



A novel deep learning method for detection and classification of plant diseases

Waleed Albattah¹ · Marriam Nawaz² · Ali Javed² · Momina Masood² · Saleh Albahli¹ 

Received: 11 May 2021 / Accepted: 7 September 2021 / Published online: 28 September 2021
© The Author(s) 2021

Abstract

The agricultural production rate plays a pivotal role in the economic development of a country. However, plant diseases are the most significant impediment to the production and quality of food. The identification of plant diseases at an early stage is crucial for global health and wellbeing. The traditional diagnosis process involves visual assessment of an individual plant by a pathologist through on-site visits. However, manual examination for crop diseases is restricted because of less accuracy and the small accessibility of human resources. To tackle such issues, there is a demand to design automated approaches capable of efficiently detecting and categorizing numerous plant diseases. Precise identification and classification of plant diseases is a tedious job due because of the occurrence of low-intensity information in the image background and foreground, the huge color resemblance in the healthy and diseased plant areas, the occurrence of noise in the samples, and changes in the position, chrominance, structure, and size of plant leaves. To tackle the above-mentioned problems, we have introduced a robust plant disease classification system by introducing a Custom CenterNet framework with DenseNet-77 as a base network. The presented method follows three steps. In the first step, annotations are developed to get the region of interest. Secondly, an improved CenterNet is introduced in which DenseNet-77 is proposed for deep keypoints extraction. Finally, the one-stage detector CenterNet is used to detect and categorize several plant diseases. To conduct the performance analysis, we have used the PlantVillage Kaggle database, which is the standard dataset for plant diseases and challenges in terms of intensity variations, color changes, and differences found in the shapes and sizes of leaves. Both the qualitative and quantitative analysis confirms that the presented method is more proficient and reliable to identify and classify plant diseases than other latest approaches.

Keywords Agriculture · CenterNet · DenseNet · Deep learning · Plant disease

Introduction

According to the Food and Agriculture Organization (FAO) of the United Nations, the population of the world will rise to 9.1 billion by 2050. Therefore, to tackle the nutriment needs of such a huge number of people, the growth rate of food should be increased to 70% by 2050 [1]. However, several factors are constraints in increasing food productivity i.e., unavailability of large regions for cultivation and clean water. Moreover, crop diseases result in a substantial reduction in

both the quantity and quality of crops. Such damages have a profound effect on the economy which ultimately results in a decrease in farmer's income and an increase in food prices for the customers. Furthermore, it may lead to intense food shortages following severe hunger and starvation particularly in under-developed states where the availability of preventive measures is limited. Mostly, crop diseases are identified by human resources through physically visiting the crop areas. However, it is a time-consuming activity and practically impossible for humans to analyze each plant [2]. Therefore, frequent delays are faced in the analysis procedure of crop diseases due to the limited availability of manpower. Hence, crop disease recognition at an initial phase is mandatory as it not only enhances the production rate of the food but can also protect the farmers from costly spray processes. To cope with the challenges of manual plant disease detection, the research community has shifted its focus to introduce and

✉ Saleh Albahli
Salbahli@qu.edu.sa

¹ Department of Information Technology, College of Computer, Qassim University, Buraydah, Saudi Arabia

² Department of Computer Science, University of Engineering and Technology, Taxila 47050, Pakistan

examine computer-aided approaches to ease this complex job and to develop effective automated methods for crop analysis [3].

Initially, techniques proposed in molecular biology and immunology were utilized to detect crop diseases at the earliest stage [4, 5]. However, these approaches required human experts, huge resources, and cost to be established. According to FOA, the majority of cultivation areas are small and run by people in under-developed nations having low income [6]. Therefore, such expensive solutions are impractical for them and researchers need to propose efficient and effective approaches that are accessible to all farmers [7]. Recently, new machine learning (ML) based hand-coded approaches are employed in the field of agriculture to enhance decision-making power [8]. Due to the advancement of digital methods, a huge amount of information is being gathered in real-time on which ML-based approaches are applied to make an optimized decision. These methods like decision trees (DT) [9], support vector machines (SVM) [10], K-nearest neighbors (KNN) [11], and Gaussian frameworks [12], etc. are also heavily tested for crop disease detection. Hand-coded keypoints computation methods are easier to apply and do not need a huge amount of training data, however, such techniques are time-consuming and need the skills of human expertise. Furthermore, the conventional ML-based keypoints computation methods always have a trade-off between computational complexity and detection robustness. As the computation of large keypoints vector increases the economic burden while utilizing a small feature-set reduces the localization efficiency of the system [13]. Therefore, still, there exists a need for performance improvements, particularly for decision-support frameworks that assist in converting the huge amount of data into valuable recommendations.

Now, we witnessed the efficacy of deep learning (DL)-based approaches such as CNN [14], Recurrent neural networks (RNNs) [15], and deep belief networks [16, 17] in several application areas including image segmentation [18], classification [19], change detection [20] and agriculture. DL-based techniques for example CNN are empowered to automatically perceive the optimal key points from the input samples without the need of human experts. The working of the DL-based framework imitates the processing of the human brain. People identify and localize objects visually by observing numerous object samples. These approaches follow the same procedure for object and pattern recognition. DL architectures generate more precise results than custom ML-based methods, which allow improved decision making. Due to the extensive progression of hardware equipment, DL frameworks are heavily explored to solve challenging problems in reasonably minimum time duration. DL-based methods exhibit state-of-the-art accuracy in the field of agriculture and generalized well to several tasks. Different forms

of deep neural networks (DNNs) have attained significant performance over hyperspectral analysis [21]. Some of the popular CNN models are GoogLeNet [22], AlexNet [23], VGG [24], and ResNet [25] which performed well in crop-related classification tasks i.e. predicting the yield quantity, crop heads detection, fruit counting, plant disease detection, classification, and many others. These frameworks are capable of achieving robust results with minimum preprocessing and computation effort. Furthermore, DL-based techniques deliver better performance for object localization by exploiting the topological data from the input images and are invariant to post-processing attacks i.e. rotation, translation, etc. In addition, these methods generate better quality output with pre-trained frameworks. Although, a huge amount of work has been presented for crop disease detection and classification, however, still there is a room for improvement. The ability of ML methods to resolve challenging real-world problems is remarkable over human brain intelligence. The major issues of ML-based techniques overcrop disease detections are the less efficacy and high processing time as these frameworks produce lengthy codes that increase the computational complexity. To deal with the issue of long codes, DL methods have emerged, however, at the cost of enhanced code complexity. Moreover, existing methods are not generalized well to real-world scenarios and are not applicable to identify several diseases over one sample or to locate multiple instances of the same disease in one image [26].

An efficient and effective automated localization and classification of crop disease is still a complex task because of the presence of huge variations in the size, shape, color, and position of leaves. Moreover, the variations in brightness during the leaves image capturing process also complicates the detection process. In this work, we have tried to deal with the aforementioned challenges by introducing a customized CenterNet framework with DenseNet-77 at the keypoints computation level to calculate the deep key points of input samples and localization and classification of various plant diseases. Our results show that the presented technique is robust to changes in size, rotation, color, brightness, contrast, lightning conditions, blurring, and extensive noisy input samples. The main contributions of the introduced framework are as follows:

- We present a customized CenterNet model with DenseNet-77 for features computation to improve the detection and classification accuracy of the plant diseases while minimizing the training and testing time complexity.
- Our method provides accurate localization of the affected portion of plant leaves due to the robustness of the CenterNet model.
- Our method achieves improved classification accuracy of plant leaves diseases due to the power of the CenterNet model to tackle the over-fitted model training data.

- We proposed a computationally efficient technique for plant leaves disease detection as CenterNet employs one stage object detection framework.
- Extensive experimentation has been carried out in comparison to other latest plant disease detection approaches on a standard PlantVillage database that is diverse in terms of distortions like blurriness, chrominance, intensity variations, high-density noise, rotational, and scaling variations to exhibit the robustness of the presented approach.

The remaining of the paper has been structured as: "Related work" contains the related work, while "Proposed method" comprises a detailed description of the proposed model. In "Experiment and results", the detection performance of our framework is demonstrated. And, "Conclusion" draws the conclusion of our work.

Related work

In this part, we have performed a critical analysis of the work from the literature being employed for plant disease identification and classification. The work from researchers for plant disease classification is categorized into two types namely ML-based techniques or DL-based approaches.

Le et al. [27] introduced a method to detect and classify crops and weeds-based diseases. Initially, morphological opening and closing operations were performed to remove the noise from the input images. Then a customized framework namely filtered local binary pattern method with contour mask and coefficient k (k -FLBPCM) was proposed to compute the features from the processed sample. The extracted features were used to train the SVM classifier to categorize several plant diseases. The technique in [27] shows better plant disease classification accuracy, however, it may not perform well over the samples with perspective distortions. Ahmad et al. [28] proposed a framework to locate and classify plant diseases. In the first step, Directional Local Quinary Patterns (DLQP) were applied to the input image to compute the keypoints. Then the SVM classifier was trained on computed key points to obtain the plant disease classification results. This approach [28] shows better plant disease recognition accuracy, however, performance can be further improved by employing the shape and color-based information of the input sample. Sun et al. [29] proposed a framework to recognize and classify tea plant diseases. In the first step, Simple Linear Iterative Cluster (SLIC) was applied to convert an input sample into blocks, on which the Harris method was applied to extract the significant key points. In the next step, the convex hull approach was used to obtain the fuzzy salient region contour and Gray Level Co-occurrence Matrix (GLCM) approach was used for keypoints extraction which was used for training the SVM classifier to

perform the tea plant leaf diseases classification. The method in [29] exhibits better classification accuracy, however, it is suffering from high computational cost. Pantazi et al. [2] proposed an approach to recognize and classify various plant diseases. Initially, the GrabCut method was applied to the input image to perform segmentation. Then, the HSV transform was applied over the segmented sample. In the next step, LBP was applied over the obtained ROIs to compute the features, which were later used to train the class SVM classifier. This approach [2] works well for plant disease classification, however, exhibits poor detection accuracy for noisy samples. Similarly in [30], a hand-coded feature extraction-based approach was presented for plant disease identification and classification. In the first step, the input samples were resized and the histogram equalization (HE) method was applied over them to enhance the visual quality of images. Then the K-means clustering approach was employed over the processed samples to perform the segmentation. In the next phase, GLCM and LBP descriptors were applied over the segmented regions to compute the features. In the last step, the SVM classifier was trained on the calculated key points to perform the plant disease classification. The method in [30] shows better classification accuracy, however, results are reported for a small dataset. Ramesh et al. [31] introduced an approach for plant disease classification. Histogram of Oriented Gradients (HOGs) features was employed for feature extraction and trained the Random Forest (RF) classifier using these features to classify the samples into healthy and diseased categories. The method [31] is robust to plant disease classification, however, performance needs further improvement. Kuricheti et al. [32] introduced a method to classify turmeric leaf diseases. After preprocessing, the K-means algorithm was applied to the input image to perform the image segmentation. Then GLCM approach was applied for features extraction, based on which the SVM classifier was trained to perform the leaves classification. This method [32] exhibits better plant disease classification results, however, unable to provide better performance on samples having huge brightness variations. From the above discussed ML-based approaches, it can be concluded that these works are simple to apply, however, requires extensive training data and are highly dependent on the skills of human expertise. Moreover, these techniques are not robust to the extensive variations in the size, color, and shapes of leaf plant diseases. Therefore, there is a need of more robust approach employing the latest approaches to improve the recognition accuracy of several Plant leaf diseases [33, 34].

Recently, DL approaches are highly explored in several automated applications because of their high recall rate [35]. Argüeso et al. [36] presented a DL-based method namely Few-Shot Learning (FSL) to detect and classify plant disease. Initially, the Inception V3 framework was employed to calculate the key points. Then a multiclass support vector

machine (SVM) was trained over the extracted features. This method [36] is robust to plant disease classification, however, results are reported on a small size dataset and need to be evaluated on a large-scale and diverse corpus. Agarwal et al. [37] proposed a CNN-based architecture to localize and categorize the tomato crop disease. This method [37] consisted of 3 convolutions along with max-pooling layers to extract the key points from the input samples and to categorize them. This approach [32] exhibits better tomato disease classification accuracy, however, this framework suffers from the problem of over-fitting over a small number of classes. Richey et al. [38] introduced a mobile-app-based technique to detect and classify maize crop disease. A DL-based model namely ResNet50 was trained over the ImageNet database for computing the deep key points from the input images and to classify them into respective classes. The approach in [38] provides a mobile phone-based solution to crop disease classification, however, this method is computationally complex and not much suitable for cellphones due to constraints of limited memory, processor, and battery power. Zhang et al. [39] introduced an improved DL-based framework for classifying tomato crop diseases. In [39], researchers introduced a custom Faster-RCNN approach in which the deep residual framework was employed for features extraction instead of the VGG16 model. Moreover, the k-means clustering approach was utilized to group the bounding boxes. The method [39] shows better tomato crop disease classification results, however, at the increase of economic burden. Batool et al. [40] proposed an approach to identify and categorize tomato leaf disease at the early stage. In the first stage, the AlexNet framework was employed for extracting the deep keypoints from the input sample that were then used to train the KNN for the classification of the images as healthy or affected. This approach [40] shows better classification accuracy, however, KNN is a slow and time-consuming algorithm. Goncharov et al. [41] introduced a DL-based model for classifying diseases of various crops like wheat and corn. A deep Siamese network was used to compute the deep features that were later used to train the KNN for image categorization. The framework in [41] is robust to plant disease detection, however, it suffers from the problem of over-fitting on a large-scale dataset. Karthik et al. [42] presented a DL-based approach to detect disease in the tomato leaves. In [42], a residual network was applied to compute the deep features on input samples. Then a CNN classifier was trained to compute key points to classify the healthy and affected leaves. This approach [42] shows better leaves disease classification accuracy, however, this method is economically inefficient. TM et al. [43] introduced a DL-based framework to localize and categorize tomato leaf diseases. Initially, the input samples were resized before using them for further processing. Then a DL model namely LeNet was utilized to extract the points and classify the samples into

healthy and affected classes. The approach in [43] provides a low-cost solution for tomato crop disease classification, however, unable to show robust performance for noisy samples. Sembiring et al. [44] proposed a solution to classify the tomato plant leaf diseases. A lightweight CNN framework comprising four layers was used to compute the deep features of suspected samples and classify them into ten different classes. This work [44] is computationally efficient, however, exhibits lower performance for real-world scenarios. In [45], Turkoglu et al. proposed an ensemble technique in which several DL-based models namely AlexNet, GoogleNet, DenseNet201, ResNet50, and ResNet101 frameworks were used to compute the deep keypoints of several plants. In the next step, the computed features were employed for the SVM training to categorize several plant diseases. The approach [45] shows better plant leaf classification performance, however, at the expense of increased features computation cost.

The analysis of existing methods employed for crop disease detection is presented in Table 1. From the Table 1, it can be witnessed that there is still a need for performance enhancement both in terms of classification performance and time complexity.

Proposed method

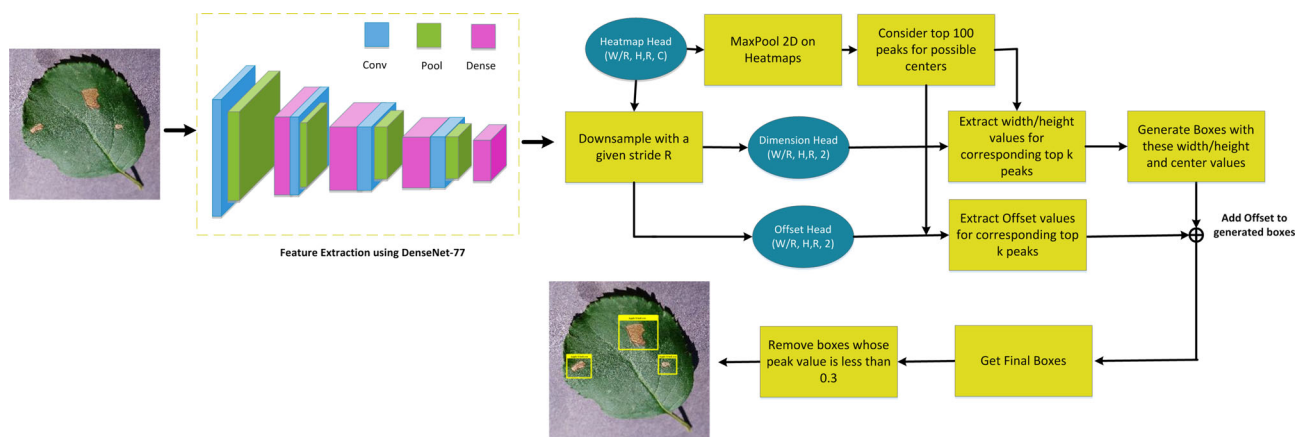
The introduced framework is comprised of two main phases i.e., the transfer learning phase and the localization and classification phase. The complete functionality is shown in Fig. 1. The implementation of the proposed technique consists of two modules. The first part is named ‘dataset preparation’ while the other is the CenterNet network trained for plant disease localization and categorization. In the dataset preparation step, the annotations for plant images are developed to specify the region of interest (RoI). While in the second module trained CenterNet over the annotated samples. We employed CenterNet with DenseNet-77 as its base network for feature computation. The features extractor of the CenterNet framework namely DenseNet-77 takes inputs of two types (suspected image and annotations). Figure 1 demonstrates the workflow of the proposed method. In the beginning, a suspected image along with the annotation’s bounding box (bbox) is passed to the DenseNet-77 framework. The bbox recognizes the RoI in the CNN keypoints therefore, reserved features from input samples are designated for further processing. After features computation, the CenterNet is trained to classify the located areas. Finally, accuracies are estimated for all units as per metrics being employed in the field of computer vision. A detailed description of the proposed method is given in Algorithm 1.

Table 1 Comparison of existing techniques

References	Method	Performance	Advantage	Limitation
[27]	A framework named k-FLBPCM along with SVM was used for crop disease classification	Accuracy = 98.63%	The work assisted to enhance classification accuracy for plants with similar morphological textures	Detection accuracy degrades for the distorted samples
[28]	The DLQP approach with the SVM classifier was introduced to categorize the various plant diseases	Accuracy = 96.53%	This work is robust to detect the plant leaf disease classification under intense scale and angle variations in input samples	Classification performance needs further improvements
[29]	Harris method was used along with the GLCM approach for features computation while the SVM classifier was employed for tea plant disease classification	Accuracy = 98.5%	The approach is capable of detecting the affected leaves portion from the complex background	This method suffers from a high computational cost
[2]	The LBP algorithm together with the SVM classifier was employed for plant disease classification	Accuracy = 95%	The model has better generalization power	Classification performance degrades over noisy samples
[30]	The key points were computed via employing GLCM and LBP descriptors, while the classification of plant disease was performed with the SVM	Accuracy = 98.2%	The framework can locate the diseased plant portion from the suspected samples under the presence of intense light variations	Results are reported for a small dataset
[31]	The HOG approach with the RF classifier was employed to categorize the diseased plant samples into various classes	Accuracy = 70.14%	The work is computationally efficient	Performance needs further improvement
[32]	The K-means clustering, GLCM methods along with SVM classifier were utilized to classify turmeric leaf diseases	Accuracy = 91%	The work can locate the diseased leaves of plants from the blurry samples	Classification performance degrades for samples having huge brightness variations
[36]	A DL-based method named FSL was introduced to detect and classify plant disease	Accuracy = 91.4%	The approach requires less training data	Results are reported for a small dataset
[37]	A CNN-based architecture was presented to localize and categorize the tomato crop disease	Accuracy = 91.2%	The technique is computationally efficient	This method suffers from the issue of over-fitting over a small number of classes
[38]	A mobile-app-based technique employing a DL-based model namely ResNet50 was utilized to classify the various maize crop diseases	Accuracy = 99%	The model has better generalization power	This method may not work well for all mobile phones due to processing power and battery consumption requirements
[39]	A custom Faster-RCNN model was introduced for classifying tomato crop diseases	mAP = 97.18%	The work is robust to the presence of noise and distortions in suspected samples	This approach is economically inefficient
[40]	The DL framework namely AlexNet along with the KNN classifier was used to classify the tomato leaves as being healthy or affected	Accuracy = 76.1%	The work can identify the affected region from low-intensity images	This approach is slow and time-consuming
[41]	A deep Siamese network together with KNN classifier was used for plant disease classification	Accuracy = 96%	The technique has improved the classification accuracy for samples with complex backgrounds	This method is suffering from the problem of over-fitting for a large-size dataset

Table 1 continued

References	Method	Performance	Advantage	Limitation
[42]	A CNN-based framework employing a residual network was introduced for deep features computation and classification	Accuracy = 98%	The work is robust to noisy samples	This method is computationally expensive
[43]	A DL-based framework using LeNet was utilized to extract the key points and classify the samples into healthy and affected classes	Accuracy = 94.8%	The work requires less training data	This framework is not robust for noisy images
[44]	Employed a lightweight CNN for tomato leaf disease classification	Accuracy = 97.15	This work is Computationally efficient	Only evaluated for tomato leaf disease classification and not robust to real-world scenarios
[45]	AlexNet, GoogleNet, DenseNet201, ResNet50, and ResNet101 along with the SVM classifier were used for plant leaf disease classification	Accuracy = 97.56%	The work is robust to plant disease classification under the presence of light variations	The approach is suffering from a high computational cost

**Fig. 1** Flow diagram of Proposed Custom CenterNet model with DenseNet-77 feature extractor

Annotations

For an efficient training process, it is mandatory to accurately specify the position of the affected region from the input plant samples. For this purpose, we have utilized the LabelImg [26] software to build the sample annotations. Some visual results are reported in Fig. 2. The generated annotations are stored in an XML file which comprises two important details: (i) class associated with each affected region, and (ii) bbox values for drawing a rectangular box over the detected region. In the next step, the training file is produced from an XML file which is utilized for training the model.

CenterNet

An efficient keypoints computation is needed to precisely categorize plant diseases into numerous classes. However,

computing a discriminative set of the feature vector is a challenging task because of the following reasons: (i) models can result in over-fitting by employing the large size keypoints vectors, and (ii) while utilizing a small key points-set, the technique may miss learning some important object behaviors i.e., structure and color changes which cause affected regions of disease indistinguishable from the healthy leaves. To achieve the discriminative and robust image keypoints, it is mandatory to employ an automated features computation technique without the need of using hand-crafted features calculation. The models using handcrafted key points are not robust to correctly identify and recognize plant diseases because of extensive changes in the size, texture, color, and position of plant leaves. To deal with the challenges, we employed a DL-based framework namely CenterNet because of its ability to directly compute the effective features from the input samples. The convolution filters of CenterNet

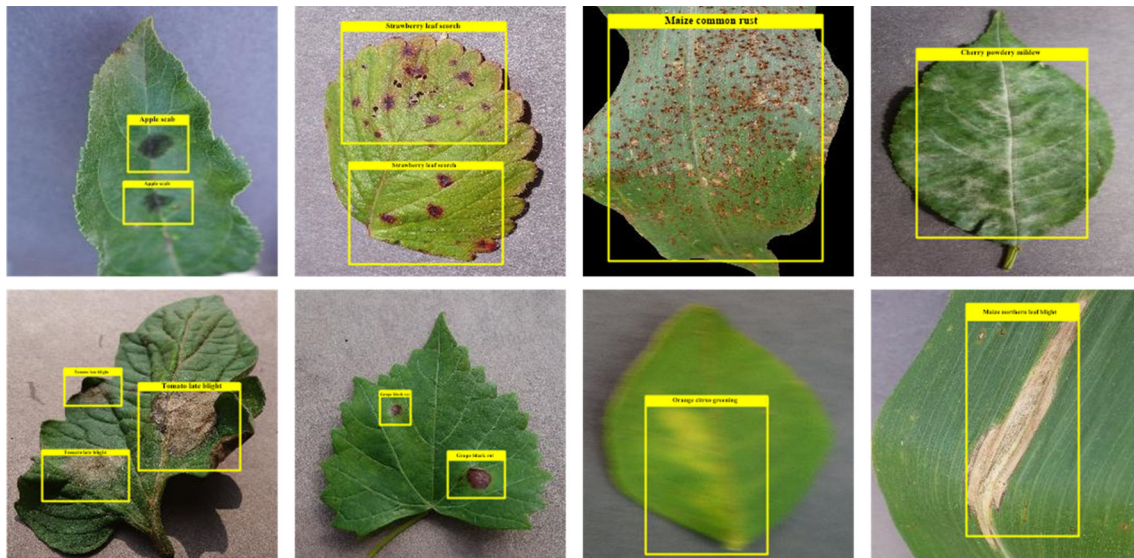


Fig. 2 Annotated samples of the PlantVillage dataset to show the region of interest

extract the features of the suspected sample by analyzing its structure.

The motivation of using the CenterNet [46] over the RCNN [47], Fast-RCNN [48], Faster-RCNN [49] approaches for plant disease recognition is that these approaches perform classification by following a two-stage object detector. In [46–49], initially, a Region Proposal Network (RPN) is employed to locate the region of interest (RoIs) which possibly surround an object. Then, using the collective key points intimate each RoIs, and separated identification heads of framework detect the category of object and draws the rectangular box. Therefore, these methods are computationally inefficient and are impractical for real-time object detection requirements. The CenterNet better tackles the problems of RCNN and Fast-RCNN and Faster-RCNN by specifying both features and location boxes of objects in input samples at the same time. Therefore, the one-stage object detection power of CenterNet makes it computationally efficient and better generalize to real-time object detection.

For plant disease classification, it is a complex task to identify the key points of interest due to the following causes: (i) locating the exact location of the affected region from the input sample due to intense light and color variations, and (ii) category of each detected object. The CenterNet framework can accurately identify and categorize the affected regions of varying categories by employing its heat maps and by replacing the two-stage object detection with a one-stage recognition algorithm. The Heat-map module works by employing the center of key points and attains better recall performance, which assists to minimize the keypoints extraction time of the presented model.

Custom CenterNet

The conventional CenterNet framework employed the ResNet-101 as a feature calculator for performing object detection and classification [46]. However, the ResNet model uses skip-connections and identity methods to evade non-linear transformations which result in the direct flow of gradient from the back to the front layers. The ResNet-101 approach comprises a large number of parameters, which eventually results in the vanishing gradient problem. To deal with this issue of the ResNet-101 framework, we introduce a densely associated convolution framework named DenseNet as the backbone network of the traditional CenterNet approach by replacing the ResNet-101 with DenseNet-77 [50]. The proposed feature computer DenseNet-77 contains a small number of parameters with thinner layers network as compare to ResNet-101 which gives a computational benefit over ResNet-101. DenseNet has numerous dense blocks (DBs) that are consecutively linked with each other via employing added convolutional and pooling layers in sequential dense blocks [51, 52]. DenseNet model can show the complicated transformation efficiently which assists in dealing with the problem of the lack of the object's location data for the significant features to some extent. Moreover, DenseNet supports the keypoints transmission procedure and boosts their reuse which makes it more convenient for crop disease classification and improves the speed of the training process. Therefore, in the presented work, we have utilized the denseNet-77 for extracting the deep features in the CenterNet framework. Table 2 shows the description of the trainable parameters for the Custom CenterNet.

The CenterNet framework follows two main steps to perform the plant disease classification which are feature

Table 2 Training parameters of the presented methodology

Model parameters	Value
No of epochs	30
Value of learning rate	0.001
Selected batch size	8
The threshold for the confidence score	0.2
The threshold for the unmatched region	0.5

extraction and detection. A detailed description is provided in this section.

Feature extraction using DenseNet-77

The presented Densenet-77 framework has two major changes from conventional DenseNet: (i) Densenet-77 comprises fewer parameters from the real framework, and (ii) the layers inside every DB are attuned to cope with the problem of economic cost. Table 3 presents the structure of the introduced DenseNet-77 framework that shows the layer’s name used for keypoints computation to perform the advanced processing by the CenterNet.

The DB is the major element of DenseNet-77 as presented in Fig. 3, where for the N -layer, $n \times n \times m_0$ exhibits the features maps (FPs), whereas, n and m_0 denote the FPs size and total channels, respectively. To minimize the channels, a non-linear transformation denoted by $H(\cdot)$ is applied which comprised various operations namely Batch Normalization (BN), Rectified linear unit (Relu), and a 1×1 convolution layer (ConL). Moreover, a 3×3 ConL is utilized to perform the keypoints reorganization. The long-dashed arrow is used to show the dense connection which combines the preceding and the next layer through the value computed by $H(\cdot)$. To end, $n \times n \times (m_0 + 2m)$ is the resultant value of the $N + 1$ layer. The extensive dense connections increase FPs considerably, therefore, the transition layer (TL) is introduced to reduce the keypoints size than the earlier DB.

The computed features are down-sampled with the stride rate $R = 4$, which are then passed to compute three types of heads.

Heatmap head The heatmap head computes a key point estimation over the down-sampled deep features from the DenseNet-77 framework to locate the affected regions of plants together with the respective class. Whereas, the key points are box center in case of object detection i.e. plant regions which are computed as follows:

$$\hat{o}_{i,j,c} = \exp\left(-\frac{(i - \hat{p}_i)^2 + (j - \hat{p}_j)^2}{2\sigma_p^2}\right). \tag{1}$$

Here, i and j are presenting the actual ground truth key point coordinates, while \hat{p}_i and \hat{p}_j are showing the locations of predicted down-sampled key points. σ and p are showing the object size-adaptive standard deviation and c is showing the number of classes. The $\hat{o}_{x,y,c}$ is presenting the center for a candidate key points if it has a value of one, otherwise, it is marked as background.

Dimension head The Head is responsible for predicting the coordinates of the bbox. The dimension of the bbox for a candidate object k with class c having coordinates (x_1, x_2, y_1, y_2) can be estimated through the $L1$ norm which is $(x_2 - x_1, y_2 - y_1)$.

Offset head The offset head is computed to minimize the discretization error which occurs due to performing down-sampling over the input sample. After computing the center points, these points are again mapped to a higher dimensional input image.

Multi-loss function CenterNet is the end-to-end learning technique that employs multi-loss methods to improve its performance and accurately localize the affected region with the corresponding class. Designing an effective loss function is mandatory for the robust performance of a model, therefore, the CenterNet model uses a multi-loss method. The benefit of employing the multi-loss function is that it assists the model to accurately differentiate between the actual and predicted value.

The employed multi-loss L on each sample head is defined as:

$$L_{CenterNet} = L_{map} + \lambda_{dim}L_{dim} + \lambda_{off}L_{off}. \tag{2}$$

Here $L_{CenterNet}$ presents the total loss computed by the CenterNet, whereas, the L_{map} , L_{dim} , and L_{off} are representing Heatmap, Dimension, and Offset head losses, respectively. Moreover, λ_{dim} and λ_{off} are constants with the values of 0.1 and 1, respectively. The Heatmap loss L_{map} is computed as follows:

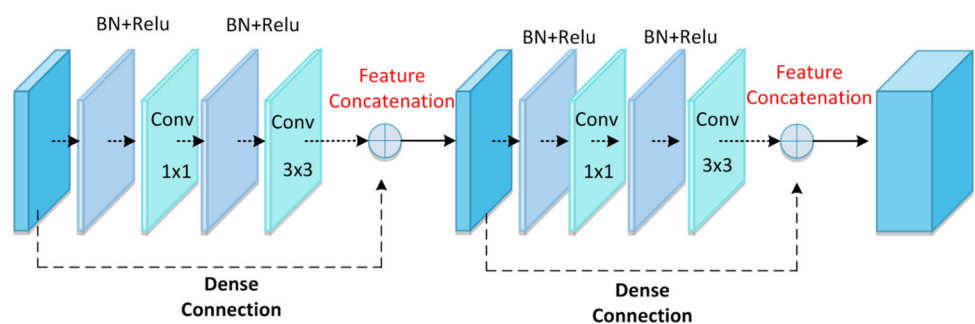
$$L_{map} = \frac{-1}{n} \sum_{i,j,c} \begin{cases} (1 - \hat{o}_{i,j,c})^\alpha \log(\hat{o}_{i,j,c}), & \text{if } \hat{o}_{i,j,c} = 1 \\ (1 - O_{i,j,c})^\beta (\hat{o}_{i,j,c})^\alpha \\ \log(1 - \hat{o}_{i,j,c}) \end{cases} \tag{3}$$

Here, n is the total number of key points, $O_{i,j,c}$ is presenting the actual candidate key point center, while $\hat{o}_{i,j,c}$ is the predicted key point center. Moreover, α and β are the hyperparameters of Heatmap loss with the values of 2 and 4

Table 3 Architectural description of DenseNet-77

Layer		Densenet-77	
		Size	Stride
ConL 1		$7 \times 7conv$	2
PoolL 1		3×3	2
DB 1		$max_pooling$ $\begin{bmatrix} 1 \times 1conv \\ 3 \times 3conv \end{bmatrix} \times 6$	1
TL	ConL 2	$1 \times 1conv$	1
	PoolL 2	2×2	2
DB 2		$avg_pooling$ $\begin{bmatrix} 1 \times 1conv \\ 3 \times 3conv \end{bmatrix} \times 12$	1
TL	ConL 3	$1 \times 1conv$	1
	PoolL 3	2×2	2
DB 3		$avg_pooling$ $\begin{bmatrix} 1 \times 1conv \\ 3 \times 3conv \end{bmatrix} \times 12$	1
TL	ConL 4	$1 \times 1conv$	1
	PoolL 4	2×2	2
DB 4		$avg_pooling$ $\begin{bmatrix} 1 \times 1conv \\ 3 \times 3conv \end{bmatrix} \times 6$	1
Classification_layer		7×7 $avg_pooling$ Fully connected layer SoftMax	

Fig. 3 Pictorial representation of DB which shows the layers architecture



for all our experiments, respectively. The Dimension Head loss is calculated as:

$$L_{dim} = \frac{1}{n} \sum_{k=1}^n |\hat{b}_k - b_k|. \tag{4}$$

Here, \hat{b}_k is the predicted bbox coordinates, while b_k is showing the actual dimensions of bboxes from ground truths. Finally, the Offset-head loss is calculated as:

$$L_{off} = \frac{1}{n} \sum_p \left| \hat{F}_{\hat{p}} - \left(\frac{p}{R} - \hat{p} \right) \right|. \tag{5}$$

Here, \hat{F} is presenting the predicted offset value, whereas, p and \hat{p} represents the actual and down-sampled key point.

Detection process

CenterNet is a DL-based approach that is independent of approaches i.e. selective search and proposal generation. So, the suspected image along with generated annotation is passed as input to the framework, on which the CenterNet framework calculates the center points of the plant diseases portion, offsets to the x and y coordinates, and the dimensions of bboxes along with the target class.

Table 4 Description of PlantVillage dataset

Class No	Class Name	Cause of disease	Training samples	Validation samples	Test samples
	Apple				
1	Scab	Fungus	441	126	63
2	Black_Rot	Fungus	435	124	62
3	Cedar_Rust	Fungus	192	55	28
4	Healthy	–	1151	329	165
	Blueberry				
5	Healthy	–	1051	300	151
	Cherry				
6	Healthy	–	598	171	85
7	Powdery_Mildew	Fungus	736	210	106
	Maize				
8	Common_Rust	Fungus	835	238	119
9	Healthy	–	813	233	116
10	Northern_Leaf_Blight	Fungus	690	197	98
11	Gray_leaf_spot	Fungus	360	102	52
	Grape				
12	Black Rot	Fungus	826	236	118
13	Black_Measles	Fungus	968	277	138
14	Healthy	–	296	85	42
15	Leaf_Blight	Fungus	753	215	108
	Orange				
16	Huanglongbing	Bacteria	3855	1101	551
	Peach				
17	Bacterial Spot	Bacteria	1608	459	230
18	Healthy	–	252	72	36
	Pepper Bell				
19	Bacterial_Spot	Bacteria	698	199	100
20	Healthy	–	1034	297	147
	Potato				
21	Early_Blight	Fungus	700	200	100
22	Healthy	–	107	30	15
23	Late_Blight	Infection	700	200	100
	Raspberry				
24	Healthy	–	260	74	37
	Soybean				
25	Healthy	–	3563	1018	509
	Squash				
26	Powdery_Mildew	Fungus	1285	367	183
	Strawberry				
27	Healthy	–	319	91	46
28	Leaf_Scorch	Fungus	776	222	111
	Tomato				
29	Bacterial_Spot	Bacteria	1488	426	213
30	Early_Blight	Fungus	700	200	100
31	Healthy	–	1114	318	159
32	Late_Blight	Infection	1336	382	191

Table 4 continued

Class No	Class Name	Cause of disease	Training samples	Validation samples	Test samples
33	Leaf_Mold	Fungus	667	190	95
34	Septoria_leaf_Spot	Fungus	1240	354	177
35	Spider_Mites	Mite	1174	335	167
36	Target_Spot	Fungus	984	280	140
37	Mosaic_Virus	Virus	262	74	37
38	Yellow_Leaf	Virus	3750	1071	536

Experiment and results

This section provides a comprehensive investigation of the obtained results after performing various experiments to measure the evaluation power of the presented approach. The description of the dataset used for performance evaluation is also demonstrated in this section. The presented framework was implemented using Python and executed on an Nvidia GTX1070 GPU-based system. In the introduced approach, instead of training the model from scratch, we used the framework with pre-trained weights acquired from MS-COCO and performed transfer learning to fine-tune the CenterNet on the PlantVillage dataset for plant leaf disease classification and classification.

Dataset

The detection and classification performance of the presented method is evaluated over the PlantVillage dataset [53]. The Plantvillage database is an extensive and publically available standard dataset of plant disease classification which is heavily used by existing approaches for performance evaluation. To analyze the robustness of the introduced technique, we performed several experiments on this database, which comprises several categories of plants and their diseases. The employed dataset namely the PlantVillage contains 54,306 images of plant leaves, with 12 healthy and 26 diseased plant classes of 14 species of plants. The samples for all 14 species of the crop including *Tomato*, *Potato*, *Apple*, *Grape*, etc., are downloaded from the Plantvillage dataset. The samples in the Plantvillage database are varied in the aspect of changes

in angle, size, color, light, and the presence of blurring, and noise, etc., which makes it a diverse database for plant disease recognition. Figure 4 presents few samples from the Plantvillage dataset and Table 4 shows the detailed description of the dataset.

Evaluation metrics

We have measured our proposed technique using different evaluation metrics e.g., Intersection over Union (IOU), accuracy, precision, recall, and mean average precision (mAP). We computed the accuracy as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}. \quad (6)$$

Equation 7 shows the mAP calculation, in which AP denoted the average precision of each class and q is the query or test image. Q is the total number of test images:

$$mAP := \sum_{i=1}^T AP(t_i)/T. \quad (7)$$

Equations 8, 9 and 10 represent the IOU, precision, and recall, respectively.

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (8)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (9)$$

$$IoU = \frac{TP}{FN + FP + TP} \times 2. \quad (10)$$



Fig. 4 Samples images from the PlantVillage dataset of different classes



Fig. 5 Test results of custom CenterNet for plant disease recognition where shows the bounding box with obtained localization score and associated class

Performance evaluation of plant disease localization

The correct detection of several plant diseases is important to build an effective model for the automated recognition of crop diseases. For this purpose, we investigated the localization

power of the presented technique by experimenting. We have tested all the samples of the PlantVillage dataset and visually presented 90 samples in Fig. 5. It can be seen from the reported results that Custom CenterNet can accurately detect and recognize plant diseases of varying categories. Moreover,

Table 5 Class-wise performance of the proposed technique

Class_Name	Precision	Recall	F1-Score
Apple_Scab	0.992	0.982	0.987
Apple_Black_Rot	0.991	0.993	0.992
Apple_Cedar_Rust	0.994	0.980	0.987
Apple_Healthy	1	1	1
Blueberry_Healthy	1	1	1
Cherry_Healthy	1	1	1
Cherry_Powdery_Mildew	0.982	0.984	0.983
Maize_Common_Rust	0.996	0.987	0.991
Maize_Healthy	1	1	1
Maize_Northern_Leaf_Blight	0.991	0.990	0.990
Maize_Gray_leaf_spot	0.992	0.981	0.986
Grape_Black_Rot	0.992	0.980	0.986
Grape_Black_Measles	0.982	0.981	0.981
Grape_Healthy	1	1	1
Grape_Leaf_Blight	0.999	0.987	0.993
Orange_Huanglongbing	0.994	0.981	0.987
Peach_Bacterial Spot	0.998	0.980	0.989
Peach_Healthy	1	1	1
PepperBell_Bacterial_Spot	0.994	0.985	0.989
PepperBell_Healthy	1	1	1
Potato_Early_Blight	0.988	0.982	0.985
Potato_Healthy	1	1	1
Potato_Late_Blight	0.989	0.982	0.985
Raspberry_Healthy	1	1	1
Soybean_Healthy	1	1	1
Squash_Powdery_Mildew	0.993	0.989	0.991
Strawberry_Healthy	1	1	1
Strawberry_Leaf_Scorch	0.996	0.991	0.993
Tomato_Bacterial_Spot	0.992	0.973	0.982
Tomato_Early_Blight	0.991	0.980	0.985
Tomato_Healthy	1	1	1
Tomato_Late_Blight	0.991	0.975	0.983
Tomato_Leaf_Mold	0.989	0.983	0.986
Tomato_Septoria_leaf_Spot	0.992	0.991	0.991
Tomato_Spider_Mites	0.994	0.976	0.985
Tomato_Target_Spot	0.993	0.984	0.988
Tomato_Mosaic_Virus	0.989	0.980	0.984
Tomato_Yellow_Leaf	0.996	0.972	0.984

the presented approach is robust to numerous post-processing attacks i.e. blurring, noise, light and color alterations, and image distortions.

The localization ability of the CenterNet technique allows it to efficiently identify and locate several plant diseases. To quantitatively measure the localization power of the presented technique, we have employed two metrics namely mAP and IOU. These metrics help to analyze the recogni-

tion performance of the system for plant diseases of several types. More precisely, we attained the mAP and mean IOU of 0.99 and 0.993, respectively. Both the visual and numeric results demonstrate that the presented technique can reliably be employed to localize and classify plant diseases.

Class wise performance

The accurate detection and categorization of several plant diseases are mandatory to compute the efficacy of a technique. So, the effectiveness of the presented method in determining the class of each plant disease is also evaluated via experimentation. To accomplish this, we applied the trained Custom CenterNet classifier on all the test images of the PlantVillage dataset. The class-wise plant disease classification performance of the introduced method in terms of precision, recall, and F1-score is demonstrated in Table 5. From Table 5, we can say that the presented approach exhibits state-of-the-art performance in terms of all evaluation parameters. The key reason for the robust plant disease recognition performance is the accurateness of the presented keypoints extraction method, which shows each category of disease in a viable manner.

To additionally show the category-wise detection power of the introduced technique, we have presented the accuracies of 38 classes in a line graph, as presented in Fig. 6. The introduced Custom CenterNet attained the average accuracy values of 99.9%, 100%, 100%, 100%, 100%, 100%, 100%, 100%, 100%, 99.8%, 100%, 100%, 100%, 100%, 100%, 100%, 100%, 100%, 100%, 99.9%, 100%, 100%, 100%, 100%, 100%, 100%, 100%, 100%, 100%, 100%, 100%, 100%, 100%, 100%, 100%, 100%, 100%, 100%, and 100% for classes of 1 to 38. Our method shows the average accuracy value 99.982%, which is showing the efficacy of the proposed method.

Evaluation of DenseNet-77

An analysis is conducted to check the efficacy of the DenseNet-77 model in performing the plant leaf disease classification by comparing the obtained results with base models. To perform this task, the classification performance of the presented DenseNet-77 based CenterNet is evaluated against other base approaches i.e. Inception-v4 [54], VGG-16 [55], ResNet-50 [56], ResNet-101 [57], ResNet-152 [57], and DenseNet-121 [58] as presented in [59].

In Table 6, we have presented the comparative results of our approach with several base models both in terms of network parameters and classification performance. The values are clearly showing that the DenseNet-77 based CenterNet is more robust than its peer approaches. Furthermore, it can be witnessed from Table 6, that the VGG-16 approach contains the highest number of parameters, while, the ResNet-152

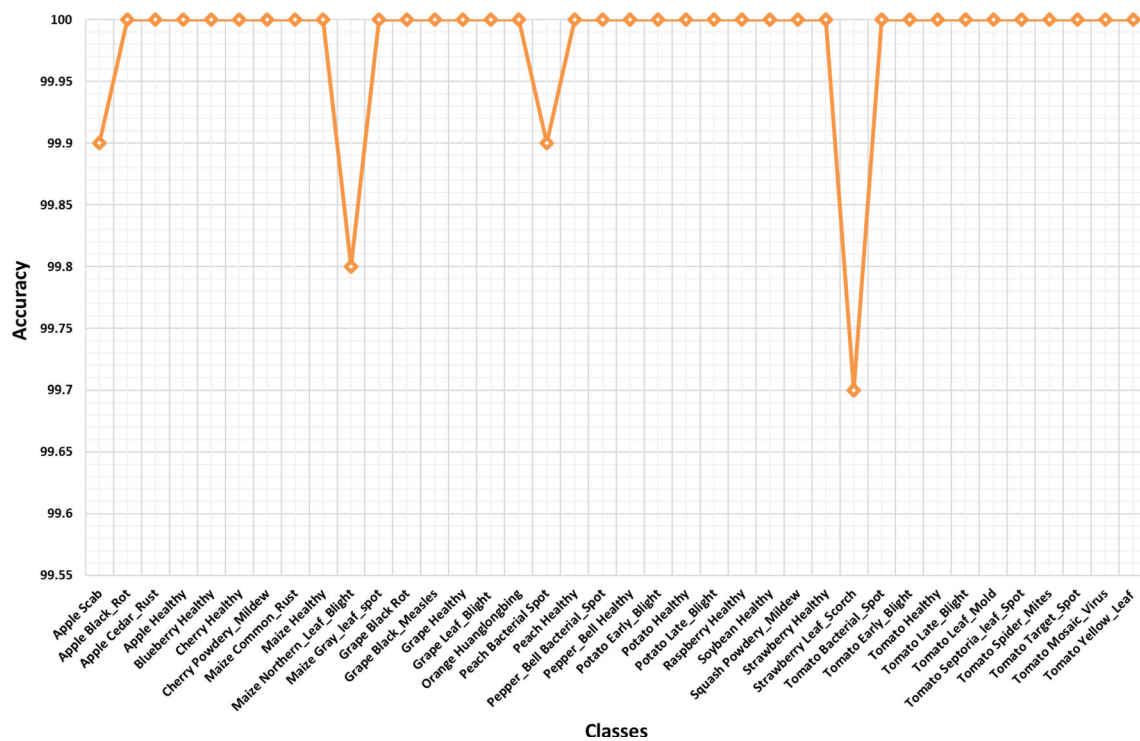


Fig. 6 Obtained accuracy rates of the proposed method on the thirty-eight classes from the PlantVillage dataset

Table 6 Comparison of the presented model with base frameworks

Parameters	Inception V4	VGG-16	ResNet-50	ResNet-101	ResNet-152	DenseNet-121	DenseNet-77
No of total model parameters (Million)	41.2	119.6	23.6	42.5	58.5	7.1	6.2
Training loss	0.0102	0.5069	$6.238e^{-04}$	$4.1611e^{-04}$	$2.4844e^{-04}$	$5.6427e^{-04}$	$6.442e^{-04}$
Test loss	0.0686	0.6055	0.02177	0.02082	0.0246	0.0159	0.0085
Train time accuracy	99.74%	83.86%	99.99%	99.99%	100%	100%	100%
Test time accuracy	98.08%	81.83%	99.59%	99.66%	99.59%	99.75%	99.983%
Execution time (s)	4042	1051	1583	2766	4366	2165	1067

has the largest processing time. Whereas, in comparison, our approach is computationally most robust as it only takes 1067 s for processing a suspected sample. The major cause of the better performance of the DenseNet-77 framework is due to its shallow network structure that makes effective reuse of model parameters without employing redundant keypoint maps. Such architectural settings of the DenseNet-77 framework cause to decrease in the number of model parameters extensively. While the comparative base networks are suffering from the high computational cost and are not robust to sample post-processing attacks i.e. blurring, noise and light variations, etc. Hence, the proposed method better deal with the problems of comparative approaches by presenting an efficient framework for keypoints computation and exhibits complex sample transformations accurately, which causes to enhance its recognition accuracy under several image distortions. It can be concluded from the discussed results that our

improved DenseNet-77 based CenterNet model shows robust accuracy in comparison to other DL-based approaches both in terms of classification performance and execution time.

Comparison with other DL-based object detection techniques

We designed an experiment to investigate the various DNN-based object detection models for plant disease classification. We have evaluated these algorithms for different scenarios like for the presence of several leaves in a single image or for the plants of different categories to check whether these models can locate the healthy and affected leaves with complicated background settings.

To accomplish this, we have considered two types of object detection models namely one-stage and two-stage detectors. The key difference among both models is that

Table 7 Comparison of the presented method with other object detection models

	Base	mAP	IOU	Test time (sec/img)
Two-stage models				
Fast-RCNN	VGG-16	0.85	0.870	0.25
Faster-RCNN	VGG-16	0.88	0.893	0.25
Faster-RCNN	ResNet-101	0.97	0.977	0.23
One-stage models				
YOLOv3	DarkNet-53	0.83	0.852	0.28
SSD	ResNet-101	0.81	0.837	0.36
RetinaNet	ResNet-101	0.92	0.902	0.38
Proposed custom CenterNet	DenseNet-77	0.99	0.993	0.21

two-stage detectors work by first locating the positions of the primary object in an image via employing several region proposal techniques, which are later narrow down, and then the final classification task is performed. Whereas, in the case of single-stage detectors, both the class and location boxes of primary objects in input samples are defined in a single step. In our work, the models from both types of detectors that have shown robust performance over the COCO dataset [60] were considered for the PlantVillage dataset and were split into 70%, 20%, and 10% for training, validation, and testing sets, respectively. The two-stage detectors namely Fast-RCNN [61] and Faster-RCNN [62] and one stage-stage detector namely RetinaNet [63] were trained over the PlantVillage database with a batch size of 32 along with stochastic gradient descent (SGD) algorithm. While for the remaining single-stage detectors namely You Only Look Once (YOLO)[64] and single shot detector (SSD) [65], the batch sizes were set to 64 and 16, respectively. Moreover, the learning rate for all detectors was set to 0.001 with a momentum value of 0.9. Table 7 shows the obtained results of the object detection frameworks on the test set.

To conduct the performance analysis of all object-detection models, we have used the mAP and IOU metrics as these were selected by many researchers as standard metrics in object identification problems. Moreover, we have compared the test time of all the models to analyze them in the aspect of computational complexity. From the reported results, it can be seen that the presented framework has attained the highest mAP value with minimum test time. Moreover, the CenterNet approach attains the IOU with a value of 0.993 which is higher than the comparative models due to its efficient features computation. The Faster-RCNN with ResNet-101 has attained comparable results with the presented technique, however, it is computationally more expensive due to its two-stage detector network. Moreover, in the case of single-stage detectors, the YOLO model is unable to locate the leaves of small sizes while the SSD computes poor features in shallow layers and results in loss of features in deep layers which results in degraded performance.

The RetinaNet can locate the leaves of small sizes, however, for the acentric features, the RetinaNet is unable to locate the important anchors. Visual results of all methods are presented in Fig. 7. The presented technique better addresses the limitations of existing one-stage and two-stage detectors by introducing Custom CenterNet with the DenseNet-77 base network. The DenseNet allows the CenterNet to learn a more representative set of features which assists in better locating the plant diseases of various categories. Moreover, the one-stage detector nature of CenterNet has given it a computational advantage over other models as well.

Comparison with ML-based classifiers

To further show the efficacy of the presented approach, we have used another experiment to show the classification performance analysis of our technique with other ML-based classifiers. For this evaluation, the two well-known ML classifiers namely SVM and KNN are chosen and results are reported in Table 8. The introduced custom CenterNet classifier attained the highest accuracy with a value of 99.982%. The SVM classifier obtained the second-best results with an accuracy of 98.01%. While the KNN exhibited lower performance with an accuracy value of 91.01%. The obtained results clearly show that our approach is more robust to plant disease detection and classification than the ML-based classifiers due to its ability to deal with the over-fitted training data.

Comparative analysis with existing methods

In this section, we have performed the comparison of our technique with the latest approaches using the same dataset for plant disease classification. For performance analysis, we compared the average highest results of our framework with the average results reported in these studies [67–70]. Table 9 demonstrates a quantitative comparison employing various performance measurement metrics i.e. precision, recall, and accuracy.

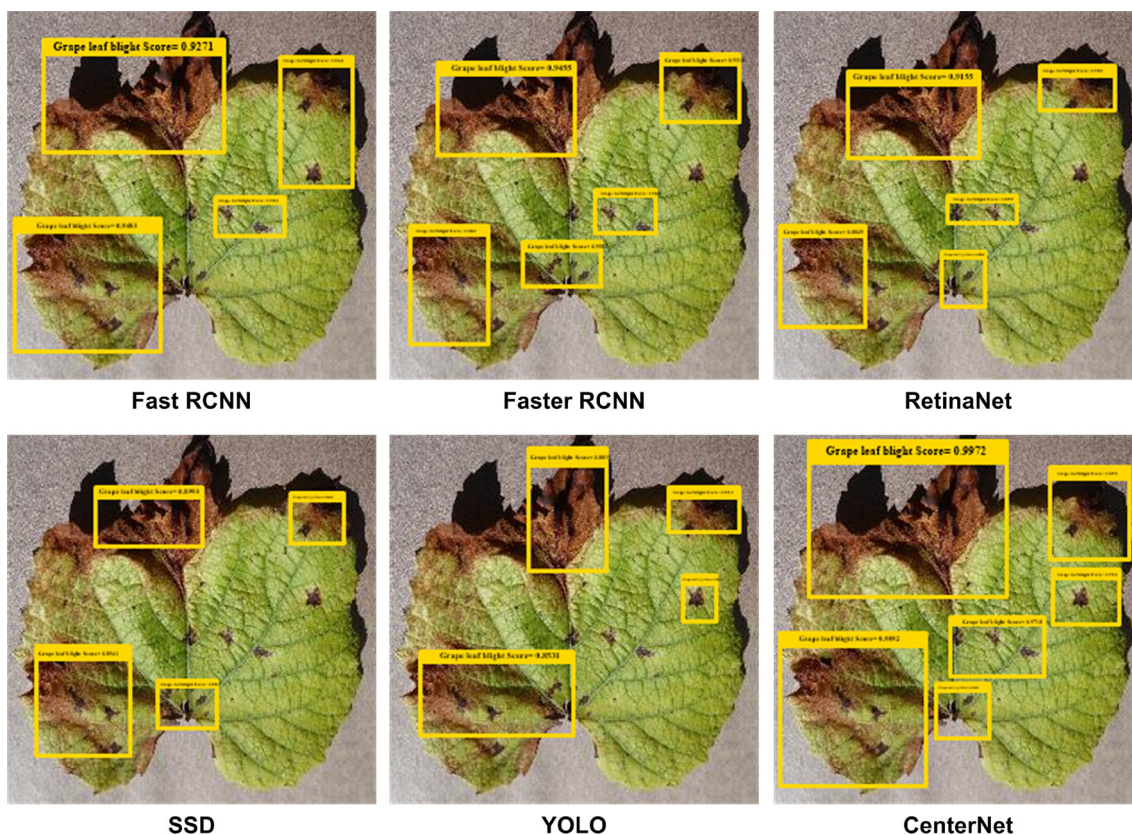


Fig. 7 Visual results of Fast-RCNN, Faster-RCNN, RetinaNet, SSD, YOLO, and proposed CenterNet framework

Table 8 Comparative analysis of proposed approach with other ML-based classifiers

Classifier	Accuracy (%)
Deep-features + SVM [66]	98.01
Deep-features + KNN [66]	91.01
Proposed	99.982

Table 9 Comparison with the state-of-the-art techniques

Technique	Precision	Recall (%)	Accuracy(%)
Mohanty et al. [67]	99.35%	99.35	99.350
Geetharamani et al. [68]	96.47%	99.89	96.460
Chen et al. [69]	–	99.01	99.850
Atila et al. [70]	99.39%	99.38	99.970
Proposed	99.52%	99.92	99.982

Mohanty et al. [67] presented a DL-based approach namely GoogleNet for the detection and classification of plant diseases and attained an average accuracy of 99.35%. Similarly, the approach in [69] also proposed a DL-based model namely MobileNet-Beta and exhibited an accuracy of 99.85%. Geetharamani et al. [68] introduced a CNN model for plant disease classification and obtained an accuracy of

96.46%. Atila et al. [70] presented a DL model namely EfficientNet for plant disease classification with an accuracy value of 99.97%. From Table 9, we can visualize that the presented framework attained 99.982% accuracy, 99.52% precision, and 99.92% recall values which are the largest in comparison to all other methods. The proposed solution exhibited an average precision of 99.52%, whereas, the competitor approaches acquired the average precision of 98.40, so, the presented technique obtained a 1.12% performance gain. Moreover, the proposed solution has a recall value of 99.92 while, the average recall value of comparison methods is 99.40, which shows 0.52% performance gain. Moreover, our work achieved 99.982% average accuracy, while the other techniques have an average accuracy of 98.907%. Thus, it can be said that custom CenterNet approach presented a 1.075% performance gain and more effective to plant disease identification and categorization in comparison to other approaches.

These results clearly show that the introduced solution outperforms the comparative methods [67–70], as these approaches deploy very deep architectures which can easily result in an over-fitting problem. However, the introduced framework uses DenseNet-77 for deep features computation that extracts more representative and reliable features based on which our Custom CenterNet provides a more effective representation of plant diseased portions over comparative

methods. Moreover, these methods [67–70] are computationally more expensive than our methodology. So, we can say that our method is more effective and efficient for the classification of plant diseases.

Conclusion

The presented approach introduces a novel framework for the automated detection and categorization of plant diseases. In this work, we have proposed a Custom CenterNet with the DenseNet-77 as a base network. More explicitly, we introduce the DenseNet-77 to extract the representative set of features from the input sample. Then, the computed key points are used to train the CenterNet classifier to recognize and classify plant diseases of numerous types. The introduced framework can efficiently locate and classify the 38 types of crop diseases from the PlantVillage dataset. Moreover, our method is robust to plant disease classification under the presence of several artifacts, i.e., variations in the light, intensity, color, size, orientation, and shapes of plant leaves. Experimental results demonstrate that the introduced model outperforms the existing latest plant diseases classification approaches. Although, we have achieved comparable results, however, the proposed model cannot be deployed on mobile-based devices, therefore, we aim to introduce some lightweight model for leaf disease recognition. Furthermore, the time complexity can be further reduced by using a more efficient feature extractor. For future work, we plan to evaluate the Custom CenterNet on the images from real-world scenarios and apply this work to other parts of plants like stems. Furthermore, we plan to check the recognition power of our method over more complex datasets and consider other state-of-the-art DL techniques.

Funding Funding was provided by Qassim University, the Deanship of Scientific Research under the number (10338-coc-2020-1-3-I), 1441 AH / 2020 AD.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Bruinsma J (2009) The resource outlook to 2050: by how much do land, water and crop yields need to increase by 2050. in Expert meeting on how to feed the world in
- Pantazi XE, Moshou D, Tamouridou AA (2019) Automated leaf disease detection in different crop species through image features analysis and One Class Classifiers. *Comput Electron Agric* 156:96–104
- Wolfenson KDM (2013) Coping with the food and agriculture challenge: smallholders' agenda. Food Agriculture Organisation of the United Nations, Rome
- Sankaran S et al (2010) A review of advanced techniques for detecting plant diseases. *Comput Electron Agric* 72(1):1–13
- Dinh HX et al (2020) Molecular genetics of leaf rust resistance in wheat and barley. *Theor Appl Genet* 133(7):2035–2050
- Ferentinos KP (2018) Deep learning models for plant disease detection and diagnosis. *Comput Electron Agric* 145:311–318
- Patil S, Chandavale A (2015) A survey on methods of plant disease detection. *Int J Sci Res* 4(2):1392–1396
- Gebbers R, Adamchuk VI (2010) Precision agriculture and food security. *Science* 327(5967):828–831
- Rokach L, Maimon O (2005) Decision trees. *Data mining and knowledge discovery handbook*. Springer, New York, pp 165–192
- Joachims T (1998) Making large-scale SVM learning practical. Technical report
- Liao Y, Vemuri VR (2002) Use of k-nearest neighbor classifier for intrusion detection. *Comput Secur* 21(5):439–448
- Birgé L, Massart P (2001) Gaussian model selection. *J Eur Math Soc* 3(3):203–268
- Bello-Cerezo R et al (2019) Comparative evaluation of hand-crafted image descriptors vs. off-the-shelf CNN-based features for colour texture classification under ideal and realistic conditions. *Appl Sci* 9(4):738
- Roska T, Chua LO (1993) The CNN universal machine: an analogic array computer. *IEEE Trans Circuits Syst II Analog Digit Signal Process* 40(3):163–173
- Zaremba W, Sutskever I, Vinyals O (2014) Recurrent neural network regularization. *arXiv preprint arXiv*
- Hinton GEJS (2009) Deep belief networks. *Scholarpedia* 4(5):5947
- Salakhutdinov R, Hinton G (2009) Deep boltzmann machines. In: *Artificial intelligence and statistics*. PMLR
- Abdollahi A et al (2020) Building footprint extraction from high resolution aerial images using Generative Adversarial Network (GAN) architecture. *IEEE Access* 8:209517–209527
- Tao Y et al (2017) GAN-assisted two-stream neural network for high-resolution remote sensing image classification. *Remote Sens* 9(12):1328
- Jian P, Chen K, Cheng W (2021) GAN-Based One-Class Classification for Remote-Sensing Image Change Detection. *IEEE Geosci Remote Sens Lett*
- Gewali UB, Monteiro ST, Saber E (2018) Machine learning based hyperspectral image analysis: a survey. *arXiv preprint arXiv:1807.01087*
- Szegedy C et al (2015) Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*
- Yuan Z-W, Zhang J (2016) Feature extraction and image retrieval based on AlexNet. In: *Eighth International Conference on Digital Image Processing (ICDIP 2016)*. 2016. International Society for Optics and Photonics.
- Vedaldi A, Zisserman A (2016) Vgg convolutional neural networks practical. *Dep Eng Sci Univ Oxford* 2016:66
- Thenmozhi K, Reddy US (2019) Crop pest classification based on deep convolutional neural network and transfer learning. *Comput Electron Agric* 164:104906

26. Paul A et al (2020) A review on agricultural advancement based on computer vision and machine learning. *Emerging technology in modelling and graphics*. Springer, New York, pp 567–581
27. Le VNT et al (2020) A novel method for detecting morphologically similar crops and weeds based on the combination of contour masks and filtered Local Binary Pattern operators. *GigaScience* 9(3):giaa017
28. Ahmad W, Shah S, Irtaza A (2020) Plants disease phenotyping using quinary patterns as texture descriptor. *KSII Trans Internet Inf Syst* 14(8):3312–3327
29. Sun Y et al (2019) SLIC_SVM based leaf diseases saliency map extraction of tea plant. *Comput Electron Agric* 157:102–109
30. Oo YM, Htun NC (2018) Plant leaf disease detection and classification using image processing. *Int J Res Eng* 5(9):516–523
31. Ramesh S et al (2018) Plant disease detection using machine learning. In: 2018 International conference on design innovations for 3Cs compute communicate control (ICDI3C). IEEE.
32. Kuricheti G, Supriya P (2019) Computer Vision Based Turmeric Leaf Disease Detection and Classification: A Step to Smart Agriculture. In: 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI). IEEE
33. Liu, M., et al., *Comparison of multi-source satellite images for classifying marsh vegetation using DeepLabV3 Plus deep learning algorithm*. *Ecological Indicators*, 2021. **125**: p. 107562.
34. Abdollahi A, Pradhan B (2021) Urban Vegetation Mapping from Aerial Imagery Using Explainable AI (XAI). *Sensors* 21(14):4738
35. Walter MJggXS (2016) Is this the end? machine learning and 2 other threats to radiology's future, 13
36. Argüeso D et al (2020) Few-Shot Learning approach for plant disease classification using images taken in the field. *Comput Electron Agric* 175:105542
37. Agarwal M et al (2020) ToLeD: Tomato leaf disease detection using convolution neural network. *Procedia Comput Sci* 167:293–301
38. Richey B et al (2020) Real-time detection of maize crop disease via a deep learning-based smartphone app. in *Real-Time Image Processing and Deep Learning 2020*. International Society for Optics and Photonics
39. Zhang Y, Song C, Zhang D (2020) Deep learning-based object detection improvement for tomato disease. *IEEE Access* 8:56607–56614
40. Batool A et al (2020) Classification and Identification of Tomato Leaf Disease Using Deep Neural Network. In: 2020 International Conference on Engineering and Emerging Technologies (ICEET). IEEE
41. Goncharov P et al (2020) Deep Siamese Networks for Plant Disease Detection. In: EPJ Web of Conferences. 2020. EDP Sciences
42. Karthik R et al (2020) Attention embedded residual CNN for disease detection in tomato leaves. *Appl Soft Comput* 86:105933
43. Tm P et al (2018) Tomato leaf disease detection using convolutional neural networks. In: 2018 Eleventh International Conference on Contemporary Computing (IC3). IEEE
44. Sembiring A et al (2021) Development of concise convolutional neural network for tomato plant disease classification based on leaf images. In: *Journal of Physics: Conference Series*. IOP Publishing
45. Turkoglu M, Yanikoğlu B, Hanbay D (2021) PlantDiseaseNet: convolutional neural network ensemble for plant disease and pest detection. *Signal, Image Video Processing*, 1–9
46. Duan K et al (2019) Centernet: Keypoint triplets for object detection. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*
47. Aceto G et al (2020) Toward effective mobile encrypted traffic classification through deep learning. *Neurocomputing* 409:306–315
48. Hinton GE, Osindero S, Teh Y-W (2006) A fast learning algorithm for deep belief nets. *Neural Comput* 18(7):1527–1554
49. Aceto G et al (2019) MIMETIC: Mobile encrypted traffic classification using multimodal deep learning. *Comput Netw* 165:106944
50. Wang Y et al (2019) Multi-Scale DenseNets-Based Aircraft Detection from Remote Sensing Images. *Sensors* 19(23):5270
51. Albahli S et al (2021) An improved faster-RCNN model for handwritten character recognition. *Arab J Sci Eng* 1–15
52. Albahli S et al (2021) Recognition and Detection of Diabetic Retinopathy Using Densenet-65 Based Faster-RCNN. *Comput Mater Contin* 67:1333–1351
53. Hughes D, Salathé M (2015) An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:08060*
54. Szegedy C et al (2016) Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:07261*
55. Yu W et al (2016) Visualizing and comparing AlexNet and VGG using deconvolutional layers. In: *Proceedings of the 33rd International Conference on Machine Learning*
56. Han S et al (2016) Dsd: Dense-sparse-dense training for deep neural networks. *arXiv preprint arXiv:04381*
57. Canziani A, Paszke, Culurciello E (2016) An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:07678*
58. Solano-Rojas B, Villalón-Fonseca R, Marín-Raventós G (2020) Alzheimer's Disease Early Detection Using a Low Cost Three-Dimensional Densenet-121 Architecture. In: *International Conference on Smart Homes and Health Telematics*. Springer
59. Too EC et al (2019) A comparative study of fine-tuning deep learning models for plant disease identification. *Comput Electron Agric* 161:272–279
60. Lin T-Y et al (2014) Microsoft coco: Common objects in context. In: *European conference on computer vision*. Springer
61. Girshick R (2015) Fast r-cnn. In: *Proceedings of the IEEE international conference on computer vision*
62. Ren S et al (2016) Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell* 39(6):1137–1149
63. Lin T-Y et al (2017) Focal loss for dense object detection. In: *Proceedings of the IEEE international conference on computer vision*
64. Redmon J, Farhadi A (2018) Yolov3: An incremental improvement. *arXiv preprint arXiv:02767*
65. Liu W et al (2016) Ssd: Single shot multibox detector. In: *European conference on computer vision*. Springer
66. Mohameth F, Bingcai C, Sada KA (2020) Plant disease detection with deep learning and feature extraction using plant village. *J Comput Commun* 8(6):10–22
67. Mohanty SP, Hughes DP, Salathé M (2016) Using deep learning for image-based plant disease detection. *Front Plant Sci* 7:1419
68. Geetharamani G, Pandian A (2019) Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Comput Electr Eng* 76:323–338
69. Chen J, Yin H, Zhang D (2020) A self-adaptive classification method for plant disease detection using GMDH-Logistic model. *Sustain Comput Inform Syst* 28:100415
70. Atila Ü et al (2021) Plant leaf disease classification using efficient-net deep learning model. *Ecol Inform* 61:101182

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.