

Received 28 July 2022, accepted 10 August 2022, date of publication 23 August 2022, date of current version 31 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3201104



## RESEARCH ARTICLE

# A Performance-Optimized Deep Learning-Based Plant Disease Detection Approach for Horticultural Crops of New Zealand

MUHAMMAD HAMMAD SALEEM<sup>ID1</sup>, JOHAN POTGIETER<sup>ID2</sup>, AND KHALID MAHMOOD ARIF<sup>ID1</sup>, (Senior Member, IEEE)

<sup>1</sup>Department of Mechanical and Electrical Engineering, School of Food and Advanced Technology, Massey University, Auckland 0632, New Zealand

<sup>2</sup>Massey AgriFood Digital Laboratory, Massey University, Palmerston North 4472, New Zealand

Corresponding author: Khalid Mahmood Arif (k.arif@massey.ac.nz)

This work was supported by the Ministry of Business, Innovation, and Employment (MBIE), New Zealand, Science for Technological Innovation (SfTI) National Science Challenge.

**ABSTRACT** Deep learning-based plant disease detection has gained significant attention from the scientific community. However, various aspects of real horticultural conditions have not yet been explored. For example, the disease should be considered not only on leaves, but also on other parts of plants, including stems, canes, and fruits. Furthermore, the detection of multiple diseases in a single plant organ at a time has not been performed. Similarly, plant disease has not been identified in various crops in the complex horticultural environment with the same optimized/modified model. To address these research gaps, this research presents a dataset named NZDLPPlantDisease-v1, consisting of diseases in five of the most important horticultural crops in New Zealand: kiwifruit, apple, pear, avocado, and grapevine. An optimized version of the best obtained deep learning (DL) model named region-based fully convolutional network (RFCN) has been proposed to detect plant disease using the newly generated dataset. After finding the most suitable DL model, the data augmentation techniques were successively evaluated. Subsequently, the effects of image resizers with interpolators, weight initializers, batch normalization, and DL optimizers were studied. Finally, performance was enhanced by empirical observation of position-sensitive score maps and anchor box specifications. Furthermore, the robustness/practicality of the proposed approach was demonstrated using a stratified k-fold cross-validation technique and testing on an external dataset. The final mean average precision of the RFCN model was found to be 93.80%, which was 19.33% better than the default settings. Therefore, this research could be a benchmark step for any follow-up research on automatic control of disease in several plant species.

**INDEX TERMS** Convolutional neural networks, deep learning, optimization algorithms, cross-validation, plant disease detection.

## I. INTRODUCTION

According to the latest fresh facts report by a New Zealand (NZ) research and development organization named Plant and Food Research, the horticultural industry achieved a record export of over NZ\$6.6 billion by June 2020 [1]. The most prominent fresh fruits were kiwifruit, apples, and avocados with an export value of NZ\$2.5, NZ\$0.9, and NZ\$0.1 billion, respectively, followed by the New Zealand

wine at NZ\$1.9 billion [1]. Furthermore, the largest crop area of 39,935 ha has been estimated for grapevines, whereas 12,905 ha of kiwifruit and 10,750 ha of apples, pears, and nashi have been reported. Based on these statistics, horticultural crops generate a great impact on New Zealand's economy. Hence, addressing the problems associated with horticultural crops could further strengthen the export value of the horticultural sector.

Among several real field problems, plant diseases affect crop yield, and quality [2], and cause economic losses [3]. The precise detection of the disease is an important step

The associate editor coordinating the review of this manuscript and approving it for publication was Turgay Celik<sup>ID</sup>.

in reducing its spread to neighboring plants, application of appropriate disease control treatments, and improving crop productivity. In this regard, this research is dedicated to the accurate identification of plant diseases by deep learning (DL) in the most valuable crops of NZ, including kiwifruit, apple, pear, avocado, and grapevine. Furthermore, several research gaps related to the dataset, real horticultural conditions, and deep learning-based plant disease recognition have been addressed in this study.

Deep learning, a subset of machine learning, has been reported in literature as a successful technique in recognizing plant diseases. A recent review summarized and compared various pre-processing steps (image resizing, data augmentation, normalization and standardization, data annotation, and others), datasets, convolutional neural networks, training techniques, deep learning frameworks, and optimization algorithms [4]. Another review article presented various modified DL models, plant disease detection tasks, problems, and challenges of DL-based plant disease classification [5]. For instance, the significance of the recent solutions for the small datasets was presented, such as transfer learning, few-shot learning, and one-shot learning. Furthermore, the early plant detection problem was explained with the use of hyperspectral imaging (HSI).

In the early stages of research in DL-based plant disease identification, the major focus remained on the classification tasks. For example, [6] did the early work in the domain of DL-based recognition and classification of plant disease by using two well-known DL models, namely AlexNet and GoogLeNet. Similarly, the task of the classification of plant diseases was presented in various articles by transfer learning and fine-tuning methods [7], [8]. These articles showed the importance of using the latest training techniques.

In the next stage, research community focused on the dataset size, particularly small datasets, as it played a significant role in the performance of the DL models. A novel data augmentation technique to classify disease in cassava leaves, tested on a modified MobileNet model, was presented in [9]. Another research [10] presented a generative adversarial network (GAN) for classifying the disease in PlantVillage dataset [11], which contains 38 classes of healthy/disease leaves for 14 plant species. Yet another article discussed a GAN-based model to classify tomato leave disease [12]. These articles formed a basis for data augmentation in plant disease detection. However, they only studied the performance of the DL models on single datasets leaving questions around their performance with other datasets containing disease in different crops.

Modification of well-known DL models is another area of research that has seen continuous focus for a long time. A modified CenterNet model with DenseNet-77 was proposed by [13] to identify plant disease while a MobileNet was modified for the classification of plant disease by [14].

A research showing the effectiveness of deep learning optimizers was presented in [15]. Then, a study was focused on the plant disease identification task that contains both

classification and localization in a single framework, using the same PlantVillage dataset [16]. Although, this research presented an improvement in the accuracy of plant disease detection and classification tasks, the major limitation was the analysis of the deep learning technology in a controlled environment dataset.

Some of the studies also focused on datasets collected in a real agricultural environment. For example, an article presented tomato disease detection in real agricultural conditions, using three DL object detectors named Faster Region-based Convolutional Neural Network (RCNN), Single-shot MultiBox Detector (SSD), and Region-based Fully Convolutional Network (RFCN) [17]. Various real-world scenarios were considered. However, the external dataset could also be tested to validate the research. A study presented a Convolutional Neural Network (CNN) named SoyNet, to classify the disease in soybean leaves after segmenting the images of leaves [18]. This study presented variations in the parameters of the DL model, such as dropout, pooling operations, and inclusion of activation functions. These adjustments were found to be successful in improving the performance of the model. Moreover, the usefulness of the proposed method was compared with other techniques. An article presented the classification of cardamom plant diseases using the EfficientNet-V2 model [19]. This study did not provide training profiles/plots. A multilayer convolutional neural network was presented for the classification of disease on mango leaves [20]. Although this article presented the significance of the DL model compared with other machine learning-based techniques, better effectiveness should have been shown by comparing it with the DL models as well. In recent research, an improved version of the Xception model was proposed for the identification of peach diseases [21]. The novelty of this work was shown by comparing the proposed method with the well-known models. However, other modified versions of the state-of-the-art DL models could be used for the analysis. Tomato disease were detected using a modified version of the you look only once (YOLO) model [22]. It was observed that the training performance of the models was presented with limited information. Another study presented a DL-based method for the detection of tomato disease divided into target and control classes [23]. This research proposed a new way of performing plant disease detection task that can open various opportunities for future research. An improved region proposal network was proposed for the detection of northern maize leaf blight [24]. A few studies have also proposed real-time detection of plant disease. A DL model was presented for the identification of tomato disease [25]. Similarly, disease detection on grape leaves was performed using a DL architecture based on Faster region-based convolutional neural network (R-CNN) [26].

After rapid advancement and research on deep learning-based plant disease identification, there are still important research gaps and considerable room for further developments to investigate the practical aspects of horticultural fields. The current literature has mainly focused on the plant

disease detection task to only plant leaves. Moreover, the available datasets emphasize on the presence of a single disease at a time in a plant leaf. Furthermore, recent studies have shown high accuracy in the PlantVillage dataset (which contains defective leaves in 14 plant species) [16], but none of the articles have provided the significance of a single deep learning model for different crops in real agricultural conditions. This is important to consider, as each crop could have different background elements. Therefore, the robustness of the DL should be analyzed for that case.

This research addresses several research questions related to the capability of DL to address various complex agricultural problems. The first question is whether deep learning can perform plant disease detection with the same trained/optimized/modified model for three problems at a time: (a) identification of diseases in several organs of plants, (b) presence of multiple diseases in a plant organ, and (c) recognition of diseases in various crops considering variations in their environments/background elements? Connecting to the previous question, can a DL architecture correctly distinguish symptomatically identical diseases in different crops? The final question is how well can the attained accuracy of the DL-based method be validated for the problems highlighted earlier?

To answer these questions, this article presents a deep learning-based performance optimization approach. First, the dataset images were collected from various New Zealand farms and horticultural fields. It contains 20 classes of healthy and defective leaves, fruits, and stems/canes of five different crops. Then, various image resizing techniques, batch normalization, and weight optimization were applied. These techniques have not yet been explored for the detection of plant disease. Furthermore, the main novelty of the RFCN model (the most suitable DL architecture obtained after comparing several models) was analyzed. In this regard, the position-sensitive score maps were empirically evaluated, and anchor boxes were modified, to obtain high accuracy for the identification of all healthy and disease classes.

This research also addresses some of the research gaps outlined in recent articles, such as the validity of deep-learning-based plant disease identification. Moreover, the data augmentation has been applied after dividing the data into training, validation, and testing sub-datasets, to avoid biased results; otherwise, there was a possibility to get similar images in the sub-datasets. Furthermore, this study provided new insights into DL-based plant disease detection, rather than giving redundant discussions using the excessively explored dataset like PlantVillage [27].

The key contributions of this research are:

- 1) A new dataset of plant diseases has been proposed for the most important horticultural crops in New Zealand, named NZDLPlantDisease-v1.
- 2) Detection of disease has been performed in multiple plant organs for five different crops.
- 3) The presence and detection of multiple diseases on a single plant organ have been addressed.

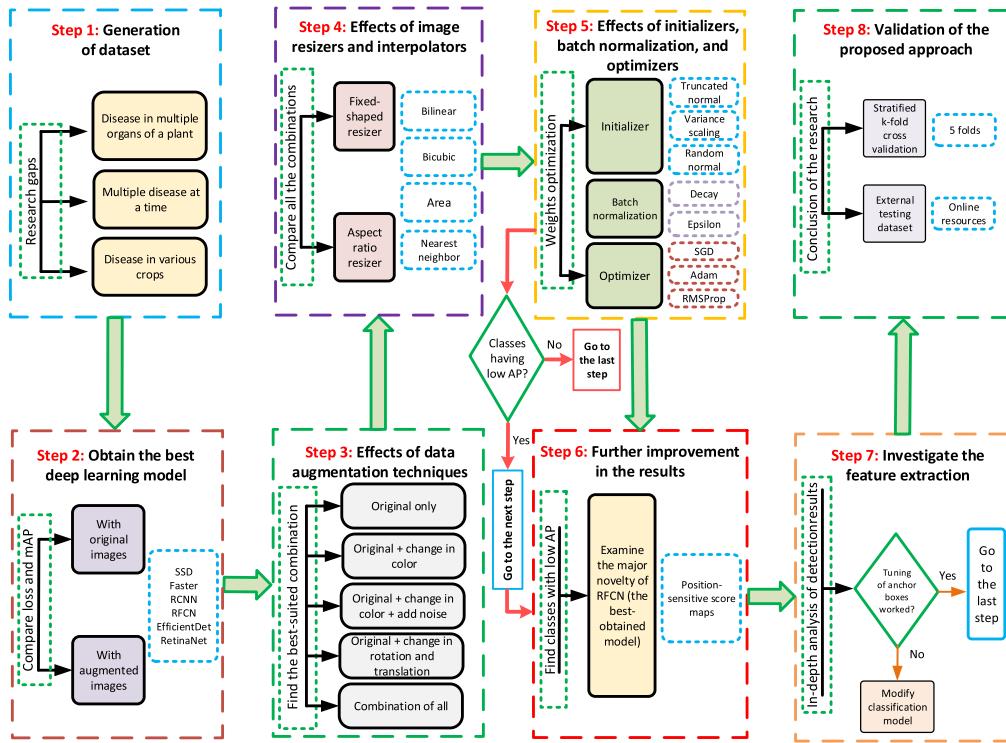
- 4) The effects of data augmentation techniques have been studied by dividing them into various categories rather than considering all conventional methods together.
- 5) A comprehensive deep learning-based plant disease detection pipeline has been presented. In this regard, various steps have been explored prior to suggesting any modification to the state-of-the-art DL models.
- 6) The confusion/false positive results in symptomatically similar diseases (occurring in different crops) have been addressed. An in-depth analysis of the best-obtained DL model named region-based fully convolution network (RFCN) has been performed by position-sensitive score maps and anchor boxes.
- 7) The proposed approach has been validated using a stratified k-fold cross-validation technique and an external testing dataset.

## II. MATERIALS AND METHODS

### A. PROPOSED APPROACH

The proposed methodology consists of various practical considerations related to the presence of plant diseases in a real horticultural environment. A comprehensive deep learning-based optimization approach has been proposed. The presented methodology has successfully solved three identified agricultural problems, including the detection of disease in multiple plant organs, the identification of disease in different crops, and the presence of multiple diseases in a plant organ at a time. These problems have been solved by different techniques presented in sub-sections. The idea was to improve the average precision of each class. The results from each of the step were evaluated, the respective problems were highlighted and addressed in the next step.

First, the research questions were outlined to begin collecting images of the dataset. Next, well-known DL architectures were compared, and the two best deep learning (DL) models were obtained, which attained the highest mean average precision. Then, the data augmentation techniques were applied category-wise, including color change (brightness, contrast, and sharpness), the inclusion of noise with variation in color, rotational and translational changes, and finally, the combination of all categories, including the original images. The next step was the performance optimization of the DL model using various techniques. In this regard, the effects of image resizers and interpolators were analyzed. This step was performed to investigate different input images for the DL-based plant disease detection. Then, different DL initializers were tested. Subsequently, batch normalization was applied to cope with the internal covariate shift. Then, DL optimizers were leveraged to optimize the weights of the deep learning model. This led to a further improvement in the performance of classes that achieved low average precision (AP). This objective was achieved by empirically analyzing the novelty of the best-obtained model. The final step was the modification of the DL model by empirically tuning its anchor box scale and aspect ratios. In case of unsatisfactory results, the feature extractor/classification model had to be modified.



**FIGURE 1.** The overall workflow of the proposed methodology.

The final results were validated using a stratified k-fold cross-validation technique and a test dataset generated through various online/open-source images. The overall methodology of this study is presented in Fig. 1

## B. NZDLPLANTDISEASE-V1 DATASET

### 1) OVERVIEW AND GENERAL INFORMATION OF THE PROPOSED DATASET

The proposed dataset has several properties of real agricultural fields that have not been presented in previous open-source datasets. A comprehensive overview of several datasets along with the new/proposed dataset for this research is presented in Table 1. Further details of the important features of the presented dataset are explained in the following subsections.

This dataset contains plant disease in five different crops in New Zealand, including kiwifruit, apple, pear, avocado, and grapevine, named NZDLPlantDisease-v1. The images were acquired by using a Samsung smartphone Galaxy S10 plus: 12 MP f/1.5-2.4 (wide), 12 MP f/2.4 (telephoto), and 16 MP, f/2.2 (ultrawide). Several local horticultural fields were visited in Auckland and Palmerston North, New Zealand. The images were taken at a working distance of 200-300 mm.

### 2) PRACTICAL CONSIDERATIONS

The dataset was collected between December 2020 and May 2021. The abrupt change in New Zealand's weather was considered a positive aspect of the dataset generation because

it helped obtain diversity in the dataset via variations in illumination and environmental conditions. Furthermore, dataset images were captured in the presence and absence of shadows to include real horticultural conditions. Several examples of these practical considerations are shown in Fig. 2.



**FIGURE 2.** Examples of healthy and defected leaves of kiwifruit and apple in the presence and absence of shadow.

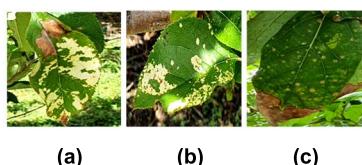
### 3) MULTI-DISEASE AND MULTI-ORGAN DATASET IMAGES

One of the research gaps addressed in this article is the detection of the disease in various organs/parts of the plants. Therefore, healthy and disease classes are considered in the leaves, stems, and fruits of apple and pear. However, the dataset classes for avocado and kiwifruit only consist of leaves. The images for the grapevine were only taken for healthy and disease cane, due to the end of the season of the grapevines at the time of dataset collection. In this research, the presence of multiple classes of disease in plant organs has also been addressed. For example, black spot, mosaic virus, and glomerella leaf spot (or two of them) were present in some of the apple leaves at one time. Similarly, algal leaf spot

**TABLE 1.** Overview of different plant disease datasets.

Datasets	Plant Organs	Number of healthy and disease classes	Main features
Cassava disease [28]	Leaves	5	Real agricultural conditions, presence, and absence of shadows
Rice disease [29]	Leaves	4	Real agricultural environment
Citrus Dataset [30]	Leaves and fruits	10	Both plain and real background
Maize disease [31]	Leaves	1	Dataset images in three different ways: a hand-held camera, a camera mounted on a UAV, and a boom. Annotations are also provided. The largest open-source dataset for any single plant leaf disease.
Grapevine [5]	Leaves	2	Real background
Arabica coffee [32]	Leaves	5	Cropped dataset, emphasized the region of interest
PlantVillage [11]	Leaves	38	Controlled environment, contains the highest number of plant disease/healthy classes
NZDLPlant Disease-v1	Leaves, stem/cane, and fruits	20	Real agricultural conditions, disease in three plant organs, five different crops, the occurrence of multiple diseases in a plant organ at a time, and various conditions like different illumination conditions, presence/absence of shadows, environmental changes

and branch canker were present on avocado leaves at the same time. Samples of multiple disease problems are presented in Fig. 3.



**FIGURE 3.** Examples of multiple disease problems: (a) shows the apple mosaic virus and the glomerella leaf spot; (b) shows the apple mosaic virus and the black spot; (c) shows the avocado algal leaf spot and the branch canker.

#### 4) ANNOTATIONS OF HEALTHY AND DISEASE CLASSES

The number of images from each class ranged from 60 to 318, as presented in Table 2. The NZDLPlantDisease-v1 dataset

**TABLE 2.** Summary of NZDLPlantDisease-v1 dataset.

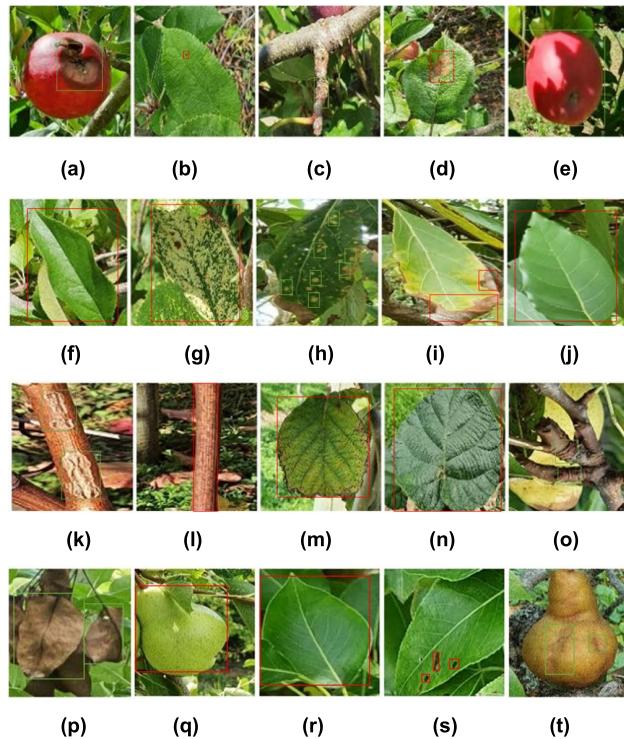
Crops	Plant Organs	Classes of NZDLPlantDisease-v1	Pathogens	Annotation labels
Apple	Leaf	Black spot (scab)- Venturia inaequalis (81)	Fungi	A_blk_spot
		Glomerella leaf spot - Glomerella cingulata (271)	Fungi	A_gl_lf_spot
		Mosaic virus (AMV) (224)	Virus	A_m_virus
	Fruit	Healthy (215)	-	A_healthy_f
		European canker - Neonectria ditissima (169)	Fungi	A_e_canker
		Black rot - Botryosphaeria obtuse (65)	Fungi	A_blk_rot_f
Avocado	Leaf	Healthy (177)	-	A_healthy_f
		Algal leaf spot - Cephaleuros virescens (104)	Algae	Av_alg_lf_spot
		Branch canker - Botryosphaeria spp. (250)	Fungi	Av_br_canker
		Healthy (226)	-	Av_healthy_1
Grapevine	Cane	Black spot - Elsinoe ampelina (80)	Fungi	G_blk_spot_c
		Healthy (170)	-	G_healthy_c
	Leaf	Bacterial canker - Pseudomonas syringae pv. Actinidiae (318)	Bacteria	Kf_bac_canker
Kiwifruit		Healthy (271)	-	Kf_healthy_1
Leaf	Fire blight - Erwinia amylovora (123)	Bacteria	P_fr_blight	
	Scab - Venturia pirina (208)	Fungi	P_scab	
	Healthy (215)	-	P_healthy_1	
Stem	European canker - Neonectria ditissima (96)	Fungi	P_canker	
	Stony pit (73)	Virus	P_s_pit	
Fruit	Healthy (209)	-	P_healthy_f	

(Parentheses indicate the number of images, and the hyphens (-) in the pathogen column indicate healthy classes.

was divided into three sub-datasets: training (70%), validation (20%), and testing (10%). The dataset images were annotated by using an open-source tool called LabelImg. The bounding box coordinates were stored in XML format, converted into CSV, and finally, to the TF records [16]. The common/scientific names of each class along with the number of images (without augmentation) are shown in Table 2. An example of each annotated healthy and disease class is presented in Fig. 4.

#### 5) DATA AUGMENTATION TECHNIQUES

When collecting the images for the dataset, some of them were taken in a group; cropping of those images increased the size of the dataset. Furthermore, several data augmentation



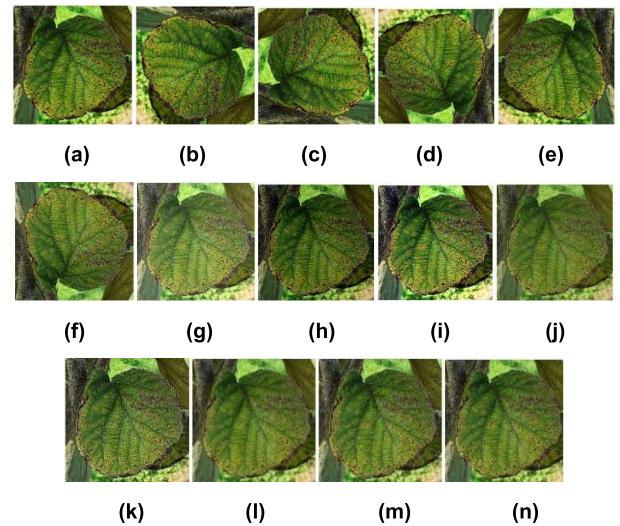
**FIGURE 4.** Annotated sample of each healthy and disease classes from the NZDLPlantDisease-v1 dataset: (a) Apple black rot, (b) Apple black spot (scab), (c) Apple European canker, (d) Apple glomerella leaf spot, (e) Apple healthy (fruit), (f) Apple healthy (leaf), (g) Apple mosaic virus, (h) Avocado algal leaf spot, (i) Avocado branch canker, (j) Avocado healthy, (k) Grapevine black spot, (l) Grapevine healthy, (m) Kiwifruit bacterial canker, (n) Kiwifruit healthy, (o) Pear canker, (p) Pear fire blight, (q) Pear healthy (fruit), (r) Pear healthy (leaf), (s) Pear scab, (t) Pear stony pit.

techniques were applied, such as a 30% increase and decrease in brightness, contrast, and sharpness [25]. Moreover, two noises are injected into the training images to study their effects and increase the variability in the dataset. In this regard, Gaussian and Laplacian noise were added by using the online available software named XnViewMP. The random intensity of 2.0 and 10.0 at a maximum scale of 10.0 and 50.0 was set, respectively. In addition, rotational/translational changes were also considered, including  $90^\circ$ ,  $-90^\circ$ ,  $180^\circ$ , horizontal, and vertical changes. An example of augmented images for a kiwifruit bacterial canker is shown in Fig. 5.

The data augmentation techniques are grouped into five categories to thoroughly understand their effects on the performance of the DL model. These categories are only original (OO), original and change in translation/rotation (OT), original and color change (OC) (brightness, contrast, and sharpness), original with an injection of noise (Gaussian and Laplacian) and color change simultaneously (OCN), and finally a combination of all (OTCN).

### C. DEEP LEARNING FRAMEWORK, HARDWARE SPECIFICATIONS, AND PERFORMANCE METRICS

All experiments are performed using the TensorFlow object-detection API. The DL models are trained using the transfer



**FIGURE 5.** Samples of kiwifruit bacterial canker with/without data augmentation methods: (a) original, (b)  $90^\circ$ , (c)  $-90^\circ$ , (d)  $180^\circ$ , (e) horizontal, (f) vertical, (g) high brightness, (h) brightness low, (i) contrast high, (j) low contrast, (k) sharpness high, (l) sharpness low, (m) Gaussian noise, (n) Laplacian noise.

learning technique with pre-trained weights on the COCO dataset. An NVIDIA GeForce GTX 1080 Ti graphics processing unit (GPU) is used with the following specifications: 11 GB memory, 1582 MHz boost clock, 3584 CUDA cores, and 484 GB/s memory bandwidth. The CuDNN library is imported to accelerate training.

The performance of the DL models is evaluated through the training and validation profiles in terms of various classification and localization losses. This helped to gain insight into the models by box classifier loss, region proposal network (RPN) loss, and total loss. Furthermore, the testing performance is presented using the mean average precision (mAP), which is a commonly used performance metric for object detection tasks [16].

### D. DEEP LEARNING MODELS

Numerous well-known DL meta-architectures have been trained, tested, and compared on the generated dataset. These models include the Single Shot Multibox Detector (SSD) [33], Faster Region-based Convolutional Neural Network (RCNN) [34], Region-based Fully Convolutional Network (RFCN) [35], RetinaNet [36], and EfficientDet [37].

The complexity of the DL models is presented by training and detection time, architectural differences, and the number of parameters, as shown in Table 3.

### E. ARCHITECTURAL OPTIMIZATION OF THE RFCN MODEL

#### 1) FUNDAMENTALS OF RFCN

Following the proposed methodology, the RFCN is selected as the best DL model for the detection of plant diseases. The main idea of this DL model is to address the ambiguity between translational invariance (identifying a particular object at different pixel values) and translational variance (identifying the exact location of the object) using position-sensitive score maps. An RFCN is a two-stage

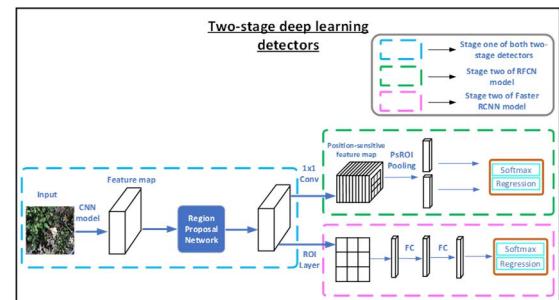
**TABLE 3.** Comparison of architectural features, number of parameters and training specification of DL architectures

DL architectures	Architectural features and loss equations	Number of parameters (in millions)	Training time (h)	Detection time (s/image)	
Faster RCNN	ResNet-50	Removes the selective search (SS) method that was present in its previous models RCNN (which runs SS multiple times) [38] and Fast RCNN (which runs SS only once) [39]. Faster RCNN used Region Proposal Network (RPN) which reduced the number of runs significantly.	43	9.5	0.35
	ResNet-101		62	10.5	0.34
	Inception-v2		13	9	0.31
	Inception ResNet-v2	$L[\{p_i\}, \{t_i\}] = \frac{1}{N_c} \sum_i L_c(p_i, p_i^*) + \lambda \frac{1}{N_r} \sum_i p_i^* L_r(t_i, t_i^*)$ (1)	59	30	0.61
SSD	Inception-v2	A single-stage DL model performs the classification and localization task in a single forward pass. The bounding box regression technique was applied	13	11.5	0.27
	MobileNet-v2	$L(x, c, l, g) = \frac{1}{N} [L_c(x, c) + \alpha L_l(x, l, g)]$ (2)	6	5.5	0.27
EfficientDet EfficientNet		A weighted bi-directional feature pyramid network (BiFPN) and a compound scaling method were proposed to allow an easy and fast multi-scale feature fusion, and scale of the depth, width, and resolution uniformly.	3.9	12	4.2
RFCN ResNet-101		Generation of position-sensitive score maps. Unlike RCNN-based models, the fully connected layer (FC) was removed after the region of interest (ROI). Therefore, the main complexity was relocated before ROI.	64	10.5	0.33
RetinaNet		$L_c = -\log(s_c^*)$ (3) $L(s, t_{x,y,w,h}) = L_c(s_c^*) + \lambda[c^* > 0]L_r(t, t^*)$ (4) Introduced feature pyramid network (FPN) to detect small objects in an image and focal losses to address the class imbalance problem. $FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t)$ (5)	36	14.5	2.1

L: overall loss,  $p_i$ : the output score from classification branch for anchor  $i$ ,  $t_i$ : output prediction of the regressor layer,  $N_c$ : number of anchors in mini-batch for classification,  $L_c$ : classification loss,  $p_i^*$ : ground truth label (0/1),  $\lambda$ : balancing parameter (for Faster RCNN),  $N_r$ : number of anchors in mini-batch for classification,  $L_r$ : regression loss,  $t_i$ : output prediction of the regressor layer,  $N_r$ ,  $L_c$ : confidence loss,  $\alpha$ : weight term,  $L_l$ : localization loss,  $L_c(s_c^*)$ : cross-entropy loss,  $\lambda$ : balancing parameter (for Faster RCNN),  $[c^* > 0]$  is an indicator, which is equal to 1 if the argument is true and 0 otherwise,  $c^*$ : ground-truth label of the region of interest (ROI),  $t^*$ : ROI's ground truth box,  $FL$ : focal loss,  $p$ : model's estimated probability for the class (RetinaNet),  $\gamma \in [0, 5]$ .

DL architecture. First, the input image is applied to the feature extraction layer using a convolutional neural network (CNN) to generate feature maps. These maps are applied to a convolutional layer to generate region-of-interest (ROI) proposals. In the former DL model named Faster RCNN, ROI proposals were used to extract the feature region in the feature map and to extract features to differentiate a particular class. However, in the RFCN, another convolutional layer is used to generate position-sensitive score maps. This map splits the ROI into  $k \times k$  bins, where each bin is used to vote for the class to which the object belongs. Therefore, the main idea was to consider the characteristics of an object divided into a region  $k \times k$  instead of as a whole. Both the RFCN and Faster R-CNN models have the same extraction of ROI proposals, but technical and computational differences arise in the application of a fully connected layer (FC) in each ROI proposal for Faster RCNN. In contrast, the RFCN generates only the proposed score maps, and the ROI is only used to vote for the regions in the score maps. Hence, the overall training and testing times of the RFCN network are much reduced than those of the Faster RCNN model [35]. This difference can also be observed in Fig. 6.

For  $w \times h$  rectangular ROI, each bin is of size  $\frac{w}{k} \times \frac{h}{k}$ . The x and y coordinates for  $(i,j)$ th bin for one of the slices of anchor

**FIGURE 6.** Architectural details of RFCN and faster RCNN models.

box bin are obtained by the following formula.

$$[i \frac{w}{k}] \leq x < (i+1) \frac{w}{k} \text{ and } [j \frac{h}{k}] \leq y < (j+1) \frac{h}{k} \quad (6)$$

Then, the pooled response  $r_c(i, j)$  on  $(i, j)$  bin for class  $c$  is equal to the sum of all the pixels within that bin coming from the position sensitive score maps. This sum is divided by number of pixels ( $n$ ) as the layer before the softmax function is the average pooling layer. Finally, we take a vote by averaging them out or taking the maximum and get the position sensitive scores that lead to the softmax to predict

the class

$$r_c(i, j|\theta) = \sum_{(x,y) \in bin(i,j)} z_{i,j,c}(x + x_0, y + y_0|)/n \quad (7)$$

$$r_c(\theta) = \sum_{i,j} r_c(i, j|\theta) \quad (8)$$

where  $\theta$  presents all learnable parameters,  $z_{i,j,c}$  is the one score map out of  $k^2(C+1)$  score maps, and  $(x_0, y_0)$  is the top left corner of the ROI.

## 2) EMPIRICAL OBSERVATIONS ON THE POSITION-SENSITIVE SCORE MAPS

The default architectural settings of the RFCN attained satisfactory outcomes for most of the plant disease classes. However, the pear scab could not be detected, showed false-positive identification. It was confused with the apple black spot. This was due to the similarity in the symptoms of both diseases. These results motivated us to empirically investigate the main novelty of the RFCN model. In this regard, the spatial bin width and height for position-sensitive score maps are tuned/analyzed for this research. The main purpose of this step is to improve the average precision (AP) of the pear scab and maintain the high AP of other classes.

## 3) PERFORMANCE ENHANCEMENT THROUGH MODIFIED ANCHOR BOXES

The final step is the improvement of classes that achieved low AP (less than 80%). These classes include apple black rot, apple black spot, apple European canker, and pear healthy (leaves) classes. In this regard, this study explored the enhancement of anchor boxes in two steps: adjustment of scale sizes and aspect ratios. Here, the scale size is gradually modified, whereas the aspect ratios are reduced/enhanced in both a step-by-step (1:2, 1:3, 1:4, and so on) and reciprocal fashion (1:2, 2:1; 1:3, 3:1; etc.). The final output attained a high AP for all healthy and diseased classes. The following steps are taken to obtain a modified or enhanced version of the anchor boxes.

- First, the RFCN model is trained by the default specifications of the scale size and aspect ratio. Therefore, the scale width and height of 256 x 256 of combinations like 1:4, 1:2, 1:1, 2:1 with aspect ratio of 1:2, 1:1, 2:1 and 1:1 are considered.
- Afterward, smaller/larger scale sizes are added to the default to understand their effects on model performance.
- After obtaining the best combination of anchor box scales, the aspect ratios are added and enhanced to obtain further refinement in the detection of plant diseases. From the default aspect ratios, the reciprocal ratios such as 1:3, 3:1, and 1:4, 4:1 are applied.
- Subsequently, an empirical adjustment of the aspect ratio is performed, and a gradual reduction/enhancement of the aspect ratio is proposed to improve the AP of several classes. The combined effect of reciprocal and gradual changes in the aspect ratio is also studied.

- Finally, the training, validation profiles and testing outcomes are compared between the proposed modifications and the default settings.

## F. IMAGE RESIZERS AND INTERPOLATORS

After obtaining the best combination of the data augmentation technique and DL architecture, the effects of image resizers on the model performance are studied. Aspect ratio and fixed shape resizers are used along with four types of image interpolators: bilinear, bicubic, area, and nearest neighbor.

## G. WEIGHTS INITIALIZERS

Three weight initialization techniques are compared to optimize the performance of the best-suited DL architecture. By default, a truncated normal is used to remove dead neurons caused by the ReLU. Then, variance scaling is applied, which is beneficial to balance the variance of the output with the input layers [40]. The last initializer is a random normal initializer used to create tensors through a normal distribution.

## H. BATCH NORMALIZATION

To accelerate training speed, batch normalization (BN) is used in this research. This technique solves the problem of internal covariate shift due to the variation in the input of the distribution of the neural network with the variation in the parameters of the previous layer [41].

The mini batch mean ( $\mu_\phi$ ) for a mini-batch ( $\phi$ ), mini-batch variance ( $\sigma_\phi^2$ ) and normalize (affine transform) are evaluated for each row of input matrix ( $x_i$ ) by:

$$\mu_\phi \leftarrow \frac{1}{N} \sum_{i=1}^N x_i \quad (9)$$

$$\sigma_\phi^2 \leftarrow \frac{1}{N} \sum_{i=1}^N (x_i - \mu_\phi)^2 \quad (10)$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_\phi}{\sqrt{\sigma_\phi^2 + \varepsilon}} \quad (11)$$

where  $N$  is the number of instances in mini batch,  $\varepsilon$  is added for the numerical stability.

There is a zero mean and variance for each component of  $x_i$  though the hidden units should have different distributions. Therefore, the normalization scheme learns the distribution by scaling the normalized values through scaling ( $\gamma$ ) and shifting ( $\beta$ ) parameters and evaluated the output of the batch normalization as follows:

$$y_i \leftarrow \gamma \cdot \hat{x}_i + \beta \equiv BN_{\gamma, \beta}(x_i) \quad (12)$$

## I. DEEP LEARNING OPTIMIZERS AND SELECTION OF HYPERPARAMETERS

In this study, three DL optimizers are used. Stochastic gradient descent (SGD) with momentum [42] is applied as a default optimization algorithm, later, root mean square propagation (RMSProp) [43] and adaptive moment estimation (Adam) Adam [44] are used to optimize the weights. A brief overview of the DL optimizers is given as follows:

### 1) SGD WITH MOMENTUM

The stochastic gradient descent (SGD) is the most commonly used optimization algorithm for neural networks. The momentum version of the SGD has a great capability of faster convergence compared to the original SGD optimizer. The exponentially weighted averages ( $Vdw$  and  $Vdb$ ) are used to evaluate the gradient and use the gradient to update the weights ( $W$ ) and biases ( $b$ ). The algorithm uses the following equations:

$$Vdw_t = \beta * Vdw_{t-1} + (1 - \beta) * dw_t \quad (13)$$

$$Vdb_t = \beta * Vdb_{t-1} + (1 - \beta) * db_t \quad (14)$$

$$W_t = W_{t-1} - lr * Vdw_t \quad (15)$$

$$b_t = b_{t-1} - lr * Vdb_t \quad (16)$$

where,  $\beta$ ,  $lr$ ,  $dw$ , and  $db$  present momentum, learning rate, gradients of the weights, and biases, respectively.

### 2) RMSProp

This DL optimizer allows to select a large learning rate. It works on the idea of using the moving average of the squared gradient and dividing the gradient by square root the mean square, using the following equations:

$$Vdw_t = \beta * Vdw_{t-1} + (1 - \beta) * dw_t^2 \quad (17)$$

$$Vdb_t = \beta * Vdb_{t-1} + (1 - \beta) * db_t^2 \quad (18)$$

$$W_t = W_{t-1} - lr * \frac{dw_t}{\sqrt{Vdw_t} + \epsilon} \quad (19)$$

$$b_t = b_{t-1} - lr * \frac{db_t}{\sqrt{Vdb_t} + \epsilon} \quad (20)$$

where,  $\epsilon$  is used for the numerical stability in the denominator.

### 3) ADAM

The Adam optimizer is a combination of RMSProp and SGD with momentum optimizers. Like RMSProp, Adam takes squared gradients to scale the learning rate and uses moving average of the gradients similar to the SGD with momentum. As it has an adaptive learning rate, it calculates separate learning rates for each parameter. Adam contains estimations of first (mean) and second moments (uncentered variance) of gradient that are used to adapt the learning rate for each weight of the DL model/neural network. Whereas the moment is considered as the expected value of a variable to the power of n. The first ( $mdw$ ,  $mdb$ ) and second moment ( $vdw$ ,  $vdb$ ) estimates are evaluated by equations (21)-(24).

$$mdw_t = \beta_1 * mdw_{t-1} + (1 - \beta_1) * dw_t \quad (21)$$

$$mdb_t = \beta_1 * mdb_{t-1} + (1 - \beta_1) * db_t \quad (22)$$

$$vdw_t = \beta_2 * vdw_{t-1} + (1 - \beta_2) * dw_t^2 \quad (23)$$

$$vdb_t = \beta_2 * vdb_{t-1} + (1 - \beta_2) * db_t^2 \quad (24)$$

The bias-corrected first ( $mdw^{\hat{}} mdb^{\hat{}}$ ) and second ( $vdw^{\hat{}} vdb^{\hat{}}$ ) moment estimates are evaluated by equations (20)-(23).

$$mdw^{\hat{}} t = mdw_t / (1 - \beta_1^t) \quad (25)$$

$$mdb^{\hat{}} t = mdb_t / (1 - \beta_1^t) \quad (26)$$

$$vdw^{\hat{}} t = vdw_t / (1 - \beta_2^t) \quad (27)$$

$$vdb^{\hat{}} t = vdb_t / (1 - \beta_2^t) \quad (28)$$

Then, the weights and biases are evaluated by:

$$W_t = W_{t-1} - lr * \frac{mdw^{\hat{}} t}{\sqrt{vdw^{\hat{}} t} + \epsilon} \quad (29)$$

$$b_t = b_{t-1} - lr * \frac{mdb^{\hat{}} t}{\sqrt{vdb^{\hat{}} t} + \epsilon} \quad (30)$$

where  $\epsilon$  is equal to  $10^{-8}$

### 4) SELECTION OF HYPERPARAMETERS

The hyperparameter values are selected using the random search method [45], presented in Table 4. For example, the learning rate ( $lr$ ) of the SGD optimizer to train the RFCN model was tuned exponentially from  $10^{-5}$  to  $10^{-1}$ , while the momentum ( $mom$ ) was tuned with a difference of 0.1. The hyperparameter tuning was started with  $lr$  of  $10^{-1}$  and zero mom, the RFCN did not get the training convergence. Then, the  $lr$  was started to reduce and mom was increased. The training of the RFCN model started to settle down. For example, at  $lr$  of  $10^{-3}$  and  $mom$  of 0.8, the mAP was 61.60% with a total training loss of around 0.41%. A further reduction in the  $lr$  positively influenced the performance of the RFCN. At the learning rate of  $10^{-4}$  and mom of 0.8, the training loss was reduced to 0.23% with the mAP of 73.256%. But a further increase in  $lr$  ( $10^{-5}$ ) significantly increased the training time. Therefore, small random changes were made for  $lr$  and  $mom$  and the performance of the RFCN model was checked in various values. It was found that  $lr$  of in  $3 \times 10^{-4}$  and  $mom$  of 0.9, the loss was reduced to around 0.09% and the mAP improved significantly to 74.47%.

**TABLE 4. Hyperparameters of deep learning optimizers.**

DL optimizers	DL meta-architectures	Hyperparameters
SGD with momentum	Faster RCNN ResNet-50	$lr = 2 \times 10^{-4}$ , $mom = 0.9$
	Faster RCNN Inception-v2	$lr = 1 \times 10^{-4}$ , $mom = 0.9$
	EfficientDet	$lr = 3 \times 10^{-4}$ , $mom = 0.9$
	SSD Inception-v2	$lr = 3 \times 10^{-4}$ , $mom = 0.9$
	SSD MobileNet-v2	$lr = 2 \times 10^{-4}$ , $mom = 0.9$
	SSD ResNet-50 (RetinaNet)	$lr = 4 \times 10^{-4}$ , $mom = 0.9$
	Faster RCNN Inception ResNet-v2	$lr = 2 \times 10^{-3}$ , $mom = 0.9$
SGD with momentum		$lr = 3 \times 10^{-4}$ , $mom = 0.9$
RMSProp	Faster RCNN ResNet-101	$lr = 2.5 \times 10^{-4}$ , $rho = 0.9$ , $mom = 0.9$ , $eps = 1 \times 10^{-3}$
Adam		$lr = 1 \times 10^{-4}$ , $eps = 1.0^{-2}$
SGD with momentum		$lr = 3 \times 10^{-4}$ , $mom = 0.9$
RMSProp	RFCN ResNet-101	$lr = 3 \times 10^{-4}$ , $rho = 0.9$ , $mom = 0.9$ , $eps = 1 \times 10^{-3}$
Adam		$lr = 2 \times 10^{-4}$ , $eps = 1.0^{-3}$

$lr$ : learning rate,  $mom$ : momentum,  $eps$ : epsilon

## J. VALIDATION METHODS

### 1) STRATIFIED FIVE-FOLD CROSS-VALIDATION

The proposed DL-based approach has been validated using two techniques. First, owing to the class imbalance problem due to the different number of images of each class (can be seen in Table 2), a stratified cross-validation method is used. This method retains the particular number of data points/sample size of each class in each fold [46]. It ensures the unbiased distribution of the dataset among all folds. Otherwise, random sampling could generate bias in the folds when all dataset images are randomly shuffled and split into a certain number of folds.

### 2) TESTING ON AN EXTERNAL DATASET

Another contribution of this study is the validation of the final results using an external test dataset (obtained by a random search on various websites). This was done to show the effectiveness and robustness of the work that presented DL-based approach would also be applicable under different environmental conditions than the one used for dataset generation.

## III. RESULTS AND DISCUSSIONS

After the dataset generation, the proposed approach is divided into several steps to get the optimized DL model for plant disease detection. The results presented in this section follow the methodology of the research, as shown in Fig. 1. First, a comparison of the DL architectures was performed. This step was performed to obtain the top two models. The training and validation plots are presented (to understand the performance of several DL models) and the detection results (to evaluate the mAP). The best two DL models were trained with all data augmentation methods to understand their effects. Later, the effects of image resizing techniques and interpolators are provided in terms of training, validation losses and mAP. These methods evaluated the impact of the input image on the DL model. Afterward, performance optimization has been explained by weight initializers, batch normalization, and DL optimizers. The effects of various parameters of the weight initializers are also provided. Similarly, the performance of the best-obtained DL model was evaluated in the presence of batch normalization, to show the better convergence ability of the DL model. DL optimizers are also compared to optimize the weights of the best-obtained DL architecture.

After the optimization of the DL model, further in-depth class-wise analysis has been performed. In this regard, the performance of the individual classes is evaluated. The classes that attained the lowest AP were explicitly focused. This step also aimed to maintain the high AP of the other classes obtained in the previous step. The position-sensitive score maps are analyzed, as it was one of the major novelties of the RFCN model (the best-obtained model). The detection results are shown to understand the impacts of the spatial bin width and heights of the score maps. Furthermore, the anchor boxes were enhanced to show the influence of various anchor

box scales and aspect ratios. The results are shown by the training and validation plots and average precision of each class, along with the mAP of each enhanced version. Finally, the stratified k-fold cross-validation method was used due to the class imbalance problem in the proposed dataset and to validate the final mAP of the optimized DL model.

## A. COMPARISON BETWEEN DL ARCHITECTURES

First, the DL models are trained on the original (without augmentation) images. Subsequently, the two best models are retrained on the augmented images. It is empirically found that DL architectures should be trained to 200K steps to achieve training convergence. The input images are resized to  $300 \times 300$  pixels with fixed image resizers for SSD MobileNet-v2 and SSD Inception-v2 and  $640 \times 640$  pixels for SSD ResNet-50 (RetinaNet). An aspect ratio resizer with minimum and maximum pixel dimensions of 600 and 1000, respectively, is considered for the models including all versions of Faster R-CNN and RFCN. The EfficientDet model is also trained with an aspect ratio resizer with 512 minimum and maximum pixels, according to the GPU requirement. Furthermore, SGD with a momentum optimizer is used to train the models for this stage of the research. Different batch sizes are tested, and the most reasonable is found to be 4 to reduce the trade-off between accuracy and training time. Four models required the lowest iteration steps of 170K to achieve training convergence: Faster RCNN ResNet-50, RFCN ResNet-101, EfficientDet, and RetinaNet. The lowest training times are obtained for SSD MobileNet around 5.5 h. However, the Faster RCNN Inception ResNet-v2 required the highest time to complete the training. The following observations are made on the training and testing performance of the DL models.

### 1) TRAINING PERFORMANCE

- Plots of the total training and validation losses for each model are shown in Fig. 7. It can be observed that the Faster R-CNN ResNet-101 and RFCN ResNet-101 models attained the lowest total training and validation losses of approximately 0.05-0.08%, 0.06-0.09%, and 0.04-0.2%, 0.06-0.18% respectively. Both models took around 10.5 hours to achieve convergence.
- However, the versions of SSD models with Inception-v2 and MobileNet-v2 have approximately 1.5% total loss. This is comparatively higher than that of the other DL models, apparently reflected in their detection results as a low mAP as shown in Table 5.
- Later, the two best models are retrained with augmented images due to their lowest training and validation losses (after training on non-augmented images), including RFCN ResNet-101 and Faster RCNN ResNet-101. Their loss plots are shown in Fig. 8. It can be concluded from the plots that RFCN ResNet-101 has a slightly lower training and validation losses of approximately 0.7% and 1.0%, respectively.

**TABLE 5.** Summary of the plant disease detection results by the DL architectures in terms of average precision (in %) of each class divided into various augmentation techniques.

Healthy and disease classes	DL architectures with backbone models																	
	Faster RCNN		Effici entDet		SSD			Faster RCNN					RFCN					
	ResN et-50	Incep ion-v2	Effici entNet	Incep tion-v2	Mobil eNet-v2	ResNet-50 (Retina Net)	Incep tionResNet-v2	OO	OO	OC	OT	OCN	OTCN	OO	OC	OT	OCN	OTCN
	OO	OO	OO	OO	OO	OO	OO	OO	OO	OC	OT	OCN	OTCN	OO	OC	OT	OCN	OTCN
A_blk_rot_f	62.77	64.28	0	18.18	33.33	0	63.73	64.5	74.3	80.66	80.1	72.61	65.41	76.72	71.77	39.63	77.17	
A_blk_spot	3.54	2.14	0	1.44	4.95	0	10.82	11.31	23.6	9	9.01	0	0	16.15	5	5.18	5	8
A_e_canker	81.57	39.58	49.09	0	18.18	49.09	67.8	33.5	33.3	51.55	85.34	83.4	40.27	80.19	68.83	6.01	95.78	
A_gl_lf_spot	73.91	82.27	44.55	41.67	51.37	0	78	82.03	72.4	79.67	87.7	87.99	91.98	92.86	92.24	62.08	97.12	
A_healthy_f	94.07	95	42.83	90.91	100	0	90.55	92.75	100	100	90.24	80.08	100	95.91	99.92	100	88.88	
A_healthy_l	49.21	61.6	40	63.07	43.15	49.09	80.01	62.55	47.1	74.65	80.99	90.24	63.07	86.82	65.34	56.53	99.97	
A_m_virus	28.04	33.19	43.9	32.78	32.91	47.27	38.74	42.18	58.4	45.11	7.57	7.25	40.97	31.07	65.77	33.54	9.19	
Av_alg_if_spot	70.6	70.32	0	90.26	88.11	0	58.15	77.62	26.5	71.57	12.51	5.23	75.62	9.55	63.93	10.88	8.26	
Av_br_canker	53.59	70.67	0	32.66	22.98	0	77.69	55.88	77.3	80.92	19.31	6.31	76.21	54.48	99.8	46.5	8.94	
Av_healthy_i	65.41	75.17	44.55	43.54	32.07	44.55	71.91	87.46	90.3	100	0.05	0	79.47	5.19	100	57.7	8	
G_blk_spot_c	76.81	76.82	0	42.46	0	0	61.36	56.55	29.9	53.12	3.28	1.04	71.27	15.85	88.39	33.26	8	
G_healthy_c	94.43	92.35	0	90.91	81.82	0	89.32	100	100	100	99.88	67.84	100	95.91	99.84	95.91	63.63	
Kf_bac_canker	95	95	47.74	100	64.69	56.56	91	92.91	76.2	100	90.73	69.66	99.52	95.91	100	99.78	57.44	
Kf_healthy_i	95	95	65.04	100	100	50	91	100	89.2	100	72.74	7.14	100	85.16	100	95.91	8.58	
P_canker	87.23	78.23	40	36.36	61.74	58.18	86.8	81.59	82.9	89.94	42.87	7.54	82.53	55.76	97.56	99.76	8	
P_fr_blight	34.45	57.91	0	36.36	36.36	0	53.34	62.83	53.1	43.28	53.69	43.11	81.08	76.23	24.16	100	47.79	
P_healthy_f	95	95	0.83	100	100	0	91	100	100	92.91	100	90.91	100	100	95.91	100	98.91	
P_healthy_l	95	95	67.27	100	100	58.18	91	100	23.7	48.54	99.96	88.97	100	99.68	24.37	24.55	81.83	
P_s_pit	66.7	61.52	0	27.27	57.77	0	87.38	74.73	97.7	82.76	78.67	99.01	50.6	94.76	98.85	67.49	98.19	
P_scab	45.88	49.79	58.18	26.26	26.33	49.09	49.96	66.99	0.29	59.62	56.72	1.85	55.34	45.65	73.26	67.22	13.78	
mAP (%)	68.41	69.54	27.20	53.71	52.79	23.10	71.48	72.27	62.8	73.16	58.12	45.51	74.47	65.13	76.76	60.09	49.37	

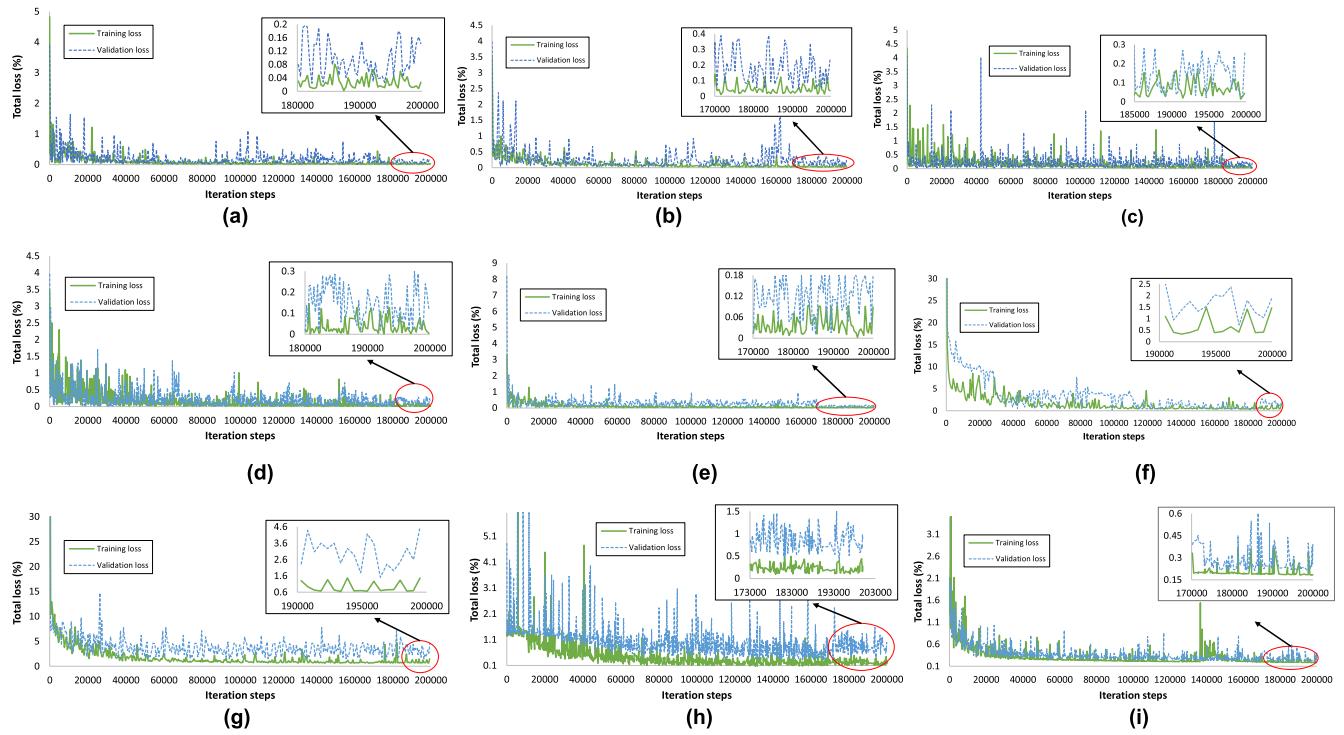
- It can be noticed from Figs. 7-8 that the DL models did not suffer from overfitting. Both training and validation losses were settled down with a small fluctuation and there was no such sudden increase in the validation loss after achieving the convergence in both the absence and presence of the augmented images.

## 2) TESTING PERFORMANCE

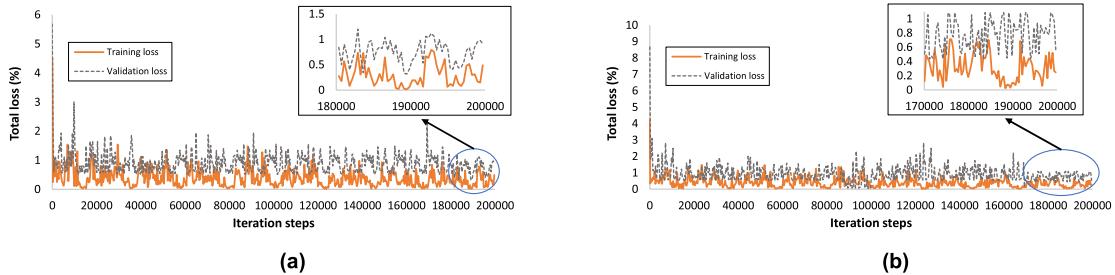
- The results obtained from the test dataset (without augmentation) revealed that the RFCN ResNet-101 model achieved the highest mAP of 74.47%, followed by Faster RCNN ResNet-101 and Faster RCNN Inception ResNet-v2, as shown in Table 5. This is because RFCN achieved a high AP of 10 healthy/disease classes. A sample of each class is shown in Fig. 9 (a).
- The Faster RCNN trained with Inception ResNet-v2 and ResNet-101 attained a higher mAP than the rest of the models, including Faster RCNN ResNet-50, Faster RCNN Inception-v2, SSD models, RetinaNet and EfficientDet. The Faster R-CNN ResNet-101 is found to be the most useful model for the healthy class of avocado (Av\_healthy\_l) and attained the highest AP among all DL models. It is also noticed that some of the testing

images of apple glomerella leaf spot and pear fire blight obtained false positive and false negative detections, respectively, with Faster R-CNN ResNet-101, as shown in Fig. 9 (b). Similarly, classes such as apple healthy leaves and pear stony pits are well detected using Faster Inception ResNet-v2, as shown in Fig. 9 (c-d).

- Although the RFCN model attained the highest mAP, it misclassified some of the testing images of the classes, such as the stony pit on the pear, as shown in Fig. 9 (d).
- The testing performance of the models, including EfficientDet and RetinaNet, is unsatisfactory. This was due to several classes remaining undetected and giving false positive results, as presented in Fig. 9 (e).
- It can also be seen from Table 5 that the black spot on the apple leaves failed to be detected and localized by all DL models, an example from each DL model is presented in Fig. 9 (f). Some of the classes attained 0% average precision when trained by the models like EfficientDet, SSD Inception-v2, SSD MobileNet-v2, and SSD ResNet-50. Because these models failed to detect a few of the classes, that was observed in two ways. First, the testing images of those classes were undetected, second, the false positive results were obtained due to the



**FIGURE 7.** Total loss plot for deep learning models before augmentation: (a) Faster RCNN ResNet-101, (b) Faster RCNN ResNet-50, (c) Faster RCNN Inception-v2, (d) Faster RCNN Inception ResNet-v2, (e) RFCN ResNet-101, (f) SSD MobileNet-v2, (g) SSD Inception-v2, (h) EfficientDet, (i) RetinaNet.



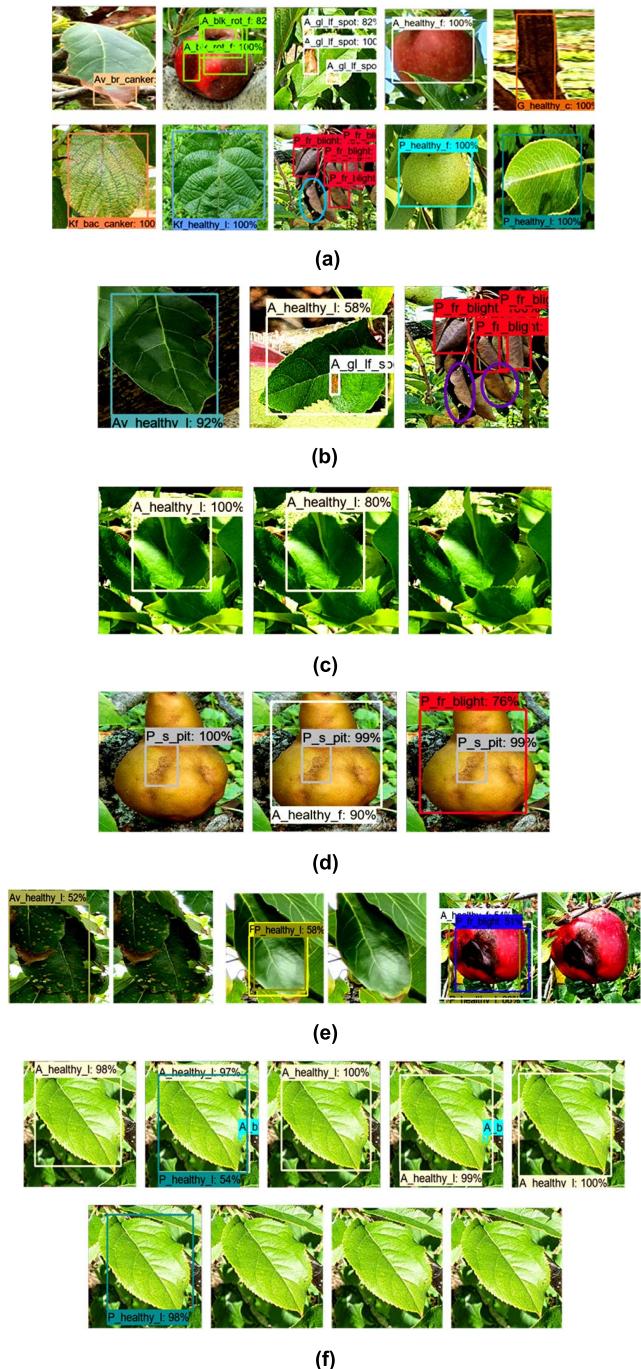
**FIGURE 8.** Total loss plot for deep learning models after the application of all augmentation techniques: (a) Faster RCNN ResNet-101, (b) RFCN.

confusion with another plant disease/healthy classes. In Fig. 9 (f), an example of the undetected/false negative outcome for apple black spot is presented for EfficientDet and RetinaNet.

- Further analysis was required to validate the selection of the best DL model for the next phase of research.
- In this regard, the top two DL models (RFCN ResNet-101 and Faster RCNN ResNet-101) in terms of the lowest training, validation losses and the highest mAP are retrained using augmented images (considering all 13 augmentation categories).
- RFCN ResNet-101 has achieved a higher mAP than Faster RCNN ResNet-101, as shown in Table 5.
- Some classes, including apple glomerella leaf spot, apple European canker, and apple healthy classes, with the RFCN model, have achieved a higher AP compared

to the Faster RCNN ResNet-101. A few examples from RFCN are shown in Fig. 10.

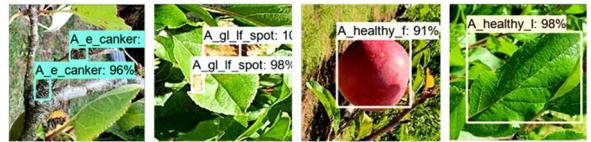
- Classes such as apple black rot, apple European canker, and healthy leaves of apples have improved their AP after training the RFCN through augmented images. On the contrary, the AP of almost 13 classes is significantly reduced by training in the augmented images, as shown in Table 5.
- The main finding of this step is that the RFCN model has achieved the highest mAP with and without augmented images. Another important observation is that the augmented images helped improve the AP of only a few classes. Moreover, the RFCN has shown its ability to address problems such as the detection of plant diseases in different organs and the identification of disease in different fruits, as shown in Figs. 9–10. However, the



**FIGURE 9.** Detection results by various DL models trained without augmented images. (a) true positive and false negative ( $P_{fr\_blight}$ ) by RFCN, (b) true positive ( $Av\_healthy\_I$ ), false positive ( $A\_gl\_if\_spot$ ), and false negative results ( $P_{fr\_blight}$ ) with Faster RCNN ResNet-101, (c) true positive outcome for  $A\_healthy\_I$  class with Faster RCNN Inception ResNet-v2, Faster RCNN ResNet-101, and false negative results for RFCN (first from the right), (d)  $P_{s\_pit}$  with Faster RCNN Inception ResNet-v2, Faster ResNet-101, and RFCN (left to right), (e) false positive and false negative results with EfficientDet and RetinaNet, (f) false positive for  $A\_blk\_spot$  by RFCN, Faster RCNN ResNet-101, Faster RCNN Inception ResNet-v2, Faster RCNN ResNet-50, and Faster RCNN Inception-v2, (first row – left to right), and false positive, false negatives by SSD MobileNet-v2, SSD Inception-v2, EfficientDet, and RetinaNet (second row – left to right).

low AP of several classes was attained in the presence of augmentation techniques. This motivated us to

individually evaluate the effects of the augmentation techniques. This helped to understand the reason for performance degradation after the application of 13 types of augmentation methods for this step of the study.



**FIGURE 10.** True positive results of RFCN after training on augmented images.

## B. EFFECTS OF DATA AUGMENTATION TECHNIQUES

The effects of data augmentation methods are studied by dividing them into five categories as described in Section II-B (5). The two top DL models are trained on all five groups of augmented images, including RFCN ResNet-101 and Faster R-CNN ResNet-101. The important results from this phase of the study are discussed in the following.

- The RFCN model has achieved the highest mAP after training with the OT data augmentation category, followed by the results obtained through the OO images, as shown in Table 5. However, comparatively lower mAP values are observed for OC and OCN. Whereas, RFCN has achieved the lowest mAP with the OTCN.
- To further perform an in-depth analysis of the data augmentation techniques, a class-wise analysis is performed. For example, OT is found to be the best method due to its higher AP by the RFCN in eight classes of healthy individuals and diseases. Therefore, the OT category has attained superior results.
- The effectiveness of the OT category is also validated using Faster RCNN ResNet-101, as shown in Table 5. The mAP is higher in the OT group than in all other categories. However, with the OTCN, the model has achieved the lowest mAP value.
- There could be several reasons for performance degradation when training with OC and OCN. The nature of the real agricultural environment could contribute to the confusion in discriminating between plant diseases. Because the real field contains several background elements, a change in color or addition of noise to the original images distracts/fails the model to extract and learn the specific and distinct features of the disease symptoms. There may be similarities between the symptoms of the disease and background elements [47] after adding noise and changes in brightness, contrast, and sharpness. This has resulted in a low AP for the individual classes and a low mAP for the 20 classes.
- The above statement can be further understood by taking examples of some classes that achieved comparatively lower AP. For example, avocado branch canker is confused with apple glomerella leaf spot, healthy avocado

leaf with healthy pear leaves, and apple mosaic virus with healthy apple healthy leaves. Similarly, the black spot-on grapevine cane could not be detected and/or is misclassified as an apple European canker. The stony pit virus on the pear is also confused with fire blight. Furthermore, in avocado leaves, several algal leaf spots are not detected. Some examples of false-positives and undetected outcomes are shown in Fig. 11.

- In contrast, the improved performance with OT has shown the practical aspect of this work. Because the location of disease spots in a real agricultural environment varies from one plant to another. Furthermore, if a DL model such as RFCN can detect the disease in the translated and rotated images, it shows the importance of the OT-based data augmentation technique for the identification of plant diseases.
- Fig. 11 showed the significance of the original with translational changes (OT-based augmentation method) by taking examples of five plant classes. These classes consist of diseases in plant organs including leaves (apple and avocado), fruit (pear), and cane (grapevine), including avocado branch canker, avocado healthy leaves, apple mosaic virus, grapevine black spot, and pear stonypit. An improvement of 23.59, 20.53, 24.8, and 17.12%, and 48.25%, respectively, was attained as compared to the previous step.

### C. EFFECTS OF IMAGE RESIZERS AND INTERPOLATORS

The next step is to study the effects of image resizers with different interpolators on RFCN. In this regard, the RFCN model is initially trained using an aspect ratio resizer and a bilinear interpolator. Eight possible combinations of image resizers and interpolation methods are used to obtain the best results, as shown in Table 6. The performance of each combination was not only evaluated by mAP, training, and validation losses, but also by the training time as considered in [48]. The main observations and discussions of this step of the analysis are presented below.

- The aspect ratio is considered the default resizing technique with minimum and maximum dimensions of 600 and 1000 pixels, respectively [35]. This image resizing technique is used in conjunction with bilinear interpolation.
- As presented in Table 6, the aspect ratio resizer with the other three interpolators, such as bicubic, area, and nearest neighbor, has degraded the performance of RFCN in terms of lower mAP.
- Later, a fixed-shape resizer with a default value of  $300 \times 300$  pixels is applied with bilinear interpolation. Subsequently, three other interpolators are tested. The RFCN model trained with the bicubic interpolator with fixed shape resizer required the lowest training time, training loss (0.52%) and validation loss (0.85%) as shown in Table 6. Both losses were lower than the loss acquired with the default resizer/interpolator, as shown in Fig. 12.



**FIGURE 11.** Detection outcomes of RFCN model by three augmentation categories. True positive, false positive, and false-negative results from left to right in each example belong to OT, OC, and OCN respectively.

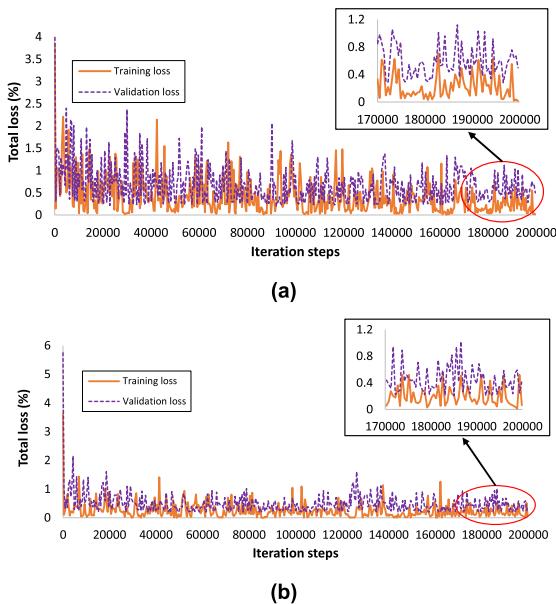
- There is no such sign of overfitting of the RFCN model as the validation loss was also settled to the final value and no such increase in the loss was observed. It is validated with a higher mAP of 80.59%, which is 3.83% better than that obtained using the default settings. Furthermore, the AP of avocado algal leaf spot, pear fire blight, and healthy pear leaves are significantly improved to 88.55%, 98.16%, and 49.9%, respectively.
- The bilinear interpolator with a fixed-shape resizer has also performed slightly better than the nearest neighbor and area interpolation.
- Bicubic interpolation considers a  $4 \times 4$  or 16-pixel square and evaluates the resulting interpolated pixels, compared to  $2 \times 2$  pixels for bilinear interpolation. Therefore, better-quality images are obtained for the healthy and disease classes to be fed into the RFCN model, producing a better mAP.
- In conclusion, both the training and testing performances of the RFCN model are improved with an enhancement in the AP of the three classes (after training with a fixed-shape resizer along with a bicubic interpolator). In the future, other relevant datasets can be tested using the combinations of resizers and interpolators presented in this phase of the research.

### D. EFFECTS OF WEIGHT INITIALIZERS, BATCH NORMALIZATION, AND DL OPTIMIZERS

The next phase of the proposed approach is the optimization of the RFCN model. The appropriate selection of weight

**TABLE 6.** Effects of image resizing techniques and interpolators on the RFCN ResNet-101 model.

Image resizing techniques/interpolators	Aspect ratio				Fixed Shape			
	Bilinear	Bicubic	Area	Nearest neighbor	Bilinear	Bicubic	Area	Nearest neighbor
Training loss (%)	0.60	0.625	0.62	0.61	0.55	0.52	0.56	0.56
Validation loss (%)	1.1	1.35	1.15	1.2	0.83	0.85	0.88	0.91
Training time (sec/steps)	0.189	0.189	0.195	0.195	0.141	0.138	0.147	0.147
mAP (%)	76.76	70.06	71.96	70.62	75.59	80.59	74.33	74.63

**FIGURE 12.** Total loss plots from two different image resizers/interpolators: (a) aspect ratio resizer with bilinear interpolator, (b) fixed shape resizer with bicubic interpolator.

initializers along with their parameters solves the problems of vanishing and exploding gradient descent. Three weight initialization methods are used: truncated normal (default initializer), scaling variance, and random normal. After determining the most suitable initialization technique, the effects of batch normalization are studied. Then, the best DL optimizer is selected, and the hyperparameters are tuned using a random search method [45]. These steps are performed before proposing any modifications to the RFCN model. The training and testing evaluations of this phase of the study are summarized below.

- First, the RFCN model is trained using a truncated normal initializer. Subsequently, it is trained using the scaling variance and random normal initializers.
- The random normal initializer has achieved the lowest training loss and the highest mAP with a standard deviation of 0.01 and a mean value of zero. It has been observed that the selection of an optimum value for standard deviation and mean plays an important role in the performance of the model.
- When searching for the appropriate values of standard deviation and mean, the random normal initializer was initially used with a default standard deviation of 1 and

a mean of zero. However, these values were unable to achieve convergence. Therefore, the standard deviation started to tune exponentially. It was empirically found that 0.01 was the most suitable standard deviation and the mean value of zero remained the same. This was due to the lowest training and validation losses, which resulted in a slight improvement of 0.916% in the mAP. Furthermore, the extraction of distinct features was also improved, as the AP of some of the classes was enhanced with the described settings. These classes include apple black rot, apple European canker, apple glomerella leaf spot, apple mosaic virus, and healthy pear (leaves), with an improvement of 8% to 28%. Similar standard deviations and mean values were also suitable for the truncated normal initializer.

- The best settings for scaling variance were single scaling factors with a uniform distribution and considering the average of the input and output units in the weight tensor (Fan\_avg). However, these parameters did not contribute to improving the model performance and attained an mAP of only 70.20%.
- The random normal initializer has produced the best result for a particular application of the detection of plant disease by the RFCN model. Therefore, it can be concluded from its basic functionality that the initialized weights through the generation of tensors with normal distribution performed well for the selected problem. Moreover, theoretically, the random normal is supposed to work with weights initialized very close to zero. So, each neuron of the network does not perform the same calculation.
- It was experimentally found that a small standard deviation value was not suitable. For example, at a standard deviation of 0.001, the performance declined in terms of mAP to 78.33%, compared to mAP at 81.50%, obtained with a 0.01 standard deviation.
- The next step was the use of batch normalization. The RFCN is trained with the default values of epsilon and decay of 0.001 and 0.99, respectively. It is also noted that the training convergence is achieved earlier to around 160K steps from around 170K iterations. Therefore, it can be concluded that batch normalization reduced the overall training time and showed a fast convergence ability [49].
- The decay and epsilon were started to tune, and it was experimentally found that the lower value of decay and the higher value of epsilon improved the performance

of RFCN. Therefore, the decay was set to 0.5 and the epsilon was 0.01. The training performance was slightly improved to around 0.515%.

- The testing performance of the RFCN model was also significantly improved. The model attained an mAP of 85.94% with an improvement of 4.345% compared to the one obtained in the previous step.
- The individual AP of several classes were improved such as apple black spot, apple healthy (leaves), apple mosaic virus, grapevine black spot, pear fire blight, and pear stony pit at 61.66%, 100%, 96.43%, 98.53%, 98.45%, and 96.66%.
- The last step before proposing any modification to the RFCN model is the utilization of different DL optimizers. SGD (with momentum) is used to train the model as the default DL optimizer. Subsequently, its performance is compared with that of Adam and RMSProp.
- After training the RFCN model using all three DL optimizers, it is found that SGD with momentum is the best DL optimization algorithm. Adam optimizer is unable to achieve a high mAP. Therefore, it did not effectively optimize the weights of the RFCN model. However, RMSProp has also achieved a lower mAP of 82.819%.
- Individual APs of several classes, including apple black rot, apple black spot, and pear canker are degraded by RMSProp.
- The best performance of the SGD optimizer demonstrates its generalizability in extracting the features of the healthy and disease classes and optimizing the weights of the RFCN. Therefore, it can be summarized that, for the NZDLPlantDisease-v1 dataset of healthy and diseased plant organs, the non-adaptive optimizer - SGD with momentum was quite successful compared to the adaptive optimization techniques RMSProp and Adam.
- To address one of the research gaps presented in the previous section, the RFCN model trained with a random normal initializer, using batch normalization and SGD with momentum optimizer is also successful in identifying multiple disease problems and detection of plant disease in different weather conditions (sunny, cloudy), as shown in Fig. 13 and Fig. 14, respectively.

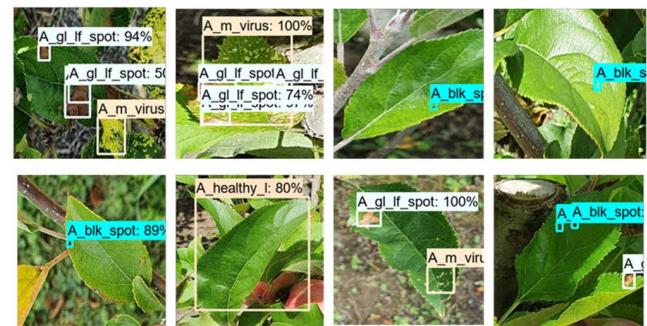
Another important observation is that the pear scab class still attained a low AP of 5.06%. Although all steps until the application of various DL optimizers significantly improved the performance of RFCN in terms of a lower training and validation losses compared to the default configurations of the model. Still, the pear scab remained almost undetected or falsely identified. This result has provided a strong basis for the next steps of the research to focus on the architectural evaluation/modifications of the RFCN model.

#### E. PERFORMANCE ENHANCEMENT OF PEAR SCAB

There are two major goals for this step of the research. First, an improvement in the AP of pear scab, which is undetected after the application of several techniques explained earlier.



**FIGURE 13.** Examples of multiclass plant disease detection after the application of random normal initializer and batch normalization for the RFCN model.

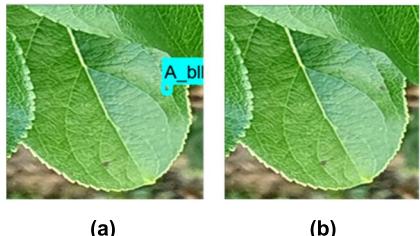


**FIGURE 14.** Examples of apple plant disease/healthy leaves in different environmental conditions.

Second, the high AP of the other 19 healthy/disease classes should be maintained. In this regard, the RFCN model has been investigated in two stages: position-sensitive score maps and enhanced/modified anchor box scale and aspect ratio.

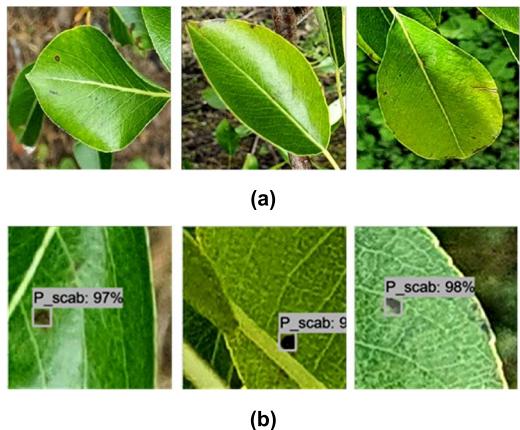
One of the primary novelties of the RFCN model is the generation of position-sensitive score maps. The spatial bin configuration was set to  $3 \times 3$  by default. It was empirically observed that the spots of pear scab were so small that the model could not extract its features and therefore could not be detected. This might be because none of the sub-regions of positive-sensitive score maps could match the pear scab for most of the testing images. Therefore, the position-sensitive region of interest (RoI) pool cannot vote for pear scab disease. In this regard, the first attempt is to increase the score maps using multiples of 3. The width and height of the  $9 \times 9$  spatial grid have yielded satisfactory results and attained an mAP of 84.68%. Otherwise, with other spatial bins, such as  $6 \times 6$ ,  $12 \times 12$ , and  $15 \times 15$ , a lower mAP of 82.819%, 82.041% and 82.59%, respectively are observed with the AP of the pear scab of 5.03%, 5.05%, and 5.28%. However, there is a slight difference in the total training and validation losses with  $9 \times 9$  grids and the model has detected pear scab with an AP of 20.1%. Still, the DL model could detect the pear scab with a high AP. Furthermore, the RFCN trained with  $9 \times 9$  spatial bins has disrupted the detection of apple black spots and achieved a lower AP of 49.69%. An example of the detection of apple black spot is presented in Fig. 15.

Another attempt has been made to solve this problem. The training images of the pear scab are magnified, and the RFCN model has been trained again. This is one of the ways to overlap the pear scab with  $3 \times 3$  score maps. RFCN has successfully detected and localized both the apple black spot



**FIGURE 15.** Detection outcomes for apple black spot using different positive sensitive score maps. (a) true positive results of apple black spot with  $3 \times 3$  spatial bins, (b) false-negative detection with  $9 \times 9$  spatial bins.

and the pear scab simultaneously. This indicates that the specific features of both disease classes are well extracted. The mAP is 87.394%, with the individual AP of the apple black spot and the pear scab at 59.85% and 94.43%, respectively. Few samples of the pear scab results are shown in Fig. 16. Moreover, a high AP of the rest of the classes is also maintained.



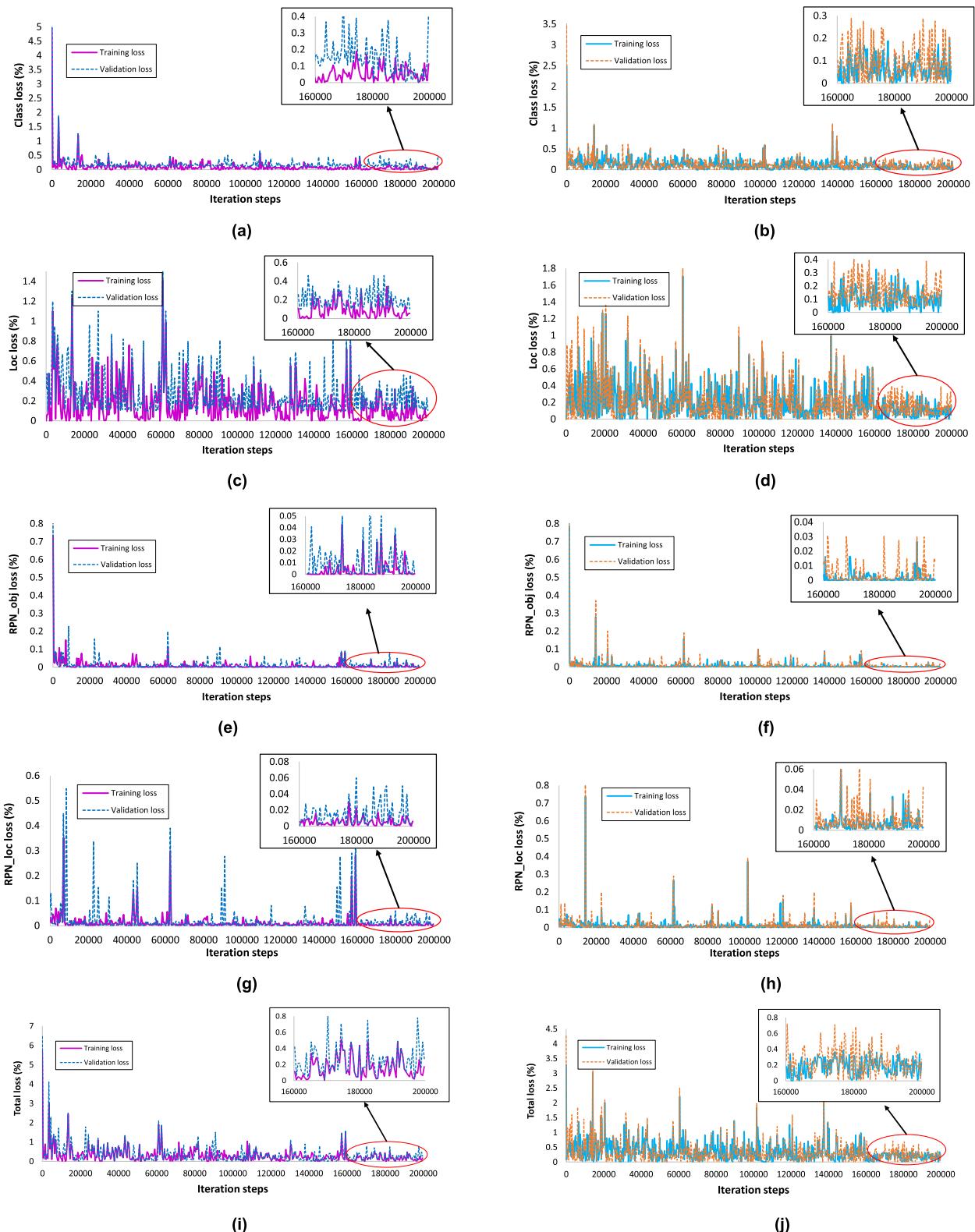
**FIGURE 16.** (a) Examples of false-negative results before zooming in the pear scab with  $3 \times 3$  spatial bins (b) Examples of true positive results after zooming in the pear scab with  $3 \times 3$  spatial bins

#### F. ENHANCEMENT OF ANCHOR BOXES

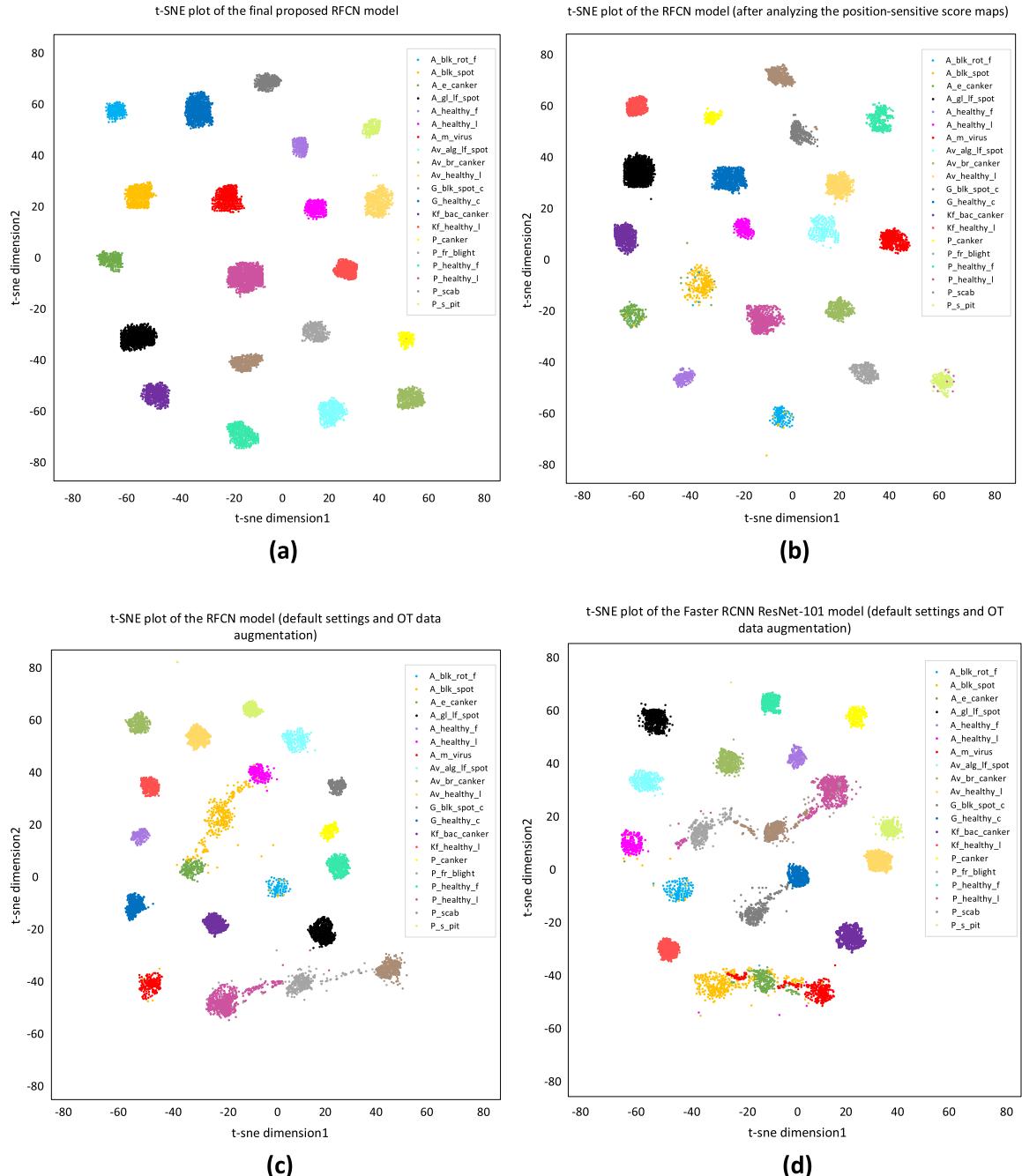
After a significant improvement in the average precision of the pear scab has been achieved in the previous step, the anchor boxes of the RFCN are enhanced. In this regard, the scale size and aspect ratios are modified to obtain an optimum anchor box that can provide an AP of more than 80% for each class. The summary of the results is as follows.

- Although the previous step has considerably improved the AP of the pear scab, a few classes also required attention towards further performance enhancement. For example, classes such as apple black rot, apple black spot, apple European canker, and pear healthy (leaves) have achieved an AP of less than 80%. During the annotation of the training images, it was empirically observed that the bounding-box coordinates of several classes varied. Therefore, different scales are tested to generate the anchor boxes.

- Hence, the addition of scales such as  $64 \times 64$ ,  $32 \times 32$ , and  $16 \times 16$  with the default scale sizes such as  $128 \times 128$ ,  $256 \times 256$ , and  $512 \times 512$ , has significantly improved the mAP.
- On the other hand, an addition of a very small scale size such as  $8 \times 8$  reduced the mAP. Similarly, a very large scale size of  $1024 \times 1024$  could not contribute to attaining better mAP.
- Next, reciprocal aspect ratios are applied and the default aspect ratios of 1:2 and 2:1 are replaced. It is found that 1:4 and 4:1 have achieved an almost similar result with mAP of 87.33% with scale sizes of  $16 \times 16$ ,  $32 \times 32$ ,  $64 \times 64$ ,  $128 \times 128$ ,  $256 \times 256$ , and  $512 \times 512$ . Otherwise, none of the other combinations of reciprocal aspect ratios has shown noticeable results.
- Subsequently, the effects of the step-by-step/gradual enhancement of the aspect ratio are studied. In this regard, a small aspect ratio was started to add from 1:4 to default ratios of 1:2, 1:1, and 2:1, and enhanced scale sizes. After several experiments, it is found that the addition of the aspect ratio like 1:2, 1:1, 2:1, 3:1, and 4:1, has improved the training and testing performance of the model. The total training and validation loss from 0.4-0.515% and 0.4-0.8% have been reduced to almost 0.3-0.37% and 0.4-0.71%, respectively. Furthermore, the individual loss of box classifier localization loss was reduced to almost 0.2% from 0.3%, as shown in Fig. 17. There is no sign of overfitting as the losses were converged, no abrupt rise in the validation loss was observed after the final iteration step, and there was a small difference between both training and validation losses.
- The feature extraction of the healthy and disease plant classes has been presented by t-distributed stochastic neighbor embedding (t-SNE) plots in Fig. 18. It can be seen for each of the healthy/defective classes trained by the final RFCN model that there is a high inter-class distance separability, a small intraclass distance and grouped clusters have been created which were concentrated in their respective features. Furthermore, the effectiveness of the proposed modifications has been presented by comparing the t-SNE plot for the previous step and the default settings of the RFCN and Faster RCNN model after the application of the OT data augmentation method.
- It can be observed in Fig. 18 (b) that after the application of several techniques (presented in section III C-E), some of the features of the classes such as apple black rot, apple black spot, apple European canker, and pear healthy leaves were not well extracted and confused with the features of other apple and pear classes. Similarly, the t-SNE plots by the RFCN model after the OT data augmentation technique (Fig. 18 (c)) attained comparatively small interclass distances. There were several features of the apple black spot, pear fire blight, and pear healthy leaves, were not extracted and



**FIGURE 17.** Training and validation losses for different anchor box configurations (a) box classifier classification loss with default scales and aspect ratio, (b) box classifier classification loss with modified anchor boxes, (c) box classifier localization loss with default scales and aspect ratio, (d) box classifier localization loss with modified anchor boxes, (e) RPN objectness loss with default scales and aspect ratio, (f) RPN objectness loss with modified anchor boxes, (g) RPN localization loss with default scales and aspect ratio, (h) RPN localization loss with modified anchor boxes, (i) total loss with default scales and aspect ratio, (j) total loss with modified anchor boxes.



**FIGURE 18.** t-SNE plots of the proposed and existing methods. (a) final proposed RFCN model, (b) optimized model after the application of several weights optimization techniques and analyzing position-sensitive score maps, (c) RFCN model with default settings (after the application of translational augmentation techniques), (d) Faster RCNN ResNet-101 model with default settings (after the application of translational augmentation techniques).

confused with apple European canker, apple healthy leaves, pear fire blight, pear healthy leaves, and pear scab. Likewise, the Faster RCNN ResNet-101 model (Fig. 18 (d)) provided a degraded clustering performance as compared to the final RFCN model. The distinct features of apple black spot, apple European canker, apple mosaic virus, grapevine black spot, pear fire blight, pear healthy leaves, and pear scab were not made a

proper cluster and confused with the features of other healthy/disease classes. This shows that the proposed modifications in the anchor boxes generate a significant difference in the feature extraction of the healthy and diseased plant classes.

- Moreover, the mAP is improved with a margin of 6.406% (Fig. 19). Also, a significant improvement in individual AP of classes such as apple black rot, apple

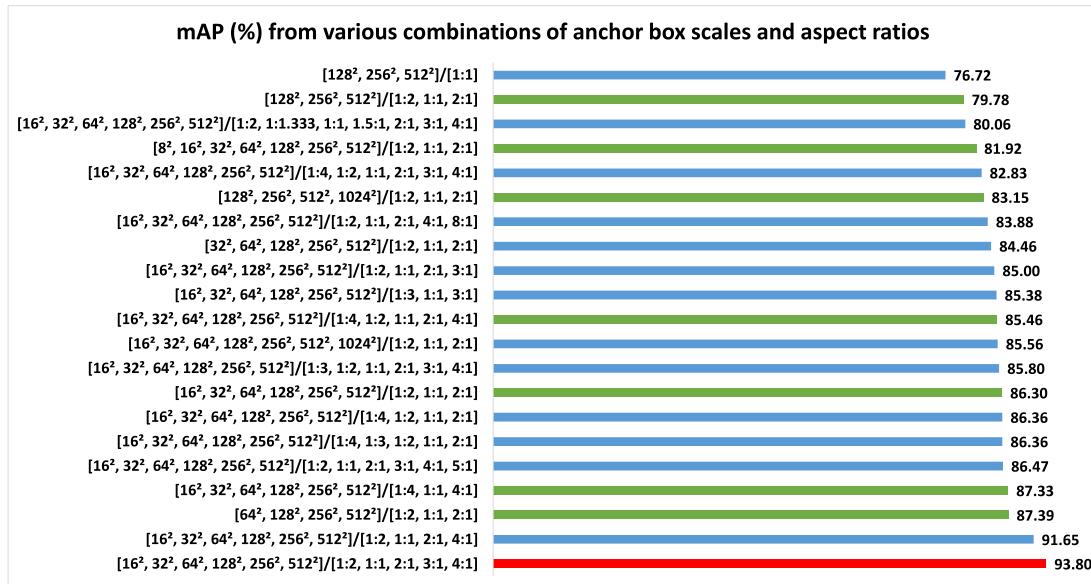


FIGURE 19. A summary of mAP with various specifications of anchor boxes.

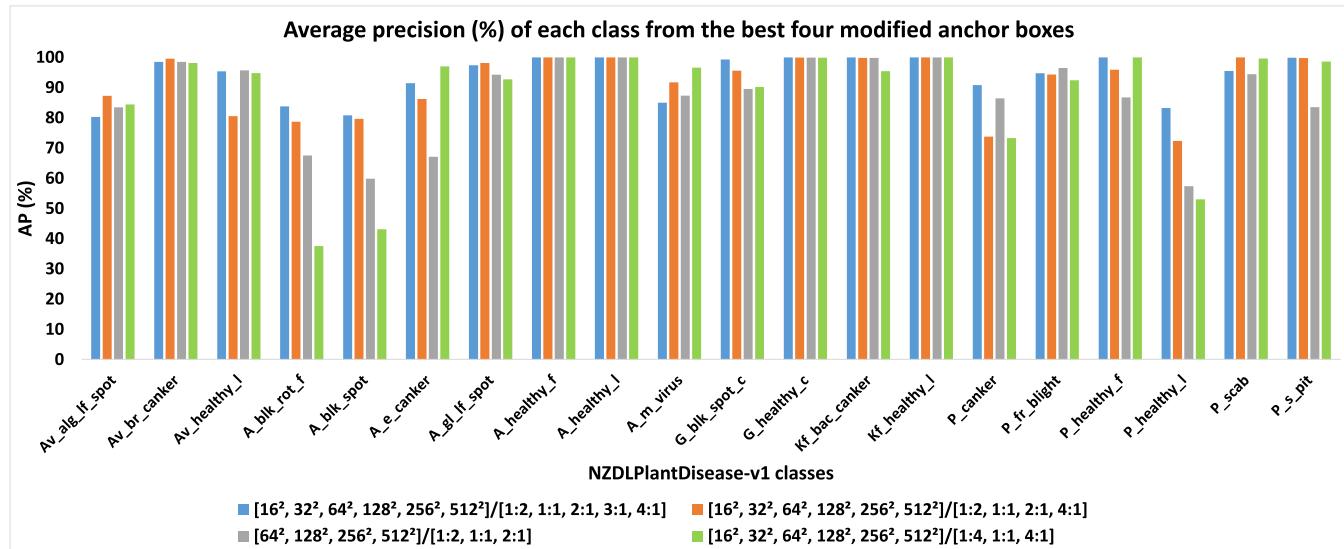


FIGURE 20. Average precision of each class from four prominent anchor box specifications.

black spot, apple European canker, pear healthy (leaves), and pear stony pit, as shown in Fig. 19.

- Other combinations of gradual addition of aspect ratios are examined, as shown in Fig. 19. One of the prominent combinations of 1:2, 1:1, 2:1, and 4:1 has attained a high mAP of 91.65
- In conclusion, the addition of various small scales like  $16 \times 16$ ,  $32 \times 32$ ,  $64 \times 64$ , and aspect ratios of 3:1 and 4:1 to the default anchor box have significantly improved the performance of the RFCN model with an mAP of 93.8. Furthermore, the AP of several classes is improved, and no class has achieved an AP of less than 80. Moreover, 12 classes have achieved

a high AP of  $> 95$ , as shown in Fig. 20. The class-wise performance of the four prominent combinations of enhanced anchor boxes is presented in Fig. 20. A few examples of the false-negative results by the default anchor boxes, solved by the enhanced anchor boxes are presented in Fig. 21. A pictorial representation of the proposed modification of the anchor boxes is presented in Fig. 22.

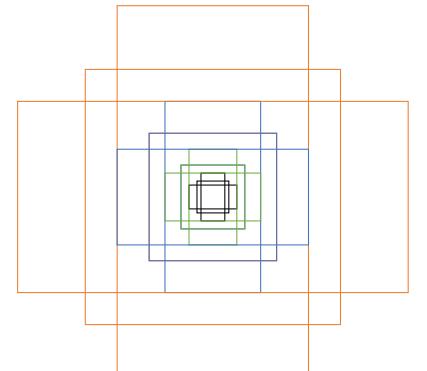
#### G. OVERALL REMARKS ON THE PREVIOUS STEPS

A summary of the results presented from Section III-A to Section III-F is provided as under:

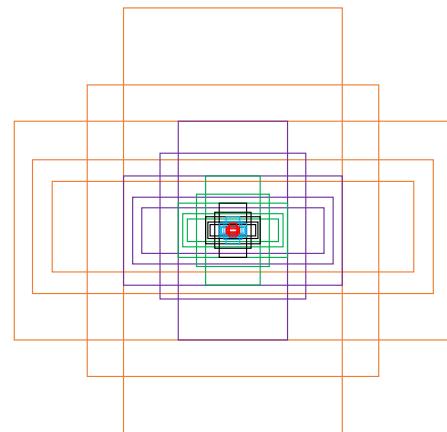


**FIGURE 21.** False-negative by default anchor box and true positive by final enhanced anchor box of classes including apple black rot, apple black spot, apple European canker, and pear healthy.

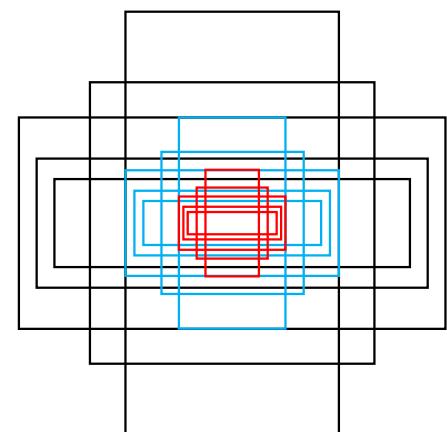
- The proposed deep learning (DL)-based method gradually enhanced the accuracy of plant disease detection from the step of comparison between deep learning architectures (Section III-A) to the enhancement of anchor boxes (Section III-F). The average precision of each class along with the mean average precision of the deep learning models are evaluated for each stage of the research. Each step has significance in terms of better training and testing results. The main reason for getting better results was that each step performed an in-depth analysis and identified a strong motivation for the subsequent steps to further improve mAP.
- For example, a comprehensive analysis of several DL models was performed to select the best-suited model. This selection was done and validated by using augmentation techniques. But the mAP obtained by all augmentation methods was significantly reduced. To cope with this problem, a category-wise comparison of the augmentation technique was performed, which gave us the best-suited technique for the selected application.
- Similarly, after the application of various techniques such as image resizers, interpolators, weight initializers, batch normalization, and deep learning optimizers, pear scab achieved unsatisfactory results. To attain a high AP of pear scab, the major novelty of the original RFCN model was analyzed, and the enhancement of the anchor boxes was attempted. In this way, the strong analyses of each step gave us the solid grounds for applying the following/next steps. A summary of the all steps including the best-selected method/model along with mAP is presented in Table 7.
- From Table 7, it can be concluded that all succeeding steps achieved a higher mAP, compared to its previous step. Furthermore, the most effective step in terms of improvement in the mAP was found to be the enhancement of the anchor boxes with an improvement of 6.406 compared to its earlier step.
- The effectiveness of the proposed approach has also been presented by the confusion matrix. For example, classes such as the apple black spot and pear



(a)



(b)



(c)

**FIGURE 22.** Anchor boxes for the RFCN model (a) present default anchor scales and aspect ratio, (b) present modified anchor boxes, (c) presents a zoomed version of 16 × 16 (red-colored boxes), 32 × 32 (blue colored boxes), and 64 × 64 (black colored boxes) scales sizes in the proposed version.

stony pit were confused with the healthy apple (leaves) and the fire blight, respectively, during the initial step of the methodology (Fig. 23 (a)). This can be verified by their detection results, already presented in

**TABLE 7.** Main findings of each step of the proposed methodology.

Steps	Best-obtained method/model	Mean average precision (%)
Step-a: Comparison between dl architectures	RFCN model	74.47
Step-b: Effects of data augmentation techniques	Original and translational changes in the images (OT)	76.76
Step-c: Effects of image resizers and interpolators	Fixed-shape resizer with bicubic interpolator	80.59
Step-d: Effects of weight initializers, batch normalization, and optimizers	Random normal (RN), batch normalization (BN), SGD with momentum (SGDm)	RN: 81.506, BN and SGDm: 85.94,
Step-e: Performance enhancement of pear scab	Position-sensitive score maps: 3x3, zoomed pear scab images	87.394
Step-f: Enhancement of anchor boxes	Scales: [16 <sup>2</sup> , 32 <sup>2</sup> , 64 <sup>2</sup> , 128 <sup>2</sup> , 256 <sup>2</sup> , 512 <sup>2</sup> ] and aspect ratios: 1:2, 1:1, 2:1, 3:1, 4:1	93.80

Class	Predicted																			
	A_blk_rot_f	A_blk_spot	A_e_canker	A_gf_if_spot	A_healthy_f	A_healthy_I	A_m_virus	Av_alg_if_spot	Av_br_canker	Av_healthy_I	G_blk_spot_c	G_healthy_c	Kf_bac_canker	Kf_healthy_I	P_canker	P_fr_blight	P_healthy_f	P_healthy_I	P_scab	P_s_pit
A_blk_rot_f	8	0	0	0	2	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0
A_blk_spot	0	13	0	5	0	7	2	0	0	1	0	0	0	0	0	0	2	2	0	0
A_e_canker	0	2	15	0	0	0	0	0	3	0	0	0	0	0	4	1	0	0	0	0
A_gf_if_spot	0	5	0	46	1	0	0	2	0	10	0	0	0	0	0	0	0	1	0	0
A_healthy_f	0	0	0	0	20	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
A_healthy_I	0	0	0	0	0	16	0	0	0	1	0	0	0	0	0	0	0	1	0	0
A_m_virus	0	10	1	4	0	5	34	1	0	0	0	0	0	0	0	0	0	0	0	0
Av_alg_if_spot	0	0	0	0	0	0	0	151	12	0	0	0	0	0	0	0	0	0	0	0
Av_br_canker	0	0	0	1	0	0	0	0	29	6	0	0	0	0	0	0	0	0	3	0
Av_healthy_I	0	0	0	0	0	1	0	0	0	17	0	0	0	0	0	0	0	0	0	0
G_blk_spot_c	0	0	0	0	0	0	0	0	0	0	12	1	0	0	0	0	0	0	0	0
G_healthy_c	0	0	0	0	0	0	0	0	0	0	1	15	0	0	0	1	0	0	0	0
Kf_bac_canker	0	1	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0
Kf_healthy_I	0	0	0	0	0	0	0	0	0	0	0	27	0	0	0	0	0	0	0	0
P_canker	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	32	0	0	0	0
P_fr_blight	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0
P_healthy_f	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	0
P_healthy_I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	52	0	0	0
P_scab	0	6	0	0	0	0	0	0	1	2	0	0	0	0	0	1	0	3	18	0
P_s_pit	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	5	0	0	7

(a)

Class	Predicted																			
	A_blk_rot_f	A_blk_spot	A_e_canker	A_gf_if_spot	A_healthy_f	A_healthy_I	A_m_virus	Av_alg_if_spot	Av_br_canker	Av_healthy_I	G_blk_spot_c	G_healthy_c	Kf_bac_canker	Kf_healthy_I	P_canker	P_fr_blight	P_healthy_f	P_healthy_I	P_scab	P_s_pit
A_blk_rot_f	47	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A_blk_spot	0	106	0	8	0	10	10	0	0	5	0	0	0	9	0	0	1	0	0	0
A_e_canker	0	0	93	5	1	1	0	0	2	0	0	0	0	1	0	0	0	0	0	0
A_gf_if_spot	0	4	0	483	0	10	8	2	16	0	0	0	1	0	0	0	0	0	0	0
A_healthy_f	0	0	0	0	135	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
A_healthy_I	0	0	0	0	0	178	0	0	0	0	0	0	0	1	0	2	0	0	0	0
A_m_virus	0	6	0	32	0	0	0	0	0	3	0	0	0	12	0	0	0	0	0	0
Av_alg_if_spot	0	0	0	0	0	0	0	319	43	9	0	0	0	0	0	0	0	0	0	0
Av_br_canker	0	7	0	2	0	0	0	0	395	8	0	0	0	0	0	0	0	0	1	0
Av_healthy_I	0	0	0	0	0	10	0	0	0	145	0	0	0	0	0	0	1	1	0	0
G_blk_spot_c	0	0	0	0	0	0	0	0	0	0	86	11	0	0	0	0	0	0	0	0
G_healthy_c	0	0	0	0	0	0	0	0	0	0	130	0	0	0	0	0	0	0	0	0
Kf_bac_canker	0	0	0	0	0	0	0	0	0	0	0	0	0	109	0	0	0	0	0	0
Kf_healthy_I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	213	0	0	0	0
P_canker	0	0	13	0	0	0	0	0	0	0	0	0	0	53	0	0	0	0	0	0
P_fr_blight	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	114	0	0	0	0
P_healthy_f	0	0	0	0	0	4	0	0	0	24	0	0	0	0	0	0	148	1	0	0
P_healthy_I	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	213	0	0
P_scab	0	4	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	3	95	0
P_s_pit	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	161

(b)

**FIGURE 23.** Confusion matrix for RFCN model (a) from the first step: training by original images with default settings, (b) from the last step: after the enhanced anchor boxes.

Figs. 9 (d) and (f). Consequently, a low recall of these classes was attained at 40.62 and 36.84, respectively. However, none of the classes suffered from a high number of wrong/missed classifications after the enhanced anchor boxes (Fig. 23 (b)), which led to significantly high mAP.

#### H. VALIDATION OF THE FINAL RESULTS

This study has also validated the results and the claims described in this article, in two ways. The first technique adopted is the stratified five-fold cross-validation, through

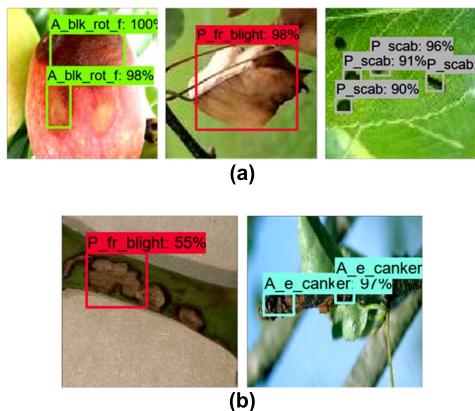
which the dataset images of each class are folded five times so that the testing images in each fold are different from one-fold to another. This technique has been applied because of the class imbalance problem in the generated dataset and it avoids biased distribution in the dataset for each fold. The mAP obtained through all folds varied from 0.65% to 1.13% in the optimized configuration of the RFCN model, as compared to the final mAP. The first fold is considered as a default in which the modified/optimized RFCN model obtained 93.80%, while fold2, fold3, fold4, and fold5 attained mAP of 93.15%, 94.93%, 94.62%, and 93.09%, respectively.

To further evaluate the final mAP by the proposed method, the variance was calculated by the formula (31) and evaluated as 0.70157.

$$\text{Variance} = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1} \quad (31)$$

where  $n$  is the number of folds,  $x_i$  is the fold number (fold1, fold2, etc.), and  $\bar{x}$  is the mean.

The next way to validate the claimed results is to test the optimized model on an external curated dataset. The original and translational/rotational change - OT augmentation method is also applied to that testing dataset. The mAP is 87.95%, which is only 5.85% lower than the final mAP obtained from the testing sub-dataset of the proposed dataset. Eleven classes, including avocado algal leaf spots, avocado branch cankers, avocado healthy, apple black rot, apple glomerella leaf spot, apple healthy (fruit), grapevine healthy, kiwifruit healthy, pear healthy (fruit), pear fire blight, and pear scab, are detected with a high AP of more than 90%. However, a few classes, such as pear canker, grapevine black spot, and pear stony pit, attained an AP of less than 80%. Therefore, the difference in mAP is obtained from that obtained by the testing images of the NZDLPPlantDisease-v1 dataset. Examples of a few classes that achieved high and low AP from the external dataset are presented in Fig. 24.



**FIGURE 24.** Results from an external dataset (a) True positive results of apple black rot, pear fire blight, and pear scab; (b) false positives of grapevine black spot and pear canker.

## I. LIMITATIONS OF THE STUDY

Although the presented methodology has successfully detected the plant disease using the proposed dataset. Still, there are a few limitations of this study that can be taken into account in future research. The presence of disease in multiple organs of apple and pear has been considered for this research. Whereas, for grapevine, avocado, and kiwifruit, disease in only one plant organ has been considered. Moreover, only one disease class is presented for both grapevine and kiwifruit. Therefore, the proposed dataset should be further extended to get more insight into deep learning-based plant disease detection. Moreover, the validation of the

modified/optimized model on an external generated dataset revealed that few of the classes did not achieve a high AP. One of the reasons could be the absence of diversity in the samples of those classes in the presented dataset. Furthermore, the dataset images from both sides of the plant organs should be considered. For example, the symptoms of the disease on the front and backside of the plant leaf could be included. This would generate more variety in the symptoms of plant disease. Also, all dataset images were collected from New Zealand horticultural fields. However, the dataset can be extended by capturing images of similar diseases in the same crops from horticultural fields in different countries. Moreover, the annotation was a bit tiring process due to the addition of the augmented images. Furthermore, as this article has addressed various practical problems, the most difficult among them was the detection of multiple diseases in a plant organ at a time. This task required even more time to correctly annotate the task. Therefore, it can be said that there is still a human intervention to use deep learning to perform complex task like plant disease detection.

## IV. CONCLUSION AND FUTURE DIRECTIONS

This study addresses various research gaps in the identification of plant diseases based on deep learning. In this regard, a new dataset called NZDLPPlantDisease-v1 is generated, and a DL-based approach is presented to detect and localize the disease in five of the most important New Zealand horticultural crops in terms of export value. After training and testing various DL architectures, the region-based fully convolutional network (RFCN) has achieved the highest mean average precision with and without the application of augmentation techniques. The proposed methodology consists of a comprehensive evaluation of various techniques impacted on the deep learning model that has not yet been explored for plant disease identification tasks. Furthermore, a modified/optimized version of the RFCN model is proposed by performing an in-depth analysis of position-sensitive score maps and anchor-box scales with aspect ratios. An improved mAP of 93.80% is achieved, which was 19.33% better than the default setting. The optimized RFCN includes training the model with a fixed-shape resizer with a bicubic interpolator, a random normal initializer, use of batch normalization, and SGD with a momentum optimizer. It is also observed that the translational/rotational augmentation method is the most suitable for obtaining satisfactory results. Furthermore, the addition of a  $16 \times 16$ ,  $32 \times 32$ ,  $64 \times 64$  scales with an aspect ratio of 3:1 and 4:1 significantly improved the performance of the RFCN. The optimized/modified RFCN model has successfully answered research questions, including the detection of diseases in several plant organs, the presence of multiple diseases in one organ at a time, and the identification of diseases in different crops using the same trained DL model. Finally, the statements and results are validated by two different methods: stratified five-fold cross-validation and testing on an external dataset. These validation approaches demonstrate the significance and novelty of this study.

Furthermore, one of the advantages of this study is that different crops (selected for this research) have certain variations in their environments/backgrounds. Therefore, high average precision in each class shows an extended potential of deep learning technology for the detection of plant diseases, considering various challenges of the horticultural environment.

The idea/methodology proposed in this study can be utilized in several ways in future studies. A deep learning-based method can be embedded in automated/robotic systems to apply disease control techniques. For example, a fungicide spray can be applied to the defective parts of plants using a robotic manipulator. Furthermore, the diseases are treated differently based on the pathogen affecting the plants. Therefore, the research question related to the detection of multiple classes of plant diseases (suffering from different diseases at a time) [50] could be useful for implementing a cost-effective protection system. For instance, black spots on apples are normally treated with a fungicide spray, whereas no such treatment is available for apple viruses [51]. Hence, this research will be helpful for growers to take appropriate treatment measures after detecting multiple plant diseases in an organ.

In addition, various tasks can be performed to further enhance research on DL-based plant diseases. For instance, advanced data augmentation techniques, including super-resolution convolutional neural networks (SRCNNs) and super-resolution generative adversarial networks (SRGANs), can be explored. Moreover, segmentation-based DL models can be leveraged and modified by using the generated dataset. Furthermore, the performance metrics presented in [52] can also be explored to perform a more in-depth analysis of the multi-label plant disease detection problem. Some other research ideas can be explored to further strengthen the research on DL-based solutions for agricultural problems. For example, the sensitivity analysis (like the one performed for a teleoperation system to examine the effects of important parameters on the system performance [53]) can be performed for the DL models to implement various agricultural operations. Moreover, the layer-wise output of the well-known DL models could be visualized to modify the hidden layer. The comparison of CNNs with CapsuleNet models can also be emphasized, as this class of machine learning is being explored for various object detection problems [54].

## AVAILABILITY OF DATA

The dataset generated and analyzed during the current study is available in the GitHub repository <https://github.com/kmarif/NZDLPlantDisease-v1>.

## ACKNOWLEDGMENT

The authors are thankful to Dr. Carl Mesarich (School of Agriculture and Environment, Massey University) for providing the common and scientific names of plant diseases. They are also grateful to Fakhia Hammad and Muhammad Taha for their help with the annotation of the dataset images.

## REFERENCES

- [1] A. G. Aitken and I. J. Warrington. (2020). *Fresh Facts*. Plant & Food Research. [Online]. Available: <https://www.freshfacts.co.nz/files/freshfacts-2020.pdf>
- [2] J. Liu and X. Wang, "Plant diseases and pests detection based on deep learning: A review," *Plant Methods*, vol. 17, no. 1, pp. 1–18, Dec. 2021.
- [3] M. H. Saleem, J. Potgieter, and K. M. Arif, "Automation in agriculture by machine and deep learning techniques: A review of recent developments," *Precis. Agricult.*, vol. 22, no. 6, pp. 2053–2091, 2021.
- [4] V. S. Dhaka, S. V. Meena, G. Rani, D. Sinwar, K. Kavita, M. F. Ijaz, and M. Woźniak, "A survey of deep convolutional neural networks applied for prediction of plant leaf diseases," *Sensors*, vol. 21, no. 14, p. 4749, Jul. 2021.
- [5] M. Alessandrini, R. C. F. Rivera, L. Falaschetti, D. Pau, V. Tomaselli, and C. Turchetti, "A grapevine leaves dataset for early detection and classification of esca disease in vineyards through machine learning," *Data Brief*, vol. 35, Apr. 2021, Art. no. 106809.
- [6] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers Plant Sci.*, vol. 7, p. 1419, Sep. 2016.
- [7] M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep learning for tomato diseases: Classification and symptoms visualization," *Appl. Artif. Intell.*, vol. 31, no. 4, pp. 299–315, 2017.
- [8] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Comput. Electron. Agric.*, vol. 161, pp. 272–279, Jun. 2019.
- [9] O. O. Abayomi-Alli, R. Damaševičius, S. Misra, and R. Maskeliūnas, "Cassava disease recognition from low-quality images using enhanced data augmentation model and deep learning," *Exp. Syst.*, vol. 38, no. 7, Nov. 2021, Art. no. e12746.
- [10] L. Bi and G. Hu, "Improving image-based plant disease classification with generative adversarial network under limited training set," *Frontiers Plant Sci.*, vol. 11, p. 1945, Dec. 2020.
- [11] D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," 2015, *arXiv:1511.08060*.
- [12] A. Abbas, S. Jain, M. Gour, and S. Vankudothu, "Tomato plant disease detection using transfer learning with C-GAN synthetic images," *Comput. Electron. Agricult.*, vol. 187, Aug. 2021, Art. no. 106279.
- [13] W. Albattah, M. Nawaz, A. Javed, M. Masood, and S. Albahli, "A novel deep learning method for detection and classification of plant diseases," *Complex Intell. Syst.*, vol. 8, no. 1, pp. 507–524, Feb. 2022.
- [14] K. Kc, Z. Yin, M. Wu, and Z. Wu, "Depthwise separable convolution architectures for plant disease classification," *Comput. Electron. Agricult.*, vol. 165, Oct. 2019, Art. no. 104948.
- [15] M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease classification: A comparative evaluation of convolutional neural networks and deep learning optimizers," *Plants*, vol. 9, no. 10, p. 1319, Oct. 2020.
- [16] M. H. Saleem, S. Khanchi, J. Potgieter, and K. M. Arif, "Image-based plant disease identification by deep learning meta-architectures," *Plants*, vol. 9, no. 11, p. 1451, Oct. 2020.
- [17] F. Alvaro, Y. Sook, K. Sang, and P. Dong, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors*, vol. 17, no. 9, p. 2022, 2017.
- [18] A. Karlekar and A. Seal, "SoyNet: Soybean leaf diseases classification," *Comput. Electron. Agricult.*, vol. 172, May 2020, Art. no. 105342.
- [19] C. K. Sunil, C. D. Jaidhat, and N. Patil, "Cardamom plant disease detection approach using EfficientNetV2," *IEEE Access*, vol. 10, pp. 789–804, 2022.
- [20] U. P. Singh, S. S. Chouhan, S. Jain, and S. Jain, "Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease," *IEEE Access*, vol. 7, pp. 43721–43729, 2019.
- [21] N. Yao, F. Ni, Z. Wang, J. Luo, W.-K. Sung, C. Luo, and G. Li, "L2MXception: An improved xception network for classification of peach diseases," *Plant Methods*, vol. 17, no. 1, pp. 1–13, Dec. 2021.
- [22] J. Liu and X. Wang, "Tomato diseases and pests detection based on improved Yolo V3 convolutional neural network," *Frontiers Plant Sci.*, vol. 11, p. 898, Jun. 2020.
- [23] A. Fuentes, S. Yoon, M. H. Lee, and D. S. Park, "Improving accuracy of tomato plant disease diagnosis based on deep learning with explicit control of hidden classes," *Frontiers Plant Sci.*, vol. 12, p. 2938, Dec. 2021.
- [24] J. Sun, Y. Yang, X. He, and X. Wu, "Northern maize leaf blight detection under complex field environment based on deep learning," *IEEE Access*, vol. 8, pp. 33679–33688, 2020.

- [25] P. Wang, T. Niu, Y. Mao, Z. Zhang, B. Liu, and D. He, "Identification of apple leaf diseases by improved deep convolutional neural networks with an attention mechanism," *Frontiers Plant Sci.*, vol. 12, p. 1997, Sep. 2021.
- [26] X. Xie, Y. Ma, B. Liu, J. He, S. Li, and H. Wang, "A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks," *Frontiers Plant Sci.*, vol. 11, p. 751, Jun. 2020.
- [27] J. G. Barbedo, "Deep learning applied to plant pathology: The problem of data representativeness," *Tropical Plant Pathol.*, vol. 47, pp. 85–94, Aug. 2021.
- [28] E. Mwebaze, T. Gebru, A. Frome, S. Nsumba, and J. Tusubira, "ICassava 2019 fine-grained visual categorization challenge," 2019, *arXiv:1908.02900*.
- [29] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera, "Deep feature based Rice leaf disease identification using support vector machine," *Comput. Electron. Agricult.*, vol. 175, Aug. 2020, Art. no. 105527.
- [30] H. T. Rauf, B. A. Saleem, M. I. U. Lali, M. A. Khan, M. Sharif, and S. A. C. Bukhari, "A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning," *Data Brief*, vol. 26, Oct. 2019, Art. no. 104340.
- [31] T. Wiesner-Hanks, E. L. Stewart, N. Kaczmar, C. DeChant, H. Wu, R. J. Nelson, H. Lipson, and M. A. Gore, "Image set for deep learning: Field images of maize annotated with disease symptoms," *BMC Res. Notes*, vol. 11, no. 1, pp. 1–3, Dec. 2018.
- [32] J. Jepkoech, D. M. Mugo, B. K. Kenduiywo, and E. C. Too, "Arabica coffee leaf images dataset for coffee leaf disease detection and classification," *Data Brief*, vol. 36, Jun. 2021, Art. no. 107142.
- [33] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. Cham*, Switzerland: Springer, 2016, pp. 21–37.
- [34] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 91–99.
- [35] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 379–387.
- [36] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
- [37] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10781–10790.
- [38] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
- [39] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
- [40] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.
- [41] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
- [42] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*.
- [43] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning," *Coursera, Video Lectures*, vol. 264, no. 1, pp. 2146–2153, 2012.
- [44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [45] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 2, pp. 1–25, 2012.
- [46] H. He and Y. Ma, *Imbalanced Learning: Foundations, Algorithms, and Applications*. Hoboken, NJ, USA: Wiley, 2013.
- [47] J. G. A. Barbedo, "Factors influencing the use of deep learning for plant disease recognition," *Biosyst. Eng.*, vol. 172, pp. 84–91, Aug. 2018.
- [48] O. Rukundo, "Effects of image size on deep learning," 2021, *arXiv:2101.11508*.
- [49] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.
- [50] M. B. Riley, M. R. Williamson, and O. Maloy, "Plant disease diagnosis," *Plant Health Instructor*, vol. 10, 2002, doi: [10.1094/PHI-1-2002-1021-01](https://doi.org/10.1094/PHI-1-2002-1021-01).
- [51] S. U. Nabi, G. Madhu, G. P. Rao, and V. K. Baranwal, "Development of multiplex RT-PCR assay for simultaneous detection of four viruses infecting apple (*Malus domestica*)," *Lett. Appl. Microbiol.*, vol. 74, no. 4, pp. 586–592, 2022.
- [52] X.-Z. Wu and Z.-H. Zhou, "A unified view of multi-label performance measures," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3780–3788.
- [53] R. Uddin, M. H. Saleem, and J. Ryu, "Parametric sensitivity analyses for perceived impedance in haptic teleoperation," *Int. J. Control. Autom. Syst.*, vol. 17, no. 8, pp. 2083–2096, Aug. 2019.
- [54] M. L. Mekhalfi, M. B. Bejiga, D. Soresina, F. Melgani, and B. Demir, "Capsule networks for object detection in UAV imagery," *Remote Sens.*, vol. 11, no. 14, p. 1694, Jul. 2019.



**MUHAMMAD HAMMAD SALEEM** received the B.E. and M.E. degrees in electrical engineering from the NED University of Engineering and Technology, Karachi, Pakistan, in 2016 and 2018, respectively. He is currently pursuing the Ph.D. degree with the School of Food and Advanced Technology (SF&AT), Massey University, Auckland, New Zealand. Before starting his Ph.D., he was a Lecturer with the Department of Electrical Engineering, NED University of Engineering and Technology. His research interests include artificial intelligence, deep learning, computer vision, machine learning, agricultural robotics, haptics, and teleoperation.



**JOHAN POTGIETER** is currently a Professor of robotics at the Massey AgriFood (MAF) Digital Laboratory. He has a well-established reputation for his work in educational robotics and has been inducted into the World Robotics Education and Competition (REC) Foundation Hall of Fame. His current research interests include additive manufacturing and advanced mechatronics/robotics with applications in medicine and automotive related areas.



**KHALID MAHMOOD ARIF** (Senior Member, IEEE) received the B.Sc. degree in mechanical engineering from the University of Engineering and Technology, Lahore, Pakistan, in 2000, the M.Eng. degree in engineering synthesis from The University of Tokyo, Japan, in 2004, and the Ph.D. degree in mechanical engineering from Purdue University, West Lafayette, IN, USA, in 2011. From 2007 to 2011, he was a Fulbright Scholar at Purdue University. He is currently a Senior Lecturer in mechatronics and robotics at Massey University, Auckland. Prior to joining Massey University, in 2012, he was an Assistant Professor with the Department of Mechatronics and Control Engineering, University of Engineering and Technology, Lahore. He is also the Director of the Sensors and Smart Systems Laboratory. His research interests include sensors, the IoT, robotics, and additive manufacturing.