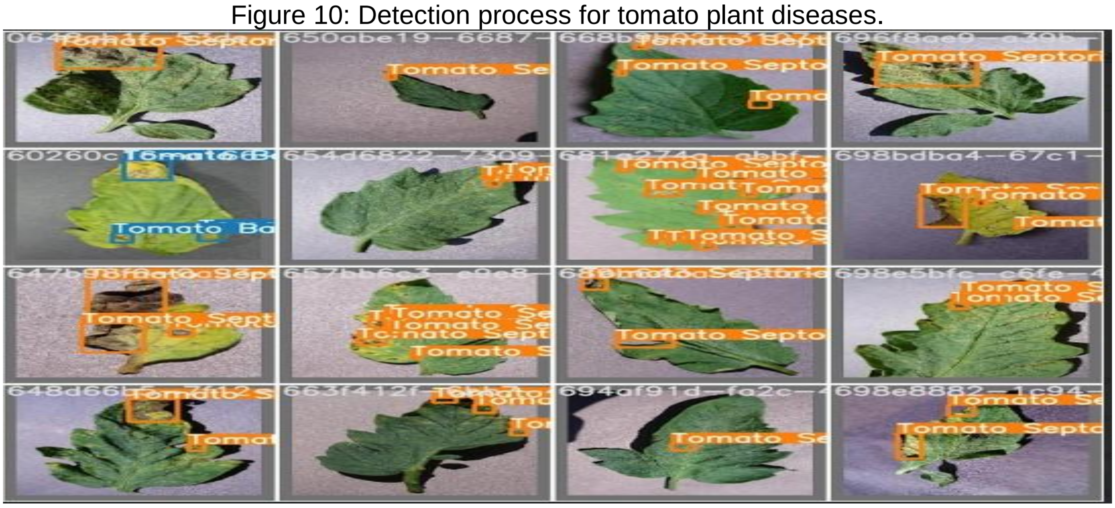
In this study, a custom YOLOv7 model was trained with modifications to its architecture, which consists of two main components: the spine and the head. The backbone, which is a Convolutional Neural Network (CNN) consisting of 38 layers. It consists of a series of convolutional and pooling layers, gradually increasing the depth and width of the layers to allow the network to distinguish increasingly complex features within the input images. Backbone plays a pivotal role in extracting features from input data, which is an essential step in object detection.

The main component, which follows the backbone, consists of a SPPCSPC layer, a series of additional convolutional layers, and a final detection layer. The SPPCSPC layer is a spatial hierarchical pooling layer and a spatial concatenation layer that enables the network to learn features at multiple levels, which is a critical aspect for object detection, as objects of different sizes may appear within the same image. Subsequent convolutional layers improve the features learned by the SPPCSPC layer. The final detection layer is responsible for predicting bounding boxes and object classes within the image, which is a fundamental aspect of object detection. The disclosure process in the YOLOv7 model is an important stage. It includes the final detection layer, which is responsible for predicting bounding boxes and object classes within the input image. This layer uses information learned over the network, including hierarchical spatial features acquired by the SPPCSPC layer and enhanced features from additional convolutional layers. The role of the final detection layer is to precisely locate objects of interest by defining their bounding boxes and assigning corresponding class labels. This step allows the YOLOv7 model to not only detect objects within images but also classify them, making it a powerful and versatile tool for various object recognition and localization tasks

5.2 CUSTOM MODEL TRAINING

In this step, the training process is executed through the command ”python train.py”. The parameter for customizing the training is the device, which is a GPU device used for training, and the ”data/custom.yaml” parameter, which contains the descriptive data for the annotations with the modified enhancements. Then, the custom model parameter is passed inside ”cfg/training/yolov7x-custom.yaml”. Afterward, the yolo7x training file parameter is passed, and the training process begins. After the training is completed, several files are produced containing the results of image testing, graphical representations, and weights/best.pt file inside the ”train/run” folder as shown in Figure 10

Figure 10: Detection process for tomato plant diseases.



Source:Authors.

5.3 RESULTS

In this section, we delve into a detailed analysis of the obtained results and present the observations derived from the previous section of the research. We discuss the training methodology employed for the YOLOv7 object detection model and present the results of the performance analysis. YOLOv7 is a state-of-the-art model known for its speed and accuracy in real-time object detection. The model was trained using a specific dataset, and evaluation metrics were used to assess its performance. We present the findings in Table 3, including accuracy and mean average precision (mAP), which comprehensively evaluate the model's effectiveness.

Table 3: Performance Metrics for YOLOv7 Object Detection Model.

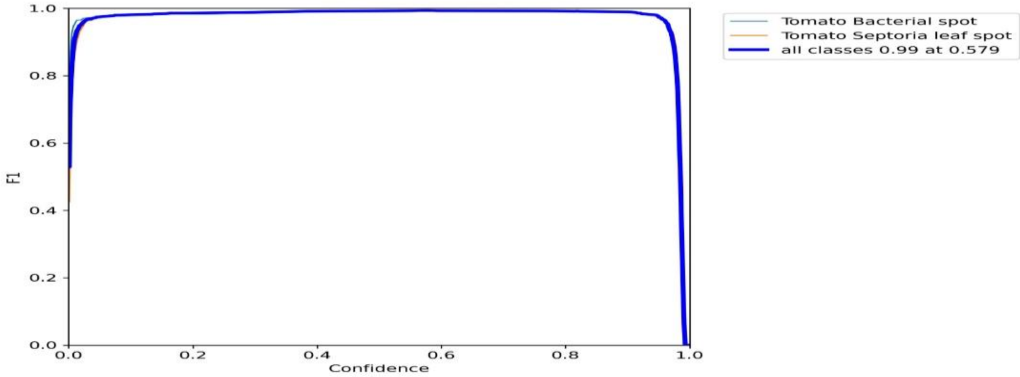
Metric	Accuracy	mAP
Result	96%	99%

Source:Authors.

Table 3 summarizes the performance metrics obtained while evaluating the YOLOv7 model. The mean average precision (mAP) assesses the model's performance across different object classes. These metrics collectively demonstrate the effectiveness of the YOLOv7 model in accurately detecting and classifying objects in real-time scenarios.

In the subsequent sections, we will further analyze the results and discuss their implications in the context of object detection tasks. Additionally, we will explore potential areas for improvement and suggest future research directions to enhance the performance of YOLOv7 in object detection applications. Our analysis comprehensively examined various performance metrics, including mean average precision (mAP), precision, and recall. Figure 11 presents graphs illustrating the F1-confidence graph, displaying the F1 score of the YOLOv7 object detection model at varying confidence thresholds. This metric, adept at striking a balance between precision and recall, proves indispensable in evaluating the overall model performance. We observed a consistent increase in the F1 score of the YOLOv7 model with an elevation in the confidence threshold. This noteworthy trend aligns with expectations, given that a higher confidence threshold indicates that the model's predictions occur only when highly certain.

Figure 11: Graphs presentation with illustrating the F1-confidence graph.



Source:Authors.

Determining the optimal confidence threshold is pivotal and depends on the specific requirements of the application in question. In the provided image, the F1 score for the YOLOv7 model attains a remarkable 0.99 at a confidence threshold of 0.579. This signifies the model's proficiency in correctly identifying 99% of the objects in the dataset with a confidence level of at least 57.9%. In summary, the F1-confidence graph emerges as an invaluable tool for comprehensively evaluating object detection models. Its effectiveness lies in vividly illustrating how the F1 score of a model evolves at different confidence thresholds, thereby facilitating the informed selection of the optimal threshold tailored to a specific application.

$$Precision(P) = TP/(TP + FP) \qquad (3)$$

$$Recall(R) = TP/(TP + FN) \qquad (4)$$

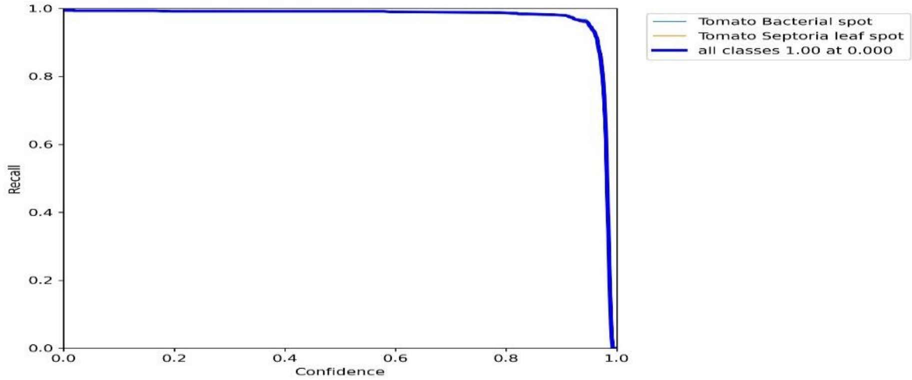
In the context of object detection evaluation, several key terms are used to assess the model's performance. These include True Positive (TP), which represents the number of instances accurately detected by the model. False Positive (FP) refers to the cases where the model incorrectly identifies instances, distinguishing between healthy and unhealthy leaves. On the other hand, False Negative (FN) indicates the number of cases the model did not detect. To evaluate the overlapping accuracy between predicted and ground truth bounding boxes, Intersection over Union (IoU) is calculated. Additionally, a threshold value denoted by 'K' determines the IoU threshold for classifying detections as true positives or false positives.

In Figure 12, we present Confidence according to the R graph, which compares and recalls growth across different experiments based on employed training techniques. This graph serves as an alternative representation of the precision-recall curve, providing insightful observations regarding the model's recall performance.

The Confidence according to the R graph illustrates how the confidence of the custom YOLOv7 model changes with increasing recall. Recall represents the percentage of true positives correctly identified by the model. The graph demonstrates that the confidence of the custom YOLOv7 model increases proportionally with recall growth. This indicates that the model becomes more confident in its predictions as it successfully identifies an increasing number of true positives. Specific results observed for the custom YOLOv7 model in the graph include: At a recall of 0.5, the custom model achieves a confidence of 0.95, signifying 95% confidence in its predictions when detecting 50% of true positives. At a recall of 0.75, the custom model reaches a confidence of 0.98, indicating 98% confidence when identifying 75% of true positives. At a recall of 0.9, the custom model attains a confidence of 0.99, reflecting 99% confidence when detecting 90% of true positives. These specific results in the graph also highlight the reliability of the custom model, achieving confidence levels of 0.95, 0.98, and 0.99 at recall levels of 0.5, 0.75, and 0.9, respectively. These findings indicate that the custom model excels in accurately identifying objects with a high level of confidence. In Figure 13, Confidence according to the P-curve graph is displayed, illustrating the evolution of precision across various experiments grouped based on the utilized training mechanisms. This graph enables us to analyze and understand how different training techniques impact the precision of the custom YOLOv7 model. Precision, in the context of the P-curve, signifies the percentage of expected positives that are true positives.

The specific results observed for the custom YOLOv7 model in the graph are as follows: At a precision of 0.5, the confidence of the custom model is 0.95.

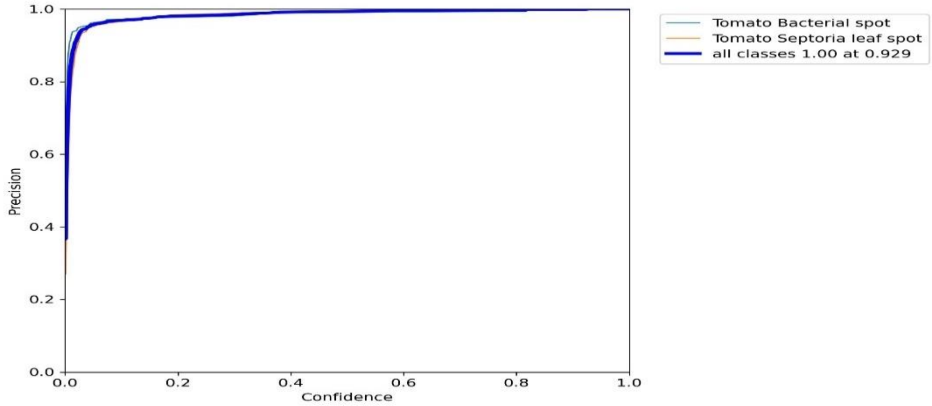
Figure 12: Confidence according to R Graph.



Source:Authors.

This indicates that the model is 95% confident in its predictions when capable of filtering out 50% of false positive results. At a precision of 0.75, the confidence of the custom model is 0.98. This suggests that the model is 98% confident in its predictions when able to filter out 75% of false positive results. At a precision of 0.9, the confidence of the custom model is 0.99. This means that the model is 99% confident in its predictions when able to filter out 90% of false positive results. The specific results in the graph demonstrate that the custom model can achieve confidence levels of 0.95, 0.98, and 0.99 at precision levels of 0.5, 0.75, and 0.9, respectively. This underscores the high reliability of the custom model in accurately detecting objects. Figure 14 presents the confusion matrix, illustrating disparities between actual and predicted values, with a specific focus on tomato bacterial spot and tomato septoria leaf spot. This matrix provides valuable insights into the model's accuracy in classifying these specific categories, highlighting instances of correct predictions (true positives) and misclassifications (False Positives and False Negatives). Tailored for tomato bacterial spot, tomato septoria leaf spot, False Positives (FP), and False Negatives (FN), the matrix's diagonal values indicate the accuracy for each class: tomato bacterial spot (0.78), tomato septoria leaf spot (1.00), and FN background (0.31). While the model excels in accurately classifying tomato septoria leaf spot.

Figure 13: Confidence according to R Graph.



Source:Authors.

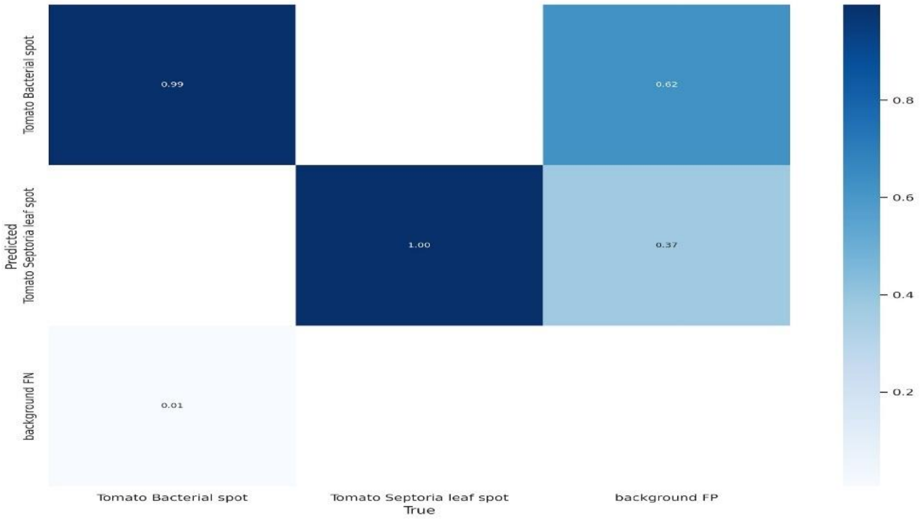
With 1.00 accuracy, it faces challenges in tomato bacterial spot (0.78) and FN backgrounds (0.31). Potential contributing factors include the quality and diversity of training data, model complexity, and other considerations. These findings highlight the model's strength in identifying tomato septoria leaf spot and suggest opportunities for improvement in accurately classifying tomato bacterial spot and FN background. Further exploration and potential adjustments to training data and model complexity could enhance its performance in these specific categories.

6 CONCLUSION

Nowadays, agriculture is considered one of the most important economic sectors that societies have relied on since ancient times. With recent technological developments, agriculture has witnessed the integration of modern technologies, such as the Internet of Things and Artificial Intelligence, to improve productivity and monitor agricultural operations.

In this context, the Smart Farm System was proposed, which provides a comprehensive solution to help farmers and agricultural sector supervisors monitor and improve agricultural activities and farm development. The system relies on Internet of Things technology, sensors, and web technologies.

Figure 14: Confusion matrix displaying the variations between the actual and predicted value.



Source:Authors.

To develop a responsive application compatible with all devices. This application allows farmers to control and monitor remotely, making it easier for them to monitor the farm and production efficiently.

The main contribution of this research includes the design and development of a comprehensive agricultural information system in the proposed framework. A rule-based policy is designed for real-time detection of plant diseases using real-time plant leaf monitoring via local edge devices. Images of plant leaves are sent to central cloud servers in real-time for disease detection using enhanced YOLOv7 technology. The proposed system was evaluated using a dataset of infected tomato leaves, where the accuracy reached 96%. The algorithm performs well, encouraging deeper exploration and development in applying real-time detection and monitoring using YOLOv7 during the cultivation phase. Expanding the model to include other types of plant diseases in future research and expanding the scope of applications using blockchain technologies for health purposes [8], this system is an important step towards improving the agriculture sector using modern technology. It can contribute to improving the agricultural economy in the future.

REFERENCES

[1] A. Abbas, S. Jain, M. Gour, S. Vankudothu, Tomato plant disease detection using transfer learning with c-gan synthetic images, Computers and Electronics in Agriculture 187 (2021) 106279.

[2] A. Casado-Garc´ıa, A. del Canto, A. Sanz-Saez, U. P´erez-Lo´pez, A. Bilbao-Kareaga, F. B. Fritschi, J. Miranda-Apodaca, A. Mun˜ozRueda, A. Sillero-Mart´ınez, A. Yoldi-Achalandabaso, et al., Labelstoma: A tool for stomata detection based on the yolo algorithm, Computers and Electronics in Agriculture 178 (2020) 105751.

[3] A. Mohandas, M. Anjali, U. R. Varma, Real-time detectionand identification of plant leaf diseases using yolov4-tiny, in: 2021 12th international conference on computing communication and networking technologies (ICCCNT), IEEE, 2021, pp. 1–5.

[4] B. Anjanadevi, I. Charmila, A. Ns, R. Anusha, An improved deep learning model for plant disease detection, Int. J. Recent Technol. Eng.(IJRTE) 8 (6) (2020) 5389–5392.

[5] B. N. Naik, R. Malmathanraj, P. Palanisamy, Detection and classification of chilli leaf disease using a squeeze-and-excitation-based cnn model, Ecological Informatics 69 (2022) 101663.

[6] B. Pico, M. J. D´ıez, F. Nuez, Viral diseases causing the greatest economic losses to the tomato crop. ii. the tomato yellow leaf curl virus—a review, Scientia Horticulturae 67 (3-4) (1996) 151–196.

[7] C. Berdjouh, M. C. E. Meftah, A. Laouid, M. Hammoudeh and A. Kumar, "Pelican Gorilla Troop Optimization Based on Deep Feed Forward Neural Network for Human Activity Abnormality Detection in Smart Spaces," in IEEE Internet of Things Journal, vol. 10, no. 21, pp. 18495-18504, 1 Nov.1, 2023, https://doi: 10.1109/JIOT.2023.3271831

[8] C. R. Kumar, C. Nayak, A. K. Pradhan, Status and determinants of crop diversification: evidence from indian states, Letters in Spatial and Resource Sciences 17 (1) (2024) 1.

[9] C.-Y. Wang, A. Bochkovskiy, H.-Y. M. Liao, Yolov7: Trainable bag-offreebies sets new state-of-the-art for real-time object detectors, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 7464–7475.

[10] F. H. Juwono, W. Wong, S. Verma, N. Shekhawat, B. A. Lease, C. Apriono, Machine learning for weed–plant discrimination in agriculture 5.0: An in-depth review, Artificial Intelligence in Agriculture (2023).

[11] F. Tabassum, I. Ahmed, M. Hasan, A. Mahmud, A. Ahnaf, Revolutionizing plant disease detection with cnn and deep learning, in: Asia Simulation Conference, Springer, 2023, pp. 415–425.

[12] G. B. Rozendo, G. F. Roberto, M. Z. do Nascimento, L. Alves Neves, A. Lumini, Weeds classification with deep learning: An investigation using cnn, vision transformers, pyramid vision transformers, and ensemble strategy, in: Iberoamerican Congress on Pattern Recognition, Springer, 2023, pp. 229–243.

[13] H. Li, Y. Deng, Y. Zhou, Z. Li, P. Chen, Z. Tang, Elod: An efficient and lightweight object detection approach for gastric precancerous disease, Authorea Preprints (2023).

[14] I. Idawati, N. Sasongko, A. Santoso, M. Septiani, T. Handayani, A. Sakti, B. Purnamasari, Cocoa farmers' characteristics on climate variability and its effects on climate change adaptation strategy, Global Journal of Environmental Science and Management 10 (1) (2024) 337– 354.

[15] I.Kadi, M.Abbas, M.Miloudi, M. C. E.Meftah,. A CNN Model for Early Leukemia Diagnosis. International Journal of Organizational and Collective Intelligence (IJOCI), 12(1), 1-20. (2022), https://doi.org/10.4018/IJOCI.304889.

[16] I.Kadi, M.Abbas, M. C. E.Meftah: Intelligent models for early Autism detection from MRI images, Studies in Engineering and Exact Sciences, Curitiba, v.5, n.2, 2024. https://DOI: 10.54021/seesv5n2-273.

[17] I. Pacal, Enhancing crop productivity and sustainability through disease identification in maize leaves: Exploiting a large dataset with an advanced vision transformer model, Expert Systems with Applications 238 (2024) 122099.

[18] J. Gao, F. Zhang, J. Zhang, T. Yuan, J. Yin, H. Guo, C. Yang, Development and evaluation of a pneumatic finger-like end-effector for cherry tomato harvesting robot in greenhouse, Computers and Electronics in Agriculture 197 (2022) 106879.

[19] K. P. Ferentinos, Deep learning models for plant disease detection and diagnosis, Computers and electronics in agriculture 145 (2018) 311–318.

[20] K. Verma, P. Sharma, D. Bhardwaj, R. Kumar, N. M. Kumar, A. K. Singh, Land and environmental management through agriculture, forestry and other land use (afolu) system, Land and Environmental Management through Forestry (2023) 247–271.

[21] M. Agarwal, A. Singh, S. Arjaria, A. Sinha, S. Gupta, Toled: Tomato leaf disease detection using convolution neural network, Procedia Computer Science 167 (2020) 293–301.

[22] M. A. Alloghani, Ai for sustainable agriculture: A systematic review, Artificial Intelligence and Sustainability (2023) 53–64.

[23] M. C. E.Meftah, C.A. Cherif, B,Meraghni & C.Berdjouh, Hybrid Architecture Based on Blockchain Technology and Smart Contracts for Healthcare Applications. International Journal of Organizational and Collective Intelligence (IJOCI), 12(1), 1-20. (2022). https://doi.org/10.4018/IJOCI.313600.

[24] M. Kara, A. Laouid, M. Hammoudeh, K. Karampidis, G. Papadourakis, A. Bounceur, A secure multi-agent-based decision model using a consensus mechanism for intelligent manufacturing tasks, Engineering Proceedings 56 (1) (2023). doi:10.3390/ASEC2023-15929. URL https://www.mdpi.com/2673-4591/56/1/234

[25] M. Kara, A. Laouid, A. Bounceur, O. Aldabbas, Arabic opinion mining using machine learning techniques: Algerian dialect as a case of study, in: Proceedings of the 6th International Conference on Future Networks & Distributed Systems, 2022, pp. 87–92.

[26] M. Khalid, M. S. Sarfraz, U. Iqbal, M. U. Aftab, G. Niedba la, H. T. Rauf, Real-time plant health detection using deep convolutional neural networks, Agriculture 13 (2) (2023) 510.

[27] N. Rohilla, M. Rai, A. Dhull, Exploring omfa-cnn for potato leaf disease identification: An assessment against existing models, International Journal of Intelligent Systems and Applications in Engineering 12 (1s) (2024) 209–221.

[28] P. Thongnim, V. Yuvanatemiya, P. Srinil, Smart agriculture: Transforming agriculture with technology, in: Asia Simulation Conference, Springer, 2023, pp. 362–376.

[29] S. P. Mohanty, D. P. Hughes, M. Salath´e, Using deep learning for imagebased plant disease detection, Frontiers in plant science 7 (2016) 1419.

[30] R. Esmaeily, M. A. Razavi, S. H. Razavi, A step forward in food science, technology and industry using artificial intelligence, Trends in Food Science & Technology (2023) 104286.

[31] R. Sharma, S. Wahbeh, B. Sundarakani, I. Manikas, M. Pachayappan, Enhancing domestic food supply in the uae: A framework for technology-driven urban farming systems, Journal of Cleaner Production 434 (2024) 139823.

[32] T. Righi, M. C. E. Meftah, Towards an organizational approach for an advanced agricultural information system (aais), International Journal of Organizational and Collective Intelligence (IJOCI) 12 (1) (2022) 1–20. DOI: 10.4018/IJOCI.313597

[33] T. Zheng, M. Jiang, Y. Li, M. Feng, Research on tomato detection in natural environment based on rc-yolov4, Computers and Electronics in Agriculture 198 (2022) 107029.

[34] W. Zhu, Q. Wang, L. Luo, Y. Zhang, Q. Lu, W.-C. Yeh, J. Liang, Cpam: Cross patch attention module for complex texture tile block defect detection, Applied Sciences 12 (23) (2022) 11959.

[35] Y. Mu, T.-S. Chen, S. Ninomiya, W. Guo, Intact detection of highly occluded immature tomatoes on plants using deep learning techniques, Sensors 20 (10) (2020) 2984.