*Plant Bulletin* Vol 2, (2) 2023: 93-103 ISSN: 2959-3379 (Print), 2959-3387 (Online)

Research Article

OPEN ACCESS Check for updates

# Efficient Diagnosis of Bacterial Leaf Spot in Tomato Plants using Deep Learning CNN Models

**Zohaib Ahmad[1], Mohammed Abdulaziz Alfehaid[2], Saira Asghar[3]*, Abdul Manan[1], Sidra Gill[4], Memoona Bibi[4]**

[1] *Department of Plant Pathology, University of Agriculture Faisalabad, Pakistan.*
[2] *Department of Biology, Imam Muhammad ibn Saud Islamic University, Saudi Arabia.*
[3] *Department of Computer Science, The Islamia University of Bahawalpur.*
[4] *Department of Botany, Faculty of Chemical and Biological Sciences, The Islamia University of Bahawalpur.*
**Corresponding Author's:* sairaasghar05@gmail.com

## ABSTRACT

Tomato is significant vegetable that provides a lot of nutrients for a low price as compared to other vegetables. Fruit is vulnerable to attack by various diseases especially bacterial spot of tomato. The disease is occurred due to the attack of numerous species of genus Xanthomonas that are responsible of the disease. Disease manifested itself in the form of spot-on leaves, stems and fruit. Symptom appears in the form of a small circular to irregular greasy spots on the leaf. Disease reduced the quality as well as the quantity of the yield. Latest field surveys showed that *Xanthomonas campestris* pv. *vesicatoria* (Xcv) is the cause of bacterial spot that has decimated commercial tomato production worldwide. The purpose of this study is to more quickly and accurately diagnose bacterial leaf spot problems in tomato plants. To do this, a deep learning-based Convolutional Neural Network (CNN) named Xantho_Net has been proposed to classify the bacterial and non-bacterial tomato leaves. Various state-of-art pre-trained models i.e., VGG16, MobileNet, DenseNet_201, Inception-ResNet_v2 were selected for comparison purposes. Experiments demonstrated that the suggested CNN model outperforms other models with 99% accuracy.

**Keywords**: Bacterial Spot, CNN model, Xantho.Net, Tomato disease

## INTRODUCTION

Tomato (*Lycopersicon esculentum* L.) is one of the most extensively produced and economically significant vegetables. China, India, the United States, and Turkey are the top producers of tomatoes globally, accounting for 70% of all product [1]. Pakistan is ranked 36th in terms of production and 52nd in terms of exports. According to [2], Pakistan contributes 1.3% of the total global tomato-cultivating area, 0.33% of tomato production, and 0.06% of tomato exports. The average yield of this crop is relatively less in Pakistan when compared to the contemporary world, and this disparity may be attributed to a variety of biotic and abiotic factors, as well as inadequate production technology and inefficient resource use[3]. Bacterial spot (BS) disease is one of these serious hazards to tomato production, affecting both fresh and processed tomatoes. It is caused by the bacterium *Xanthomonas vesicatoria*, which poses a persistent threat to the commercial production of tomatoes all over the world [4,5]The disease-causing bacterium is motile, rod-shaped, and gram-negative, strictly aerobic, and measuring 0.7 to 1.0 m by 2.0–2.4 m in diameter. Additional features include making the extracellular polysaccharide (EPS) xanthan and the yellow, water-insoluble pigment xanthomonadin[6,7]. European Plant Protection Organization classified the bacterium as a quarantine pathogen because of its rapid spread and destructive characteristics. Disease growth is facilitated by hot, humid weather and thick crop patterns [8]. Pathogens can endure in agricultural residue for some time spent in tropical and subtropical areas [9]. The pathogen causes tiny, water-soaked lesions on tomato plants after entering through stomata or minor wounds. Later on, these lesions became necrotic [10]. The disease affects the fruits, flowers, stems, and leaves of tomatoes. Initial leaf symptoms include small, atypical, oily, black blemishes. Photosynthesis is hampered by the shrinkage of the green leaf area. Multiple lesions may cause leaves to turn yellow and drop off. Flowers with greasy blemishes have severe blossom drops. Green

**Article History**
Received: September 12, 2023, Accepted: December 11, 2023, Published: December 30, 2023.

Copyright: © 2022 by the authors. Licensee Roots Press, Islamabad Pakistan.
This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

fruits start to develop little, water-soaked lesions that quickly grows. Eventually, the spots turn dark brown, slightly sunken, and scabby on the surface[11]. In addition, the pathogen's pathovars has already been identified as most significant tomato disease, with losses ranging from 10% to 60% [12]. Additionally, when significant foliar infection developed early in the plant's growth, fruit crop losses of up to 23–44% were seen[13]. Small-scale local growers have also suffered more losses as a result of the qualitative loss brought on by blemishes and spots on the skin of fresh-market tomatoes, particularly when the crop is cultivated in hot and humid environments [14]. Tomato fruit can lose up to 95% of its value in the marketplace [15,16].

Early disease identification and detecting and separating exactly what component triggered the disease are crucial for preventing crop disasters. Early diagnosis is critical for tomato growers to avoid significant economic losses. A wide range of approaches are utilized in tomato disease diagnosis. One such method that is based on an artificial neural network is deep learning, which uses multiple layers of neurons to extract characteristics from unprocessed data. Its goal is to imitate how human brain functions. While deep learning—also referred to as hierarchical learning or deep neural networks—was originally introduced as a novel field of study in machine learning in 2006, its origins can be found in the 1940s. [17].

Deep learning diagnosis of plant diseases is a preferred diagnostic tool among several scientists. By comparing this dataset to images from the actual world, a greater disease detection rate may be produced. Using hyperspectral/multispectral imaging methods in deep learning investigations is another way to improve disease diagnostic achievement. These technologies, in particular, make it feasible to diagnose illnesses in their early stages. Early disease identification implies less pesticide use, more economic benefit, and less environmental harm. Recent years have seen the development of numerous models that use deep learning to identify plant pathogens [18].

Consumed the dataset containing of 87,848 photos acquired outside and inside the lab in a variety of settings. 25 plant species and 58 illnesses were examined in the data[18]. The VGG 16 model architecture has the highest classification success rate (99.53%), according to the trials. In a subsequent study [19], the VGG19 model had the highest classification accuracy (97.86%). The development of appropriate automation systems and real-time plant disease diagnosis have advanced significantly in recent years.

Deep learning architectures that are now in use have been effectively taught to identify illnesses. Since high accuracy must be applied to the output, which may alter when applied to real-world objects (PC, tablet, mobile phone, multiple camera systems, etc.), maintaining generalizability is difficult. Creating a simpler Convolutional Neural Network (CNN) to identify tomato plant illness is the aim of this project. For this objective, a deep learning-based Convolutional Neural Network (CNN) named Xantho_Net has been proposed to classify bacterial and non-bacterial tomato leafs. The Xantho_Net results were also compared with state-of-art pre-trained models i.e., (Visual geometry group) VGG16, MobileNet, DenseNet_201, Inception-ResNet_v2.

## MATERIALS AND METHODS

### Data set

Lately, it has become possible to obtain a large number of data sets that were developed by experts. This research uses the popular tomato leafs dataset from Kaggle. Total of 2,000 images, including both infected (with bacterial spot) and healthy leaves (Fig 1), has been selected for analysis from the tomato leaf dataset.(https://www.kaggle.com/datasets/ashishmotwani/tomato).

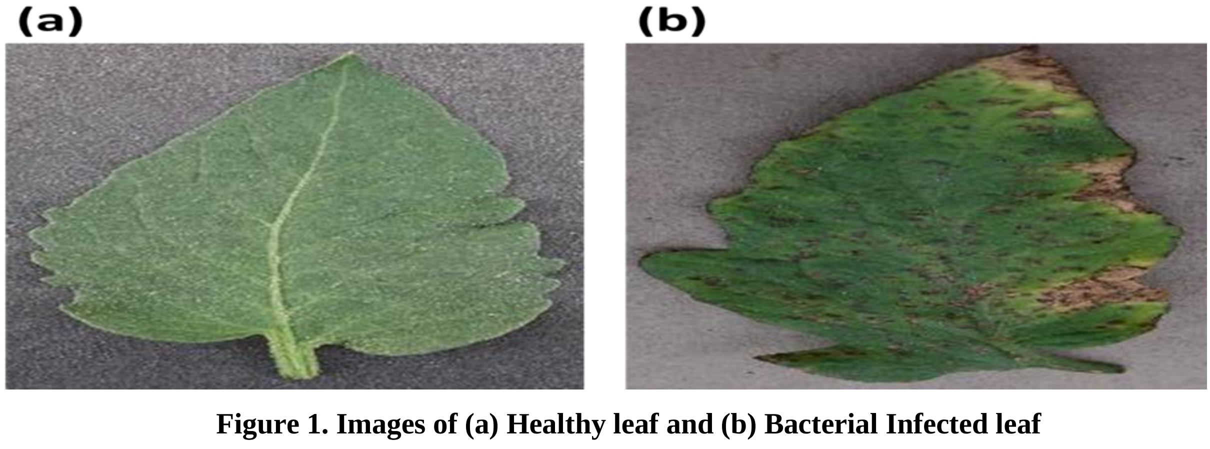



**Figure 1. Images of (a) Healthy leaf and (b) Bacterial Infected leaf**

**Convolutional Neural Network (Cnn) Base Model**

Over the past few years, deep learning—a subset of machine learning—has been applied in many other industries. The focus of modern engineering applications is on resolving difficult object recognition challenges [20]. CNN is the most popular deep learning algorithm used for object recognition applications. CNN is widely acknowledged as the underlying architecture of the deep learning idea. CNN's architecture consists of the convolutional, pooling, fully connected, dropout, and classification layers. Generally speaking, the convolution layer handles the filtering, size reduction, and reduction procedure. For the current item, a smaller-sized filter is chosen, which is then subjected to several processes to complete the filtering process [21]. This procedure is repeated until the system has been trained. The pooling layer is another dimension-reducing layer, similar to the convolutional layer.

This puts more attention on the salient characteristics. Maximum and average pooling are the two common pooling techniques used in CNN models [22].

Usually found near the end of a CNN's architecture, fully connected layers help raise classification results. Dropout eliminates some network connections, which improves learning performance. The matrix-formatted visual data that passed via the convolution and pooling layers is converted into a vector by the fully connected layer.

The CNN training process begins once the input data is received, and at the conclusion of the training phase, the final output is compared to the correct result [23]. In this study, VGG16, MobileNet, DenseNet, Inception-ResNet, and a suggested CNN model were used to identify tomato plant disease.

**VGG16**

Prior to Alex Net, (Visual geometry group) VGG16 focused on lower steps and sizes in the first convolution layer. Despite outperforming AlexNet in terms of accuracy, the VGG16 deep learning architecture uses a lot of memory because of its large number of parameters. In contrast, compared to AlexNet, smaller filters were used [7]. In all convolution layers, the fixed 3x3 dimensional filters in this architecture have configurable numbers of 64, 128 and 256 filters. In 2014, the ImageNet Large Scale Visual Recognition Challenge was won 89% of the time by VGG16, a simple CNN model that supported GPUs. The layers that come after the convolution layers that cooperate are the double or triple ones. The image's size for the input layer are 224 × 224 x 3 pixels. Final layer remains the categorization layer. There are 16 layers in this model: Pooling, Relu, Dropout, Softmax, three completely connected layers, and thirteen convolution layers [24].

**MobileNet**

MobileNet was created to enable vision applications to operate on mobile and embedded systems. Depth-wise separable convolutional layers serve as the foundation for the approach. Layer-dividing 1 x 1 convolutional layers are referred to as point convolutions. In a typical convolution, Filtering the inputs and combining them to produce a new output is one phase. However, the two sub-layers known as point convolution and in-depth convolution make up the deeply separable convolutional layer. For both layers, batch norm and ReLU nonlinearities were applied. Batch norm and ReLU followed the 3x3 convolution layer, with the exception of the final fully linked layer.[25]

**Dense Net**

Densely Connected Convolutional Networks (DenseNet) employ forward connections to connect each layer to the others, according to Zhang et al.[26]. Each layer in the DenseNet design uses the attributes of all levels that come before it as input and feeds the following layer its own layer properties. DenseNet topologies reuse features and propagate to minimize parameters. One advantage of DenseNet over ResNet is that its model parameters are fewer [27]. Its architecture includes bottleneck layers, compression, dense blocks, composite functions, and pooling layers. This research used the DenseNet201 architecture.

**Inception-ResNet**

Inception-ResNet was constructed by parallelizing 1×1, 3×3, 5×5 filters, and 3×3 maximum joint operation in Inception module convolution layers. Parallel processing results in increases in output size and complexity [18]. To address this problem, size reduction is achieved by adding 1x1 convolution layers before parallel Naive Inception convolution layers. Focusing on attributes is the technique of pooling features. This appears to be a reduction of the height and width dimensions in a normal maximum joint procedure.Furthermore, the Rectified Linear Unit (ReLU) activation function is then applied to the 1x1 convolutional layers. The first convolutional process's size was changed to 1x1 in order to optimize the residual layer. Even in situations where learning ends at the convolution outputs, the ReLU architecture requires computing a direct sum between the output of the activation function and the preceding activation value.

**Specially designed CNN Architecture Xantho_Net**

A CNN-based model with (fewer layers and parameters) called Xantho_Net has been created specifically for the classification of healthy and bacterial spots in tomato leaf images. CNN works like the visual system of humans and is built on the assumption that raw data consists of two-dimensional images, which makes it possible to encode certain properties. The proposed CNN-based architecture generates feature maps by convolving an image with kernels [29]. In a feature map, the weights of kernels are used to establish relationships between units and the layer below. During training, kernels' weights are adjusted by back propagation. The proposed Xantho_Net layer-by-layer design is shown in Figure 2, and the entire model is explained in Table. 1. The proposed model's layers are described below.

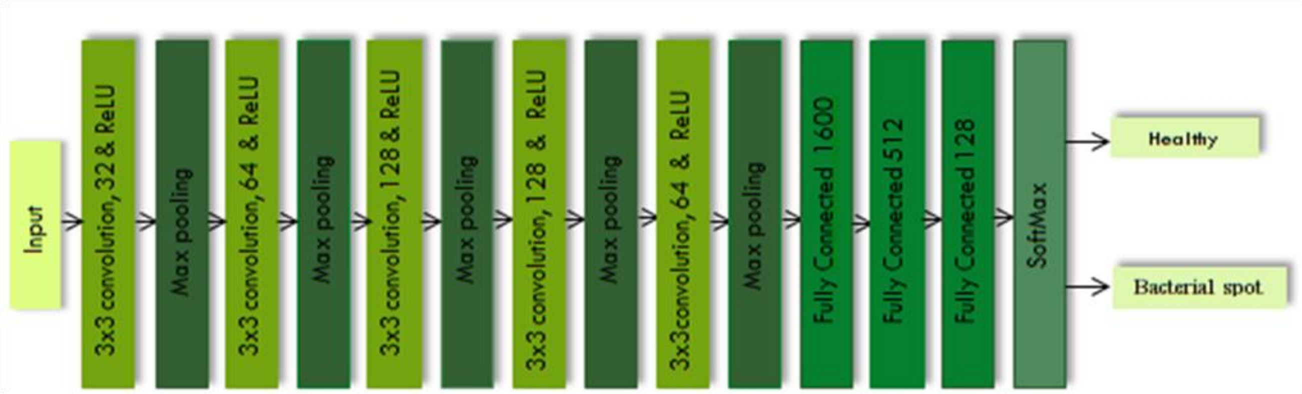



**Figure 2. Architectural diagram of proposed Xantho_Net CNN model**

**Table 1. The detail of Xantho_Net layers**

| Sr. | Layer Type | Output | Shape | Parameters |
|---|---|---|---|---|
| 1. | Conv2D | None | 222,222,32 | 896 |
| 2. | Max_pooling2D | None | 111,111,32 | 0 |
| 3. | Conv2D | None | 109,109,64 | 18496 |
| 4. | Max_pooling2D | None | 54,54,64 | 0 |
| 5. | Conv2D | None | 52,52,128 | 73856 |
| 6. | Max_pooling2D | None | 26,26,128 | 0 |
| 7. | Conv2D | None | 24,24,128 | 147584 |
| 8. | Max_pooling2D | None | 12,12,128 | 0 |
| 9. | Conv2D | None | 10,10,64 | 73792 |
| 10. | Max_pooling2D | None | 5,5,64 | 0 |
| 11. | Flatten | None | 1600 | 0 |
| 12. | Dense | None | 512 | 819712 |
| 13. | Dense | None | 128 | 65664 |
| 14. | Dense | None | 2 | 258 |
| | Total parameters | - | - | 1,200,258 |

**Convolutional Layer**

The picture features that were acquired from the input image collection are represented by feature maps produced by the convolutional layer [30]. Trainable weights are helpful when constructing feature maps with filters. Equation (1) represents an image Img of sizes M and N and a filter F of sizes p and q. A convolutional process is used to build feature maps. This process starts in the top left corner of the input image and ends in the bottom right corner.

$$conv = (Img * F)(x, y) = \sum_M \sum_N I(x - p, y - q)\, F(p,q) \quad (1)$$

**Pooling Layer**

Pooling is a simple and effective technique commonly used after the convolution processes. Pooling results in the generation of feature maps that contain local perceptual fields [30]. Critical contributing features were extracted using the pooling layer. In addition to that, it was used to reduce the dimensionality of the features that were extracted.

**Rectified Linear Unit (Relu) layers**

This layer is necessary for convolutional neural networks to successfully tackle complex problems and functions [12]. An absence of an activation function during the training of a network may lead to the linearity of the network as a result of matrix multiplication. The commonly utilized activation function, called sigmoid function as demonstrated by Equation (2). A key component of the gradient descent process is the sigmoid function, which is calculated using Equation (3).

$$s(x) = \frac{1}{1 + e - x} \quad (2)$$

$$S(x) = S(x)\big(I - S(x)\big) \quad (3)$$

Gradient descent is not employed in deep architectures because it slows down the learning process gradually. As such, while there are many ways to implement the ReLU, it is often a suitable option for deep systems. Equation (4) is the mathematical expression of the ReLU procedure.

$$ReLU(x) = max(x, 0) \quad (4)$$

**Fully Connected Layer**

According to Ferdouse Ahmed Foysal et al. [31] the fully connected layer typically emerges at the conclusion of convolutional neural network (CNN) models. This adds context to my article. The Xantho_Net model that has been suggested, along with other cutting-edge CNN models including VGG16, MobileNet, DenseNet_201, and Inception-ResNet_v2, have been used for comparative analysis.

**Training -testing data and model evaluation**

The data set was split into three parts: 70% for training, 15% for validation, and 15% for testing. The performance of the outputs was assessed using the accuracy of true positive, false positive, false negative, and true negative. The accuracy, precision, recall, and F1-score—standard assessment metrics—have been used to assess the suggested model. Regarding the model's positive class, true positive forecasts with accuracy. Similarly, true negative accurately predicts the negative class. False positives incorrectly predict the positive class in the model, whereas false negatives predict the negative class.The percentage of true predictions in each sample serves as a proxy for the model's accuracy. The percentage of accurately classified findings is referred to as precision. Recall percentage is a good way to gauge how accurately true positives are identified. The precision and recall values' harmonic average is represented by the F1-score.[13]. The metrics' formulas are shown in Equations (5), (6), (7).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

$$ReCall = \frac{TP}{TP+FN} \text{=TP/TP+FN} \quad (6)$$

$$F1 - Score = \frac{2TP}{2TP + FN + FP} \quad (7)$$

The empirically-tuned hyper-parameters of the Xantho_Net architecture are detailed in Table. 2. There are two classes to classify: healthy and bacterial spots. Anaconda3 was used for developing the Xantho_Net.

**Table 2. Hyper-parameters of the proposed Xantho_Net architecture**

| Parameter | Value |
|---|---|
| **Batch size** | 32 |
| **Epoch** | 50 |
| **Momentum** | 0.9 |
| **Learning rate** | 0.0001 |
| **Metric*** | Categorical cross entropy |
| **Patience*** | 2 |
| **Factor*** | 0.2 |
| **Verbose*** | 2 |
| **Optimization method** | Adam |

**RESULTS**

Different state-of-art CNN models including VGG16, MobileNet, DenseNet_201, and Inception-ResNet_v2, were chosen for the comparative experiments to further examine the Xantho_Net's performance. Table. 3 and Figure. 3 demonstrate that Xhoant_Net perform well as compare to VGG16, MobileNet, DenseNet, and Inception-ResNet_v2. The
results were acquired quickly and were better compared to the results of other CNN models in terms of their efficiency in multi-classification tasks. The Xantho_Net has proven useful for producing precise and rapid results with a small number of input parameters. Proposed Xantho_Net and pre-trained CNN models were trained on 50 epochs.

**Table 3. Xantho_Net comparison with state-of-art CNN models**

| Models | Training Accuracy% | Validation Accuracy% | Testing Accuracy% | Training Loss | Validation Loss |
|---|---|---|---|---|---|
| **VGG16** | 90 | 77 | 78 | 0.4623 | 2.9543 |
| **MobileNet** | 99 | 99 | 98 | 0.1996 | 0.0234 |
| **DenseNet_201** | 99 | 91 | 85 | 0.0101 | 1.6779 |
| **Inception-ResNet_v2** | 99 | 99 | 98 | 0.1036 | 1.5553 |
| **Xantho_Net** | **100** | **99** | **99** | **0.0054** | **0.0045** |

The VGG16 model demonstrates the highest training and testing loss among the performance criteria. DenseNet_201 architecture, on the other hand, has the lowest training and testing loss after Xantho_Net. The proposed Xantho_Net model showed the fewest losses. Xantho_Net, MobileNet, and Inception-ResNet_v2 presented high levels of accuracy. Figure.3 shows the training, loss graphs of the proposed validation accuracy, and pre-trained CNN models. Each model was trained on distinct epochs in order to achieve the highest accuracy. For each model, 50 epochs were selected from the experiments. Because the accuracy of each model consistently increased over a period of 50 epochs. So, the Figure. 3 give an example the accuracy and loss of various CNN models across 50 epochs.

 ISSN: 2959-3379 (Print), 2959-3387 (Online)

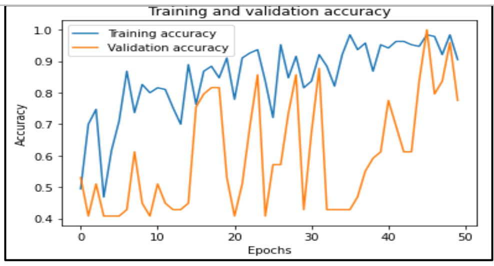
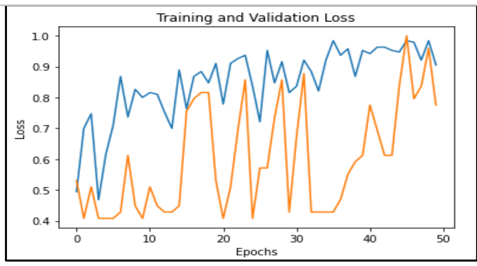
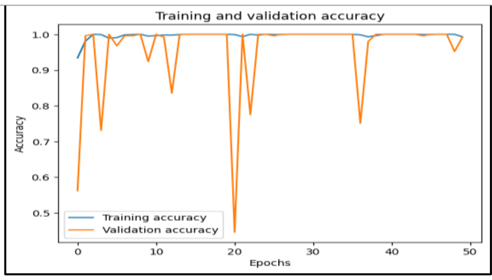
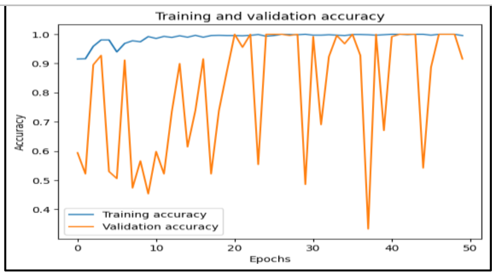
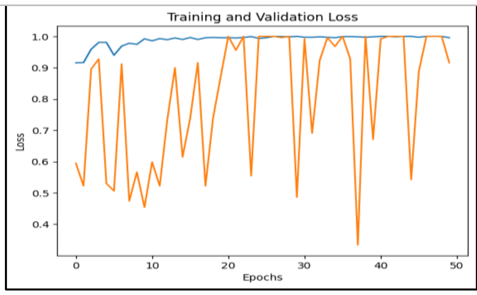
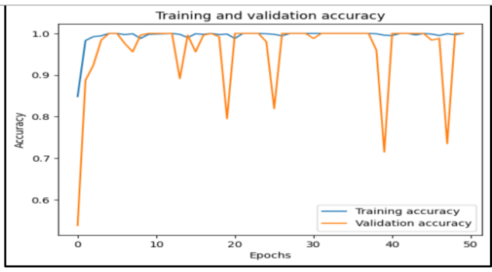
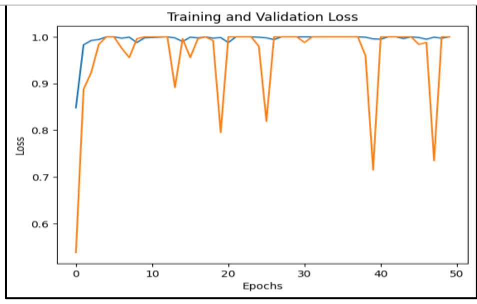
<table>
<tr><th>Models</th><th>Training and validation accuracy</th><th>Training and validation loss</th></tr>
<tr><td>VGG16</td><td>
</td><td>
</td></tr>
<tr><td>MobileNet</td><td>
</td><td>
</td></tr>
<tr><td>DenseNet_201</td><td>
</td><td>
</td></tr>
<tr><td>Inception-ResNet_v2</td><td>
</td><td>
</td></tr>
<tr><td>Xantho_Net</td><td>
</td><td>
</td></tr>
</table>

**Figure.3. Accuracy and loss of proposed Xantho_Net and state-of-art CNN models**

**Table 4. The evaluation measures of the proposed Xantho_Net model**

| | Precision | Recall | F1-Score |
|---|---|---|---|
| **Bacterial spot** | 1.00 | 0.99 | 1.00 |
| **Healthy** | 0.99 | 1.00 | 1.00 |
| **Accuracy** | - | - | 1.00 |
| **Macro avg** | 1.00 | 1.00 | 1.00 |
| **Weighted avg** | 1.00 | 1.00 | 1.00 |

The results indicate that the Xantho_Net predicts the test data with an overall accuracy of 97%. Figure. 4 shows the confusion matrix derived for the CNN model.

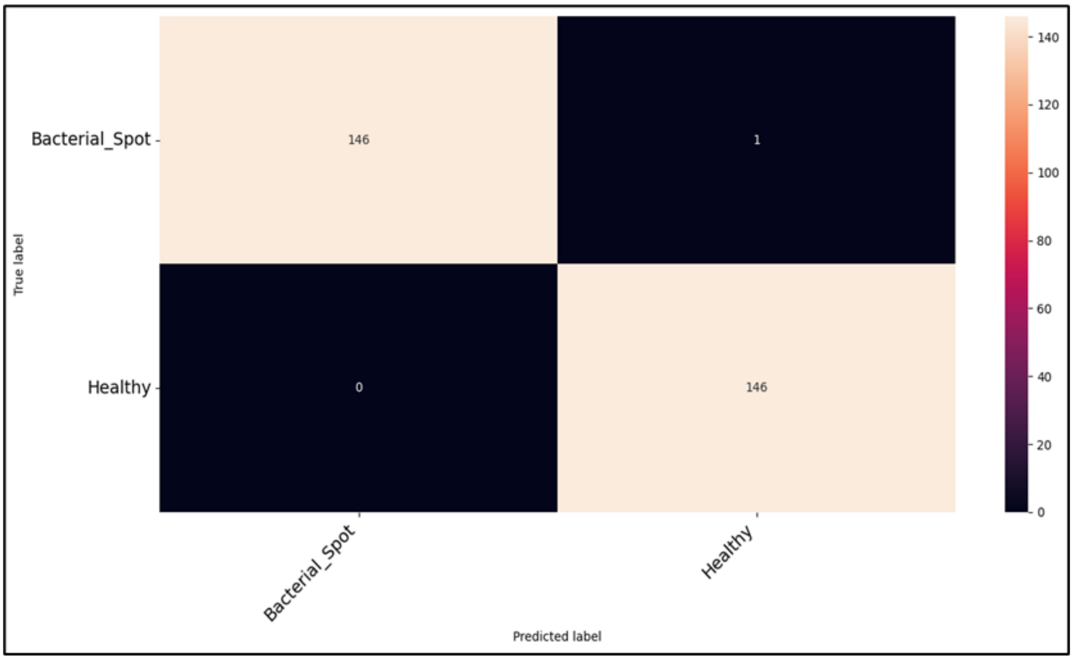



**Figure 4. Confusion matrix**

**Table 5. Numerous studies for the classification and diagnosis of diseases affecting fruits and vegetables**

| Author/ Authors | Model Architecture | Images | Disease/ Diseases | Accuracy |
|---|---|---|---|---|
| Tarek *et al.*, 2022 [32] | ResNet50, InceptionV3, Alexnet, MobileNetV1, MobileNetV2, and MobileNetV3 | 16004 | Septoria leaf spot, target spot, yellow leaf curl virus, bacterial spot, leaf mold, mosaic virus, late and early blight, and two-spotted spider mite | 99.80%.99.62%,96.68%,99.49%,98.93%,99.81%,98.99% |
| Al-Gaashani *et al.*, 2022 [33] | MobileNetV2 and NASNetMobile | 1152 | Leaf mold, Septoria leaf spot, yellow leaf curl virus, | 97%, 97% |

| | | | | |
|---|---|---|---|---|
| | | | bacterial spot, healthy, and late blight | |
| Zhang *et al.*, 2019 [26] | RCNN, RCNN-mobile, and RCNN-res101 (combined k-means analysis) faster | 4178 | Mosaic virus, powdery mildew, and leaf mold fungus | 97.01%, 97.31%, 98.54% |
| Verma *et al.*, 2020 [34] | AlexNet, SqueezNet, Inception V3 | 2342 | Tomato Late Blight: Initial, Intermediate, and Final Stages | 90.43%, 93.40%, 90.76% |
| Verma *et al.*, *2020* [34] | Faster R-CNN (VGG-16 ResNet-50 ResNet-101) and Mask R-CNN (MobileNet ResNet-50 ResNet-101) | 286 | Fruit with defects, blotchy ripening, dehiscent fruit, rot at the blossom end, viral illness, gray mold,) | 86.09%, 88.41%, 88.53%, 88.39%, 98.52%, 99.64% |
| Gerdan *et al.*, 2023 [35] | VGG16, MobileNet, Densenet201, Inception ResNetV2 and Custom CNN model | 18440 | Leaf mold, Septoria leaf spot, target spot, bacterial spot, early blight, mosaic virus, and yellow leaf curl viral | 92.12%, 92.75%, 91.50%, 84.12%, 99.82% |
| Proposed model Xhoant_Net | VGG16, MobileNet, DenseNet_201, Inception-ResNet_v2 | 2000 | Bacterial spot, healthy leaf | 78.09%<br>99.05%<br>98.11%<br>85.75%<br>100% |

## DISCUSSION

Several investigations have been carried out to enhance the categorization and diagnosis of ailments impacting fruits and vegetables. Only works on the identification of tomato disease are included in Table5. The quantity of photos in the collection, the identified ailment, the models used, and the model accuracy percentages.

Pre-trained CNN models produced the best results, according to a review of all the studies. The 99.64% accuracy result generated by Wang et al., 2019 is the highest accuracy percentage reported in the literature. Muhammad et al. [3] reported an accuracy percentage of 99.18% in their 2017 study. Furthermore, the deep learning applications in this study used the largest number of samples out of all the data sets used in the literature for tomato disease diagnosis. The proposed approach classified tomato plant illnesses with good levels of accuracy. The models employed were effective however, other experiments using pre-trained models had similar outcomes. Experimental experiments with 50 epochs have produced the following results. With a 99.05% accuracy rate, in 14.88 minutes, this experimental research was finished.

## CONCLUSION

A wide range of deep learning strategies are utilized in the process of disease detection and classification of tomato leaf diseases. Compared to other transfer learning techniques, such as VGG16, MobileNet, DenseNet_201, and Inception-ResNet_v2, the Xantho_Net performed incredibly well in terms of disease classification in tomato crops. The proposed model achieves 100% training and 99% testing accuracy. Due to these methods, farmers will be able to address plant identification problems without plant scientists. This will help them cure tomato plant infections in time, improving tomato crop quality, quantity, and profit.

## FUTURE WORK

The future work of this research is to calculate the total amount of nutrients (Carbohydrates, Calories, Fat, Protein, Potassium, Vitamin, and Folate) in tomatoes through computer vision.

## REFERENCES


1. Faostat, F. (2019). Production crops http://www. fao. org/faostat/en/# data. QC [URL].
2. Farooq, K., Mubarik, A., & Aqsa, Y. (2020). Potato Cluster Feasibility and Transformation Study. Cluster Development Based Agriculture Transformation Plan Vision-2025. Project, 131, 434.
3. Muhammad, N., Shah, S. A., Jan, A. U., Ullah, I., Ibrahim, M., & Khan, S. (2017). Allocative efficiency analysis of tomato growers in Mohmand Agency, Pakistan. Sarhad J. Agric, 33(3), 366-370.
4. Rodriguez-R, L. M., Grajales, A., Arrieta-Ortiz, M. L., Salazar, C., Restrepo, S., & Bernal, A. (2012). Genomes-based phylogeny of the genus Xanthomonas. BMC microbiology, 12(1), 1-14.
5. Sundin, G. W., Castiblanco, L. F., Yuan, X., Zeng, Q., & Yang, C. H. (2016). Bacterial disease management: challenges, experience, innovation and future prospects: challenges in bacterial molecular plant pathology. Molecular plant pathology, 17(9), 1506-1518.
6. Dai, X., Gao, G., Wu, M., Wei, W., Qu, J., Li, G., & Ma, T. (2019). Construction and application of a Xanthomonas campestris CGMCC 15155 strain that produces white xanthan gum. Microbiologyopen, 8(2), e00631.
7. Yang, H., Ni, J., Gao, J., Han, Z., & Luan, T. (2021). A novel method for peanut variety identification and classification by Improved VGG16. Scientific reports, 11(1), 15756.
8. Velásquez, A. C., Castroverde, C. D. M., & He, S. Y. (2018). Plant–pathogen warfare under changing climate conditions. Current biology, 28(10), R619-R634.
9. Stall, R., Gottwald, T., Koizumi, M., & Schaad, N. (1993). Ecology of plant pathogenic xanthomonads. In Xanthomonas (pp. 265-299). Springer.
10. Morales, C., Posada, J., Macneale, E., Franklin, D., Rivas, I., Bravo, M., Minsavage, J., Stall, R., & Whalen, M. (2005). Functional analysis of the early chlorosis factor gene. Molecular plant-microbe interactions, 18(5), 477-486.
11. Agrios, G. N. (2005). Plant pathology. Elsevier.
12. Burlakoti, R. R., Hsu, C.-f., Chen, J.-r., & Wang, J.-f. (2018). Population dynamics of xanthomonads associated with bacterial spot of tomato and pepper during 27 years across Taiwan. Plant Disease, 102(7), 1348-1356.
13. Singh, V. K., Singh, A. K., & Kumar, A. (2017). Disease management of tomato through PGPB: current trends and future perspective. 3 Biotech, 7, 1-10
14. Wang, Q., Qi, F., Sun, M., Qu, J., & Xue, J. (2019). Identification of tomato disease types and detection of infected areas based on deep convolutional neural networks and object detection techniques. Computational intelligence and neuroscience, 2019.
15. Bashan, Y., Diab, S., & Okon, Y. (1982). Survival of Xanthomonas campestris pv. vesictoria in pepper seeds and roots in symptomless and dry leaves in non-host plants and in the soil. Plant and Soil, 68, 161-170.
16. Emana, B., Afari-Sefa, V., Nenguwo, N., Ayana, A., Kebede, D., & Mohammed, H. (2017). Characterization of pre-and postharvest losses of tomato supply chain in Ethiopia. Agriculture & Food Security, 6(1), 1-11.
17. Deng, L., & Yu, D. (2014). Deep learning: methods and applications. Foundations and trends® in signal processing, 7(3–4), 197-387.
18. Dhakal, A., & Shakya, S. (2018). Image-based plant disease detection with deep learning. International Journal of Computer Trends and Technology, 61(1), 26-29.
19. Türkoğlu, M., & Hanbay, D. (2019). Plant disease and pest detection using deep

learning-based features. Turkish Journal of Electrical Engineering and Computer Sciences, 27(3), 1636-1651.

20. Tan, L., Lu, J., & Jiang, H. (2021). Tomato leaf diseases classification based on leaf images: a comparison between classical machine learning and deep learning methods. AgriEngineering, 3(3), 542-558.
21. Kattenborn, T., Leitloff, J., Schiefer, F., & Hinz, S. (2021). Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. ISPRS journal of photogrammetry and remote sensing, 173, 24-49.
22. Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., Modi, K., & Ghayvat, H. (2021). CNN variants for computer vision: History, architecture, application, challenges and future scope. Electronics, 10(20), 2470.
23. Sayeed, A., Choi, Y., Eslami, E., Lops, Y., Roy, A., & Jung, J. (2020). Using a deep convolutional neural network to predict 2017 ozone concentrations, 24 hours in advance. Neural Networks, 121, 396-408.
24. Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D. F., Weber, J., Webb, G. I., Idoumghar, L., Muller, P.-A., & Petitjean, F. (2020). Inceptiontime: Finding alexnet for time series classification. Data Mining and Knowledge Discovery, 34(6), 1936-1962.
25. Constable, E. C. (2020). The Publications of Howard Flack (1943–2017). In (Vol. 2, pp. 645-651): MDPI.
26. Zhang, K., Guo, Y., Wang, X., Yuan, J., & Ding, Q. (2019). Multiple feature reweight densenet for image classification. IEEE Access, 7, 9872-9880.
27. Wightman, R., Touvron, H., & Jégou, H. (2021). Resnet strikes back: An improved training procedure in timm. arXiv preprint arXiv:2110.00476.
28. Ronald, M., Poulose, A., & Han, D. S. (2021). iSPLInception: An inception-ResNet deep learning architecture for human activity recognition. IEEE Access, 9, 68985-69001.
29. Batool, S. N., & Gilanie, G. (2023). CVIP-Net: A Convolutional Neural Network-Based Model for Forensic Radiology Image Classification. Computers, Materials & Continua, 74(1).
30. Gilanie, G., Nasir, N., Bajwa, U. I., & Ullah, H. (2021). RiceNet: convolutional neural networks-based model to classify Pakistani grown rice seed types. Multimedia Systems, 1-9.
31. Ferdouse Ahmed Foysal, M., Shakirul Islam, M. Abujar, S., & Akhter Hossain, S. (2020). A novel approach for tomato diseases classification based on deep convolutional neural networks. Proceedings of International Joint Conference on Computational Intelligence: IJCCI 2018
32. Tarek, H., Aly, H., Eisa, S., & Abul-Soud, M. (2022). Optimized deep learning algorithms for tomato leaf disease detection with hardware deployment. Electronics, 11(1), 140.
33. Al-gaashani, M. S., Shang, F., Muthanna, M. S., Khayyat, M., & Abd El-Latif, A. A. (2022). Tomato leaf disease classification by exploiting transfer learning and feature concatenation. IET Image Processing, 16(3), 913-925.
34. Verma, S., Chug, A., & Singh, A. P. (2020). Application of convolutional neural networks for evaluation of disease severity in tomato plant. Journal of Discrete Mathematical Sciences and Cryptography, 23(1), 273-282.
35. Gerdan, D., Caner, K., & Vatandaş, M. (2023). Diagnosis of Tomato Plant Diseases Using Pre-trained Architectures and A Proposed Convolutional Neural Network Model. Journal of Agricultural Sciences, 29(2), 618-629.