


RESEARCH

Open Access



PFDI: a precise fruit disease identification model based on context data fusion with faster-CNN in edge computing environment

Poonam Dhiman¹, Poongodi Manoharan^{2*} , Umesh Kumar Lilhore³, Roobaea Alroobaea⁴, Amandeep Kaur⁵, Celestine Iwendi⁶, Majed Alsafyani⁷, Abdullah M. Baqasah⁸ and Kaamran Raahemifar^{9,10,11}

*Correspondence:
dr.m.poongodi@gmail.com

² Division of Information and Communication Technology, College of Science and Engineering, Hamad Bin Khalifa University, 602000 Doha, Qatar
Full list of author information is available at the end of the article

Abstract

Fruits significantly impact everyday living, i.e., Citrus fruits. Numerous fruits have a solid nutritious value and are packed with multivitamins and trace components. Citrus fruits are delicate and susceptible to many diseases and infections. Many researchers have suggested deep and machine learning-based fruit disease detection and classification models. This research presents a precise fruit disease identification model based on context data fusion with Faster-CNN in an edge computing environment. The goal is to develop an accurate, efficient, and trustable fruit disease detection model, a critical component of autonomous food production in a robotic edge platform. This research examines and explores four different diseases of Citrus fruits using CNN deep learning models to be adopted as edge computing solutions. Identification of citrus diseases such as cankers black spot, greening, scab, melanosis, and healthy citrus fruits are implemented using the proposed sequential model without pruning, with pruning having different sparsity levels followed by post quantization. Through the transfer learning method, this model is optimized for the assignment of fruit disease detection employing visuals from two patterns: Near-infrared (NIFR) and RGB. Early and late data fusion techniques for integrating multi-model (NIFR and RGB) facts are evaluated. The accuracy obtained from the proposed model for the canker disease is 97%, scab 95%, melanosis 99%, Greening 97%, Black spot 97% and healthy 97%. In this paper, the results of the proposed model are compared and evaluated with the sparsity levels of 50–80%, 60–90%, 70–90%, and 80–90% pruning and also obtained the results of post-quantization on each level. The results show that the model size with 60–90% pruning can be counteracted to the 47.64 of the baseline model without significant loss of accuracy. Moreover, post-quantization can reduce the 60–90% pruning from 28.16 to 8.72. In addition to enhanced precision, the above initiative is much faster to implement for new fruit diseases because it needs bounding box annotation instead of pixel-level annotation.

Keywords: Data fusion, Deep learning, Pruning, Disease, Sparsity, Quantization, Edge computing, Citrus fruit

1 Introduction

Fruit products ought to be the establishment of a sound eating regimen. Citrus fruits are the signature agricultural product; nearly everybody consumes them consistently. From last decade's research results demonstrated the criticality of fruit product quality and its impact on human well beings [1, 2].

It is possible to achieve economic progress in agriculture by utilizing modern innovative solutions. The use of pesticides by farmers to prevent and treat different diseases and improve crop productivity is widespread. Diseases in fruit crops are a significant cause of problems, including reduced productivity and financial hardship for farmers. Therefore, the fundamental necessity in the agricultural field is early illness identification. Disease diagnosis is crucial for the efficient farming framework. From past decay post-harvesting of Citrus, one of the critical challenges is the recognition of exterior defects [3].

Citrus species and cultivars are highly unpredictable in colour and texture, which causes difficulties in establishing an unsupervised system capable of identifying the apparent diseases. Secondly, edge computing adheres suitable with low computational efficiency and is well adapted to infield estimation that allows for real-time response and enables data or results transfer over narrowband networks. As all agricultural field instruments have minimal resources, using underlying DL models infield is a significant obstacle. Identifying diseases accurately is crucial for using the right treatments, so diminution or eradication of the infection in the plantation may occasionally need specialized expertise or prior knowledge from the farmer [4].

Additionally, some detection procedures necessitate collecting samples and sending them to a lab for evaluation, taking much more average time, making the farming vulnerable to illness, and opening the door for disease to spread to other areas. Alternatively, researchers have started looking into novel techniques based on artificial intelligence and mobile devices to provide precise and dependable disease diagnosis methods. Furthermore, strategies utilizing machine learning and artificial intelligence have begun to be deployed because most illnesses significantly alter the visual appearance of leaves, fruits, and plant stems [5].

The paper contributed the efficiency to fruit disease detection, i.e., Citrus fruit, by using the Faster-CNN model and data fusion technique for the edge computing environment. To execute the network infrastructure, diverse edge node specifications are employed [6]. Our paper's novelty involves optimizing network size using magnitude-based pruning and post-quantization process. In this research, Citrus fruit data with different diseases collected from various public sources are sent to the edge network for classification purposes.

Complex deep-learning networks cannot do complex calculations for disease identification on low-resource devices. Intelligent devices of the edge network contribute to obtaining disease detection and classification and also edge clusters as complex computations by offloading deep learning contexts to the edge network. Also, the network is pruned and quantized to optimize network size so farmers can further use low-resource devices to detect the different classes of Citrus fruit disease during post-harvesting.

Below is a summary of our work's essential contribution:

1. A Faster-CNN model is designed already pre-trained on a sizable image database, like ImageNet, to develop a powerful fruit disease detection model that can be quickly learned with a limited number of images.
2. Multi-model is presented data fusion perspectives that merge data from NIFR and RGB images, resulting in state-of-the-art recognition accuracy. The proposed model is trained for 2 h over real-time K-60 GPU.
3. The proposed model is the first to use data fusion using faster CNN and combining the magnitude-based pruning techniques with different sparsity levels to reduce the huge complex networks.
4. Edge computing devices with constrained capacity cannot run multifaceted DNN-based applications. Pruning with quantization for the compression of faster-CNN models can be incorporated with edge computing to resolve this issue for successful implementation in limited resource scenarios.
5. In-depth performance reviews of citric fruit data from an online UCI dataset utilizing performance measurement metrics, such as accuracy curves, precision, loss and the F1 score, are conducted.

The complete article is organized as follows: Sect. 2 covers the related work, Sect. 3 covers the materials and methods, Sect. 4 covers the experimental analysis and result discussion, and Sect. 5 covers the conclusion and future scope.

2 Related work

The scientific community pays close attention to Multi-Access Edge Computing because of its scientific, technical, and business ramifications. The ETSI guideline integration mainly consolidates the arguments around edge computing. Nevertheless, most current MEC research applications are insufficient, hindering or invalidating their adoption. Understanding a variety of experimental prototypes, implementations, and deployments is crucial to closing this gap. The Early deployments can demonstrate the capabilities, constraints, linked technologies, and development tools for multi-edge computing adoption [7].

IoT data is transmitted to the cloud under the cloud-based infrastructure for decision-making, investigation, and data handling. It might not only result in high demand on the cloud but also in excessive network latency, which would potentially negate the advantages of cloud computing. Researchers are searching for new decentralized computing models for the IoT to address these difficulties. Edge computing is one such model that is gaining popularity among academic and industrial researchers. The primary principle of edge computing is to move data processing from distant cloud servers to near-edge devices. Developing more scalable, low-latency IoT systems is promising [8]. IoT has many applications in healthcare industries also. Utilizing machine learning methods, the Internet of things can focus on providing analysis between real-time data and historical data. The automatically generated IoT-based health—care device can be used to create a handheld, relatively low-cost, digital, and anonymous device that can locate near the area of healthcare centres and will be easily accessible [9].

Images can be incarcerated using various sensors like non-destructive, non-invasive sensors for evaluating agricultural products. Multispectral acquisition systems usually receive 2-D data at a time and scan across the 3D. Multispectral scanners and imagers are categorized based on the filtering technique, like optical and electronically tunable filters [10].

The paper proposed a machine vision system that detects deformities in the orange fruits and grades the defects that occurred on the orange fruit surface. The signs of the fault mark reflect the nature of the disease and propose the best approach to coping with the illness [11]. Commercial value decreases in lemon and orange fruit infected by physiological decay. The researcher presents the image processing technique to identify the physiological disorder on X-ray Images of orange and lemon fruit. kNN techniques were used to classify the image set [12] automatically. Hybridized methods using AlexNet and random forest techniques were implemented to classify different diseases present in Citrus fruits [13]. A modern AlexNet architecture premised on deep learning was used for successful disease detection. This approach consisted of four major processes: pre-processing, segmentation, feature extraction, and classification. Firstly, pre-processing was applied to boost the input object's attributes. The images were then segmented by using the Otsu technique. The Alex-Net model was first introduced as an extractor function. For the classification of citrus diseases, a random forest (RF) classifier was finally used [14].

In comparison, the efficiency of ML, i.e. SVM, RF, SGD and DL, is evaluated and compared. In the case of disease detection, the precision of the disease classification showed that DL methods perform much better than ML methods. The inhibitor of 1-methylcyclopropene (1-MCP) on Citrus fruit post-harvest green mould was examined using vivo analyses that revealed that 1-MCP substantially reduced the occurrence of green decay and degradation in Citrus fruit post-harvest and hindered mycelia growth and *P. digitatum* fungal growth [15]. Table 1 compares various existing models for fruit disease detection and classification methods.

Table 1 Comparison of various existing models for fruit disease detection

References	Method	Plant type	Edge computing	Bounding box prediction	Transfer learning	Accuracy %
[1]	Machine learning with ANN	Plant village	No	No	No	94.25
[2]	PCA	Apple leaves	Yes	No	No	95.24
[3]	Random forest	Cassava leaves	No	No	No	93.56
[4]	CNN model	Fruit classification	No	Yes	Yes	91.24
[5]	Deep learning	Apple leave	Yes	No	Yes	90.75
[6]	SVM with ANN	Cassava leaves	Yes	Yes	No	94.56
[7]	Hybrid random forest	Cassava leaves	No	No	No	95.33
Proposed model	Context data fusion with faster CNN	Citrus fruit disease	Yes	Yes	Yes	98.96

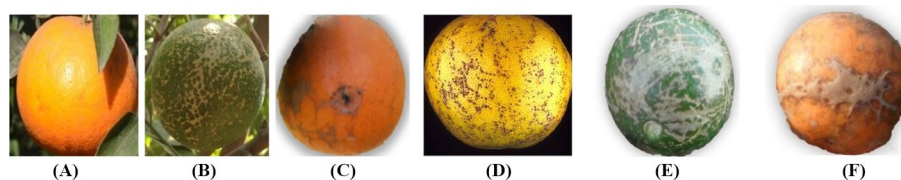


Fig. 1 Healthy and infected Fruit images collected from plant village, www and image gallery

Table 2 Count and proportion of citrus samples

Disease type	Number	Proportion
Canker	370	14.4
Scab	409	16
Melanose	280	10.9
Greening	440	17.2
Black spot	390	15.2
Healthy	667	26

3 Materials and methods

This section covers the working of the proposed model, algorithm, working, dataset and comparison parameters.

3.1 Citrus fruit disease dataset

Figure 1 shows the samples of images of citrus fruit. As target objects, healthy fruit and fruit exhibiting five kinds of typical blemishes like HLB, black spot, melanosis, canker, and scab were extracted.

Samples of each peel state are represented by Fig. 1. Figure 1A represents the healthy peel condition of the citrus sample. (Fig. 1B) HLB (*Candidatus Liberibacter asiaticus*) decreases the size of Citrus fruits and does not colour the fruit properly, i.e. it remains green, and with a bent central core, the fruit is skewed, inducing fruit deformity and cracking (Fig. 1B) [16]. Botched seeds in the affected fruit may taste salty and bitter. As several fruits fall prematurely from infected trees, HLB decreases citrus yields. Black spots (Fig. 1C) in fruits with a diameter of 0.12–0.4 are small, circular, and dangerous [17]. The presence of Citrus Black Spot symptoms can differ and cause aesthetic abnormalities on the fruit crust. The characteristic lesion, identified as a tricky spot, begins with tiny red brick spots with black borders that grow in size and produce tissue necrosis at the center of the lesion [18].

Melanose (Fig. 1D) is precipitated by *Diaporthe citri* and is distinguished by dispersed raised blotches of brown to black colour. The disease produces a gradual scar in the fruit which is barely capable of affecting the total yield of fruit production but causes noticeable blemishes, which reduce the viability of the fruit destined for the producing market [19]. The crop fungal infections commonly present in different regions producing citrus cultivars are Citrus scabs (Fig. 1E). Generally, the scab incidence seems to be more severe in-plane regions with frequent wetting compared to tropical areas. Scab damages can be identified less than a week after getting an infection of the fruit [20]. The

pathogens often occur with a gritty and irregular appearance as tiny dots. The fruit spot diameter of the Canker (Fig. 1F) is around 1–10 mm and is covered by water-dipped and yellow curve-like blemishes [21]. Table 2 represents the number of image samples for each type of disease and its proportion available in the dataset.

3.2 Proposed PFDI model

The proposed precise fruit disease identification (PFDI) model is based on Context data fusion with Faster-CNN in an edge computing environment.

3.2.1 Edge computing platform

Encouraging the remarkable resource consumption and delay in the processing time of the complex deep learning mechanism, there is a much need for an optimized tool for many applications. To detect or classify diseases present in Citrus fruits in a multi-classification manner, it’s essential to advance the existing deep learning framework employed in the automatic disease detection system. All image samples are collected from the various online platforms, while other services, such as pre-processing and computations in the edge nodes, use different clusters [22].

Figure 2 shows the working of the Edge computing architecture of the proposed model. The framework consists of four major modules, each corresponding to specific tasks. The first module consists of collector nodes. Web services or edge computing are deployed in the second module, the third module consists of the prediction process and model pruning with quantization, and user applications are deployed in module 4. The third module serves as a bridge between platform-based local and remote functions. The data is collected locally from the public domain; however, edge services are incorporated for processing and computational tasks after it has been set off [23].

Different clusters were created analogously to provide broader coverage of potential application scenarios. The platform’s applications and services are accessible to users through module four, the last module. The layer is, therefore, in charge of giving the user full access to all services offered by the edge computing platform. This module provides access to the computer vision application via virtual network computing, evaluating and

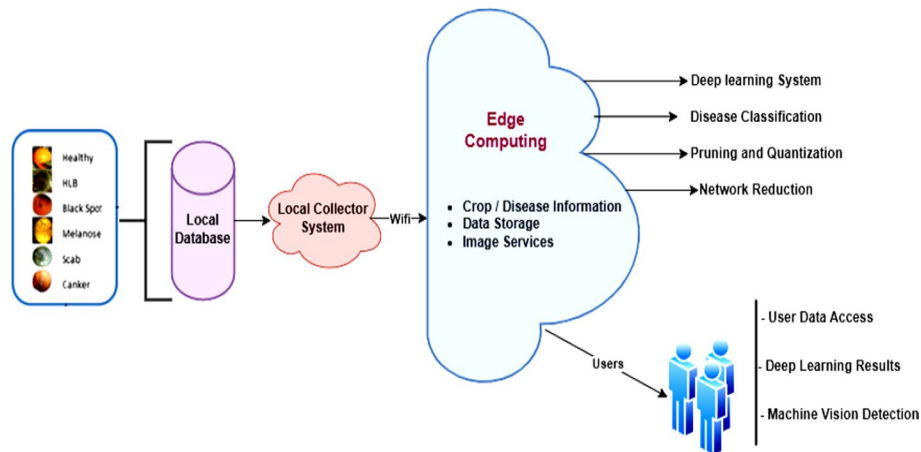


Fig. 2 Overall edge computing architecture

analyzing the outcome. The public dataset of Citrus fruit diseases of different types is sent to the local collector node. The detection layer contains a deep CNN model with magnitude pruning and quantization, and an internet connection is established. The visualization layer provides remote access to deep model output [24]. Comparing the edge learning servers to the most advanced cloud computing architecture, the edge learning servers reduce the workload of the network infrastructure. The image samples can be pre-processed to eliminate some challenges due to low contrast, such as light effects, flickering, etc. The pre-processing phase is critical as minimal disparity images decrease the lesion segmentation's accuracy in image processing [25].

We need to normalize to the equivalent image dimension to standardize the training data because all the images in the training data are sparse and have different dimension sizes. Hence, the size of 256×256 has been used for all images in the training set. The edge learning servers can also employ the deep learning-based technique for data augmentation to obtain new data without labelling costs. The data augmentation would also enhance the capabilities of the designed Faster-CNN model [26, 27]. Therefore, eight standards were included to supplement our training set and test set, i.e. horizontal flip, vertical flip, brightness, rescaling, shear, zca-whitening, rotation, height, and width shift. Initially, fruit samples collected were 2556, including six classes which reached 20,448 after applying the eight functions of data augmentation. Table 3 shows the instances of Citrus fruits obtained after performing different data augmentation operations on the dataset.

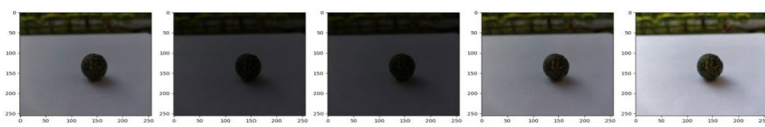
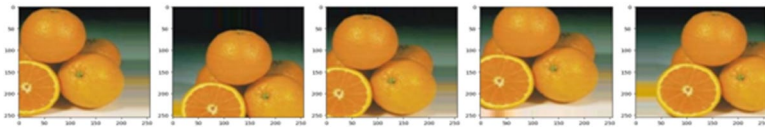
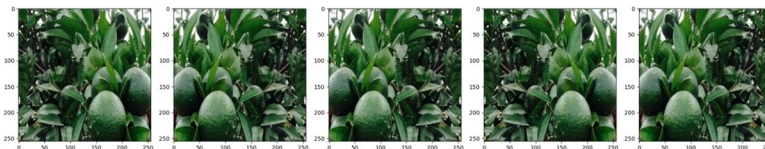
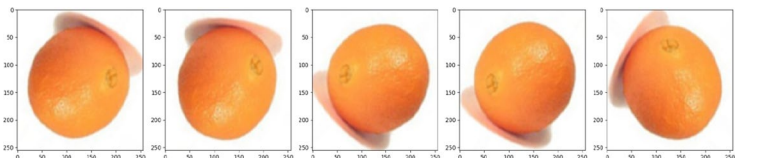
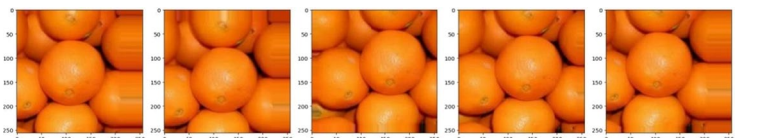
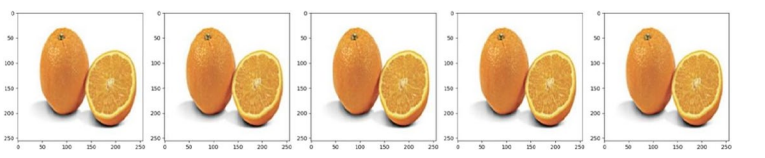
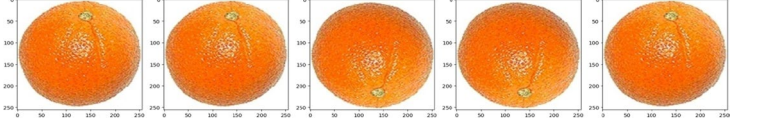
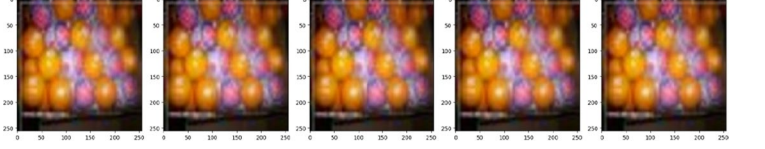
As the edge servers are near the client devices, the communication latency between the client devices and the edge servers is substantially lower than that of the cloud server. The model used for training. To offer end users consistent service, the cloud could be installed on edge servers, and image data can be continually uploaded to the cloud to modify the system. The classification process of Citrus fruit diseases is briefly explained in Fig. 3. The significant steps are (a) image pre-processing, (b) data augmentation, (c) Model implementation and training, (d) Pruning, (e) Quantization (d) Performance evaluation. As seen in Fig. 3, each phase comprises a series of steps. The detail of each process is given below.

3.2.2 Magnitude-based pruning with polynomial decay-based sparsity on faster-CNN model

By using deep learning, models have many trainable weights, where learning can be obtained. The contribution of all the weights and feature vectors to model performance is not equivalent. This paper's proposed model uses magnitude-based pruning, which removes insignificant weights after each epoch. This approach compares the exact size of the weight with a certain threshold value δ . Input vector x and weight vector w within the neuron is multiplied [28]. If the consequences in the vector are set to zero, the outcome will always be zero, as defined in Eq. 1. This, in effect, ensures that the neuron no longer contributes to model performance. Table 4 represents the proposed pruned sequential Convolutional neural network model configuration.

$$(w_t) = \{w_t : \text{ if } |w_t| > \delta 0 : \text{ if } |w_t| < \delta \quad (1)$$

Table 3 Samples of citrus fruits obtained after performing different data augmentation process

Data augmentation technique	Data augmentation sample
Brightness	
Height width shift	
Horizontal flip	
Rotation	
Shear	
Rescaling	
Vertical flip	
ZCA-whitening	

At every step, all the network connections are iteratively pruned by setting up the levels of step sparsity L using the polynomial decay function. Step sparsity L level is represented by the Eq. (2)

$$L = s_t + (s_i - s_t) \left(1 - \frac{j - i}{i - e} \right) \tag{2}$$

where the targeted sparsity level is represented by s_t ; initial sparsity level is s_i , initial iteration is i , e is the end iteration, and the current iteration is represented by j .

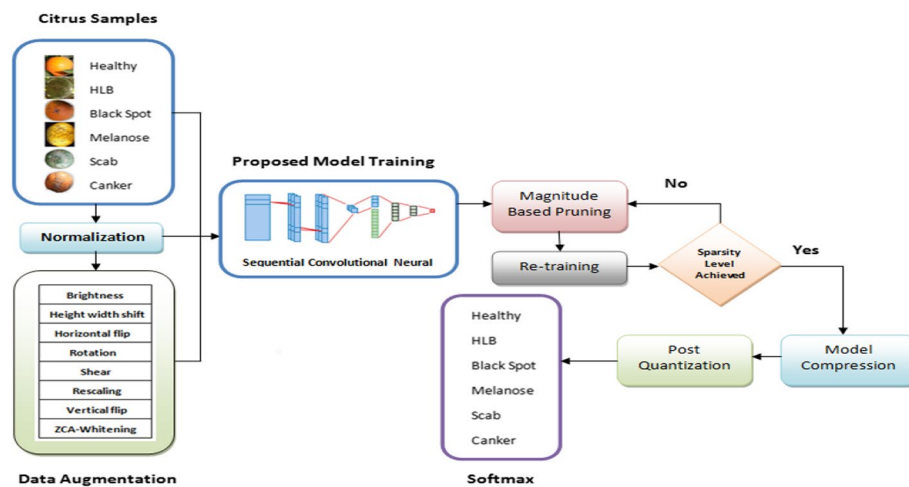


Fig. 3 Multi-classification and size reduction process

Table 4 Configuration of proposed pruned CNN model

Layer (type)	Output shape	Param #
Prune__low__magnitude__conv2d	(None, 256, 256, 32)	1762
Prune__low__magnitude__activation	(None, 256, 256, 32)	1
Prune__low__magnitude__conv2__1	(None, 256, 256, 32)	36,930
Prune__low__magnitude__activation	(None, 256, 256, 32)	1
Prune__low__magnitude__max__pool	(None, 256, 256, 32)	1
Prune__low__magnitude__dropout	(None, 256, 256, 32)	1
Prune__low__magnitude__conv2d__2	(None, 256, 256, 32)	73,794
Prune__low__magnitude__activation	(None, 256, 256, 32)	1
Prune__low__magnitude__con2d__3	(None, 256, 256, 32)	73,794
Prune__low__magnitude__activation	(None, 256, 256, 32)	12
Prune__low__magnitude__max__pool	(None, 256, 256, 32)	1
Prune__low__magnitude__dropout	(None, 256, 256, 32)	1
Prune__low__magnitude__flatten	(None,28,224)	1
Prune__low__magnitude__dense	(None,512)	28,901,890
Prune__low__magnitude__activation	(None,512)	1
Prune__low__magnitude__dropout	(None,512)	1
Prune__low__magnitude__dense__1	(None,5)	5127
Prune__low__magnitude__activation	(None,5)	1

The threshold of significant connections is calculated using the sparsity level step. The threshold value is the total weights multiplied by the position step sparsity level value. When the actual absolute value of every weight filter is below the threshold limit, all the mask values belonging to its weight are adjusted to 0, as shown in Eq. 2. Then the dot product sets the pruned weights as zero. More or less, polynomial decay-based sparsity can be used with increasing or decreasing speed as training progresses [29]. Initially, we set the model to be 50–80% sparse, increasingly getting sparser to eventually 80% and 90%. We begin at 0 and end at the end step. Finally, the prune_low_magnitude functionality, which generates the prunable model, will be executed from our initial baseline Faster-CNN model and the defined pruning_params. The requirements to digitize the

system, some numbers are allocated to various pieces of hardware. The conditions are the maximum number is offered with good resources.

3.2.3 Edge learning platform

For every six classes, including scab, HLB, melanosis, black spot, canker, and healthy Citrus fruits, the training, validation, and test sample comprise randomized input samples data with 8:2 ratios. For training, the model one set s used as training and validation another as the test set to evaluate the model training. Furthermore, an optimizer (SGD) controls the gradient steps for every dimension of the loss feature with 0.9 momentum magnitude. The model training is done on the image samples using stochastic gradient descent having 0.0001 as the learning rate, batch dimension is 64, and epoch range is 20. Furthermore, the output layer soft-max having categorical cross-entropy loss is used [30]. Figure 4 shows the edge learning framework used by the proposed technique.

3.2.4 Quantization process

Quantization lowers the model’s representation, enabling faster processing and smaller memory space. By quantizing all weights and activations, latency, processing, and power consumption can be boosted [31]. Quantization may be done during training, called pre-quantization, and post-quantization is done after training. In pre-quantization, the fixed point training approach is used to train the floating-point model directly. Post-quantization convert’s fixed-point representation from a pre-trained floating-point model and inferences can be taken from the fixed-point computation. In this paper, post-training quantization was adopted after pruning the proposed model. Fixed-point weights quantization has been applied that compressed the weights matrix from 32-bit to 8-bit floating-point values. Furthermore, Flat Buffer protocols are used as the basis for this

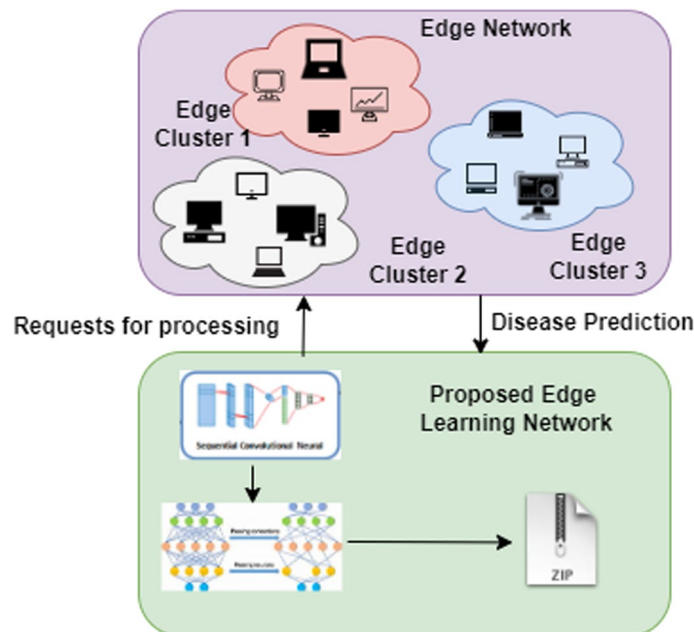


Fig. 4 Edge learning framework

transformation, which helps bypass many of the expensive traditional file parsing and un-parsing, leading to slower execution.

3.2.5 Context data fusion

Multiple data sources are combined through "data fusion," which results in statistical information that is more reliable, precise, and beneficial than the whole data supplied by any source alone. The proposed model applies a combination of early and late data fusion. The classification data from the two components colour and NIR visual imagery is combined in late fusion [31]. Each multi-model M generates initiatives for the (NM, R) terrain. Such region key points are coupled to create a particular set of (Ns * Pr = M * NM, Pr) region propositions, which combines the two modules. The prth proposed terrain of the Mth, model type is then set as a score (ScoreM, Pr). Equation 3 shows the score model formulation.

$$\text{ScoreM} = \sum_{M=1}^{NM} (\text{ScoreM}, \text{Pr}) \tag{3}$$

Like the early fusion data process, the input layer of faster-CNN has many channels 4 (one NIR and three RGB).

3.2.6 Transfer learning

It takes a lot of time and materials to train Faster-CNNs, particularly for networks with many layers. Two transfer-learning strategies, the "fixed feature extraction process" and "fine-tuning process", are employed to prevent retraining a whole network on Citrus fruit. The more straightforward option is a fixed feature extraction method that utilizes a channel that has received training on a previously untrained category or entity to identify or categorize it. To identify fruits in this instance, weights are utilized that were received training and managed to learn about Citrus fruits. The trained weights of the network that use the same architectural features are transferred directly to the main channel. We want to prepare with updated information through fine-tuning. The network's weights trained on fruit are utilized to prepare the network for detecting Citrus fruits with fewer training sets, including its base [32].

3.3 Performance measuring parameters

The proposed model PFDI and the existing model are compared based on performance measuring parameters precision (PC), Recall (RC), F1 score (FS) and accuracy (Acy).

$$\text{PC} = \left[\frac{\text{Tp}}{\text{Tp} + \text{FP}} \right] \tag{4}$$

$$\text{RC} = \left[\frac{\text{Tp}}{\text{Tp} + \text{FN}} \right] \tag{5}$$

Table 5 Classification result of proposed deep model on a prepared dataset

Disease	Predicted class						Accuracy (%)
	Canker	Scab	Melanosis	Greening	Black-spot	Healthy	
Canker	40	3	0	4	1	0	97
Scab	0	8	1	0	0	0	95
Melanosis	1	0	4	0	0	0	99
Greening	0	0	0	3	2	0	97
Black-spot	0	5	1	0	76	0	97
Healthy	0	0	0	3	6	152	97

Table 6 Class-wise precision, recall, specificity, and f-measure of the proposed model

Class	Precision (%)	Recall (%)	Specificity (%)	F-measure (%)
Canker	97.5	83.3	99.6	89.8
Scab	89	92.7	96	90.8
Melanose	66.7	80	33.3	66.9
Greening	30	60	97.7	40
Blackspot	80	88.9	97.3	84.2
Healthy	100	94.4	100	97

$$FS = \left[\frac{(2 * PC * RC)}{PC + RC} \right] \tag{6}$$

$$Acy = \left[\frac{Tp + TN}{Tp + FP + FN + TN} \right] \tag{7}$$

where TP: True positive, FP: False positive, TN: True Negative, FN: False Negative.

4 Experimental results and discussion

The model’s efficiency is assessed and calculated using the 80–20 cross-validation function. The performance of the classification model was evaluated using the cross-entropy loss function. Adam has been chosen for the pruning model, while SGD optimizer has also been adopted for the baseline model to enhance the cross-entropy parameter. The outcome of the proposed model on the citrus fruit’s image dataset is captured and represented in Table 5 as the Confusion matrix. F-measure specificity, recall, and precision, calculated for each class disease, are encapsulated in Table 6.

Canker performs the best out of the five diseases, with a precision of 97.5%, recall of 83.3%, specificity of 99.6%, and F-measure of 98.9%. Comparing the model with the other classes, Citrus scab had the lowest accuracy (95%). Because the signs generated by many diseases might be indistinguishable and can act synergistically, resulting in a low accuracy rate, at some point, the symptoms of a black spot and a scab may resemble one another, leading to a misdiagnosis. The model’s encouraging and supporting outcomes in identifying disorders from Citrus fruits samples show that DL approaches have a significant role in

Table 7 Comparison of different parameters before and after compression on the faster-CNN model

Evaluation parameter	Without pruning	50–80% pruning	60–90% pruning	70–90% pruning	80–90% pruning
Accuracy	96.92%	96.85%	96.03%	92.35%	80.36%
Loss	0.1817	0.1328	0.1557%	0.2067%	0.50%
Precision	96.07%	94.85%	96.05%	93.08%	86.41%
Recall	95.5%	94.85%	94.13%	92.35%	69.45%
F-score	95.78%	94.85%	92.71%	92.71%	77%
Size	53.78 MB	50.91 MB	28.16 MB	22.81 MB	17.33 MB
<i>Pruning + post-quantization</i>					
Size	14.11 MB	8.21 MB	8.23 MB	6.62 MB	4.81 MB
Accuracy	89.2%	83.3%	87.2%	78.8%	79.2%

disease detection and classification. A few research limitations can make a better analysis possible if more data can be gathered.

Once pruning finishes, the model’s effectiveness is evaluated by measuring how much performance and model size have changed compared to before pruning and quantization [33]. The baseline model is compared with the pruning and a post-quantization model at different sparsity levels, i.e. the starting level of sparsity begins at 50%, and sparsity ends at 80%. Likewise, results were compared with 60–90%, 70–90%, and 80–90% sparsity levels

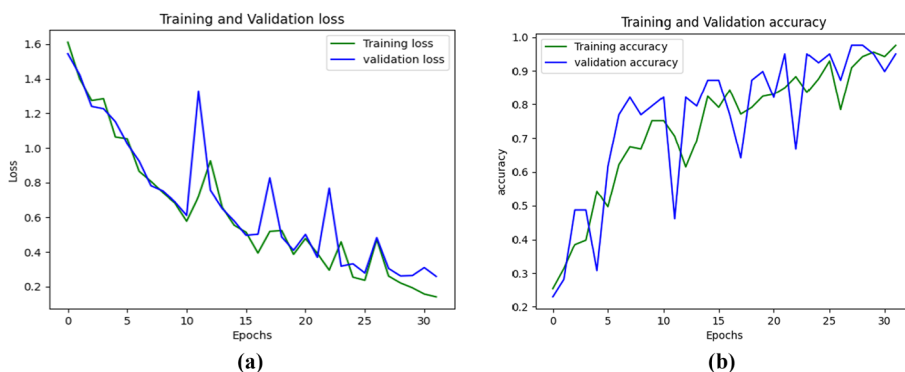


Fig. 5 Without pruning results of proposed faster-CNN model **a** training & validation loss **b** training and validation accuracy

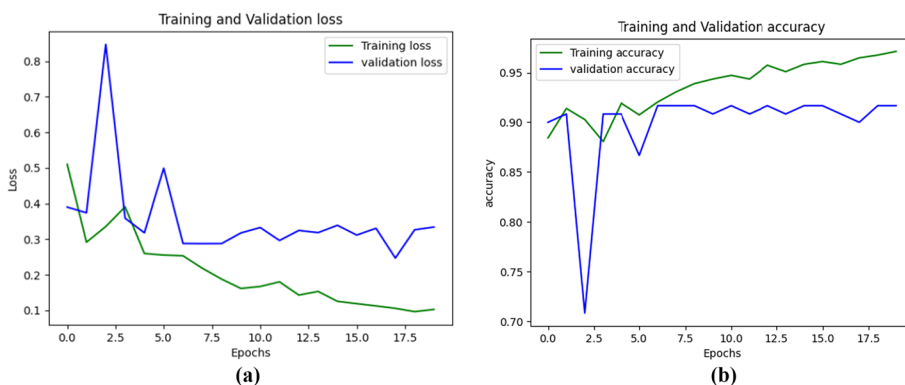


Fig. 6 With 50-80% pruning results of proposed faster-CNN model **a** training and validation loss **b** training and validation accuracy

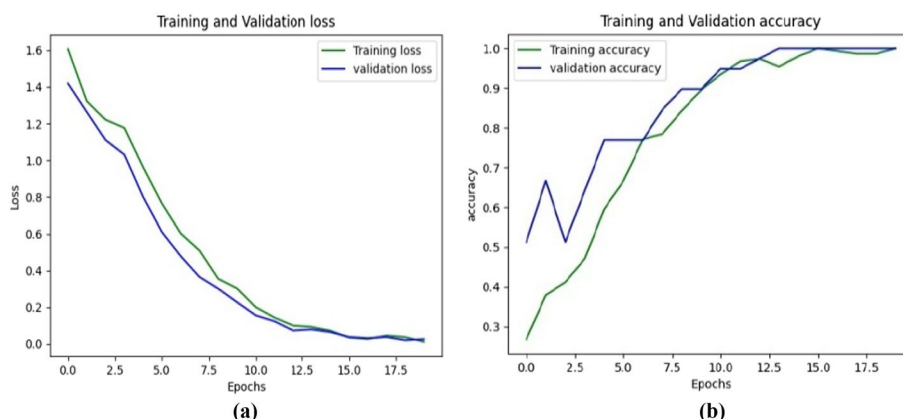


Fig. 7 With 60-90% pruning results of proposed faster-CNN model **a** training and validation loss **b** training and validation accuracy

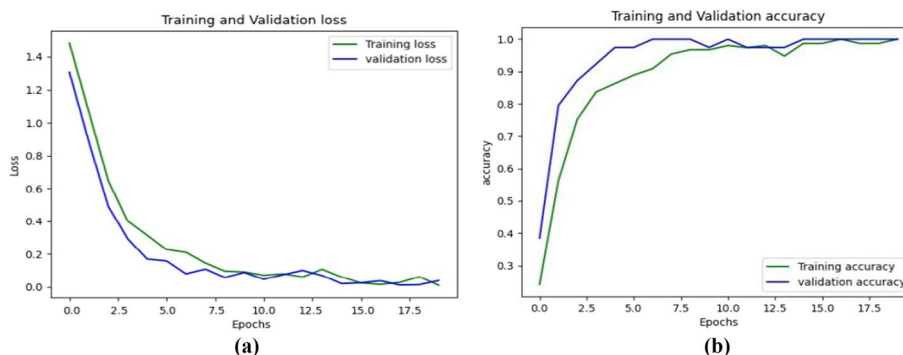


Fig. 8 With 70-90% pruning results of proposed faster-CNN model **a** training and validation loss **b** training and validation accuracy

using evaluation parameters like accuracy, loss, precision, recall, and size are represented in Table 7.

The accuracy and loss graphs are also plotted for the above-said models. It can be observed in loss graph Fig. 5a that the training and validation losses percentage of around 0.18 losses is there in the proposed Faster-CNN model without pruning represented in Fig. 5a, b shows training and validation graphs which is approaching 96.92%.

The loss graph in Fig. 6a represents that the training and validation losses are near 0.13 and approaching zero with model pruning of sparsity range of 50–80%, and the training and validation graph is coming to 96.03% with starting level of sparsity begins with 50%, and sparsity ends at 80% represented in Fig. 6b.

The loss graph Fig. 7a indicates that both the training and validation loss percentage are near 0.20, with model pruning having a sparsity range of 80–90% and the training and validation graph which is approaching 96.03% with starting level of sparsity begins with 80%, and sparsity ends at 90% represented in Fig. 7b.

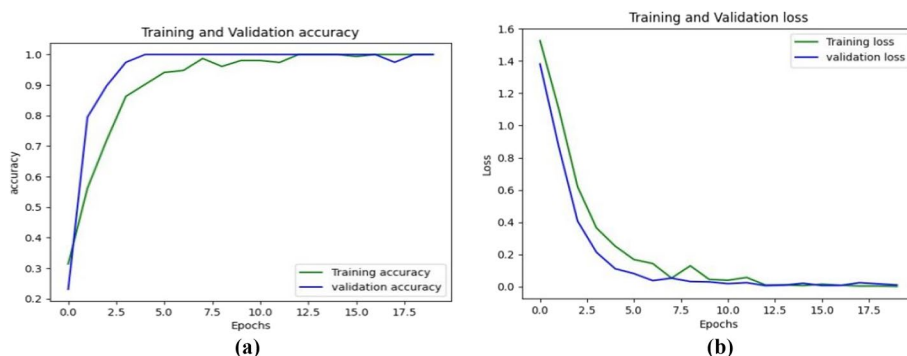


Fig. 9 With 80-90% pruning results of proposed faster-CNN model **a** training and validation loss **b** training and validation accuracy

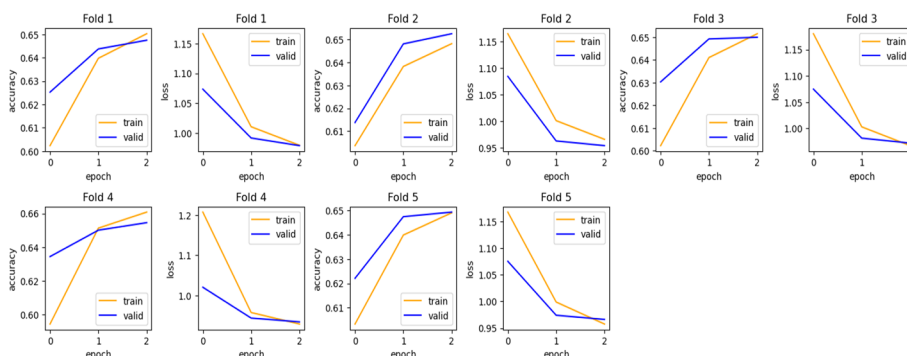


Fig. 10 Accuracy and loss results based on fold 1 to 5

Similarly, the loss graph indicates that a 15% of loss is encountered in both the training and validation loss, with model pruning having a sparsity range of 70–90% as mentioned in Fig. 8a and the training and validation graph, which is approaching to 92.35% with starting level of sparsity begins with 80% and sparsity ends at 90% represented in Fig. 8b.

The loss graph indicates that the training and validation losses deviate towards a loss percentage of around 0.50, with model pruning having a sparsity range of 80–90%, as mentioned in Fig. 9a, and the training and validation graph, which is approaching to 80.36% with starting level of sparsity begins with 80%. Sparsity ends at 90%, represented in Fig. 9b. It is observed that the training and validation accuracy is lowest and loss is higher in this scenario. Figure 10 shows the accuracy and loss results for proposed method for different folds i.e., fold one to fold five.

5 Conclusion and future work

Practical usages of resources are the primary issues MIC environment due to the devices with limited resources. As the high resource consumption for the process of the deep neural network, pruning and quantization turn to the edge network to handle their processing tasks. The paper presented the deep learning inference on low-resource devices and then outlined essential techniques for size compression, resulting in the fast execution of CNN on systems with fewer resources. Edge clusters

are groups of nodes with suitable performance standards, like pre-processing and deep learning techniques, that arise at the edge networks. The defect identification is made by implementing a Faster-CNN model to detect cankers' black spots, greening, scab, melanosis, and healthy Citrus fruits, as there is an over-parameterization issue of deep neural networks. Pruning techniques can eliminate a large percentage of network parameters while preserving accuracy.

This paper implements deep learning model pruning on the Faster-CNN model using Magnitude-based pruning with polynomial decay-based sparsity on the Faster-CNN model. The suggested model reduces memory usage and processing time while improving the resource efficiency of the DL techniques. The model attained an average accuracy of 96.92%, and intermediate precision, F-score, recall, and loss were 96.07%, 95.5%, 95.78%, 99.3%, and 0.1817, respectively, without using the pruning technique.

To assess the model's performance, the pruning process followed by quantization to compress the proposed Faster-CNN model that applied to a combined dataset collected from three different resources *www*, Kaggle, and plant village. Post-quantization, followed by pruning, compresses and deploys the model to a mobile or embedded application. On the other hand, the results indicate that by pruning 60–90% before post-quantization of the model with Magnitude-based pruning, including polynomial decay-based sparsity, it can compress the model by an average factor of 47 without considerably hampering the accuracy. Moreover, pruning followed by fixed-point weights quantization can further compress the size of the proposed model up to 8.23 MB without considerable accuracy loss. The findings are expected to promote pruning with quantization, a legitimate mechanism for compiling deep-learning complex models for implementation in scarce resource scenarios. When the edge network contains components of the same kind, the limitations of the work can be seen. This study can be expanded by implementing this experiment with real-time farming images.

Acknowledgements

The researchers would like to acknowledge Deanship of Scientific Research, Taif University for funding this work.

Author contributions

All the authors are equally contributed for the manuscript on revision stages and complete manuscript and also Fund and support for the research. All the authors helps in Prepare of first draft and also in review.

Funding

Open Access funding provided by the Qatar National Library. The researchers would like to acknowledge Deanship of Scientific Research, Taif University for funding this work.

Availability of data and materials

Please contact author for data requests.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Department of Higher Education, Government PG College, Ambala Cantt, India. ²Division of Information and Communication Technology, College of Science and Engineering, Hamad Bin Khalifa University, 602000 Doha, Qatar. ³Department of Computer Science and Engineering, Chandigarh University, Gharuan, Mohali, Punjab, India. ⁴Department of Computer

Science, College of Computers and Information Technology, Taif University, P. O. Box 11099, Taif 21944, Saudi Arabia. ⁵Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India. ⁶School of Creative Technologies, University of Bolton, Bolton, UK. ⁷Department of Computer Science, College of Computers and Information Technology, Taif University, P. O. Box 11099, Taif 21944, Saudi Arabia. ⁸Department of Information Technology, College of Computers and Information Technology, Taif University, Taif 21974, Saudi Arabia. ⁹College of Information Sciences and Technology, Data Science and Artificial Intelligence Program, Penn State University, State College, PA 16801, USA. ¹⁰School of Optometry and Vision Science, Faculty of Science, University of Waterloo, 200 University, Waterloo, ON N2L3G1, Canada. ¹¹Faculty of Engineering, University of Waterloo, 200 University Ave, Waterloo, Canada.

Received: 31 December 2022 Accepted: 25 May 2023

Published online: 22 June 2023

References

1. S. Mishra, T.H. Ayane, V. Ellappan, D.S. Rathee, H. Kalla, Avocado fruit disease detection and classification using modified SCA-PSO algorithm-based MobileNetV2 convolutional neural network. *Iran J. Comput. Sci.* **5**(4), 345–358 (2022)
2. A.O. Panhwar, A.A. Sathio, A. Lakhan, M. Umer, R.M. Mithiani, S. Khan, Plant health detection enabled CNN scheme in IoT network. *Int. J. Comput. Digit. Syst.* **11**(1), 344–335 (2022)
3. W. Zhang, Y. Liu, K. Chen, H. Li, Y. Duan, W. Wu, Y. Shi, W. Guo, Lightweight fruit-detection algorithm for edge computing applications. *Front. Plant Sci.* **12**, 2158 (2021)
4. M.A. Khan, T. Akram, M. Sharif, M. Awais, K. Javed, H. Ali, T. Saba, CCDF: automatic system for segmentation and recognition of fruit crop diseases based on correlation coefficient and deep CNN features. *Comput. Electron. Agric.* **155**, 220–236 (2018)
5. M. Cruz, S. Mafra, E. Teixeira, F. Figueiredo, Smart strawberry farming using edge computing and IoT. *Sensors* **22**(15), 5866 (2022)
6. M.A.R. Refat, S. Sarker, C. Kaushal, A. Kaur, M.K. Islam. WhyMyFace: A novel approach to recognize facial expressions using CNN and data augmentations, in *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2022, Volume 3*, pp. 553–563. Singapore: Springer Nature Singapore (2022).
7. A. El-Aziz, A. Atrab, A. Darwish, D. Oliva, and A.E. Hassanien. Machine learning for apple fruit diseases classification system, in *The International Conference on Artificial Intelligence and Computer Vision*, pp. 16–25. Springer, Cham (2020).
8. H.-Y. Hsu, G. Srivastava, Wu. Hsin-Te, M.-Y. Chen, Remaining useful life prediction based on state assessment using edge computing on deep learning. *Comput. Commun.* **160**, 91–100 (2020)
9. C. Kaushal, M.K. Islam, A. Singla, M.A. Amin, An IoMT-based smart remote monitoring system for healthcare. *IoT-Enabled Smart Healthc. Syst. Serv. Appl.* 177–198 (2022)
10. K. Elangovan, S. Nalini, Plant disease classification using image segmentation and SVM techniques. *Int. J. Comput. Intell. Res.* **13**(7), 1821–1828 (2017)
11. R.F. de Melo, G.L. de Lima, G.R. Corrêa, B. Zatt, M.S. de Aguiar, G.R. Nachtigall, R.M. Araújo, Diagnosis of apple fruit diseases in the wild with mask R-CNN, in *Brazilian Conference on Intelligent Systems*, pp. 256–270. Springer, Cham (2020)
12. A.G. Alharbi, M. Arif, Detection and classification of apple diseases using convolutional neural networks, in *2020 2nd International Conference on Computer and Information Sciences (ICIS)*, pp. 1–6. IEEE (2020)
13. H. Jiang, X. Li, F. Safara, IoT-based agriculture: deep learning in detecting apple fruit diseases. *Microprocess. Microsyst.* 104321 (2021)
14. H. Wang, Q. Mou, Y. Yue, H. Zhao, Research on detection technology of various fruit disease spots based on mask R-CNN, in *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1083–1087. IEEE (2020)
15. U.K. Lilhore, A.L. Imoize, C.C. Lee, S. Simaiya, S.K. Pani, N. Goyal, A. Kumar, C.T. Li, Enhanced convolutional neural network model for cassava leaf disease identification and classification. *Mathematics* **10**(4), 580 (2022)
16. B. Doh, D. Zhang, Y. Shen, F. Hussain, R.F. Doh, K. Ayepah, Automatic citrus fruit disease detection by phenotyping using machine learning, in *2019 25th International Conference on Automation and Computing (ICAC)*, pp. 1–5. IEEE (2019)
17. X. Chen, G. Zhou, A. Chen, Pu. Ling, W. Chen, The fruit classification algorithm based on the multi-optimization convolutional neural network. *Multimed. Tools Appl.* **80**(7), 11313–11330 (2021)
18. L. Jerlin Rubini, E. Perumal, Efficient classification of chronic kidney disease by using multi-kernel support vector machine and fruit fly optimization algorithm. *Int. J. Imaging Syst. Technol.* **30**(3), 660–673 (2020)
19. N.K. Trivedi, S. Simaiya, U.K. Lilhore, S.K. Sharma, COVID-19 pandemic: role of machine learning and deep learning methods in diagnosis. *Int J Curr Res Rev* **13**(06), 150–156 (2021)
20. M.A. Khan, T. Akram, M. Sharif, M. Alhaisoni, T. Saba, N. Nawaz, A probabilistic segmentation and entropy-rank correlation-based feature selection approach for the recognition of fruit diseases. *EURASIP J. Image Video Process.* **2021**(1), 1–28 (2021)
21. M. Poongodi, M. Malviya, C. Kumar, M. Hamdi, V. Vijayakumar, J. Nebhen, H. Alyamani, New York City taxi trip duration prediction using MLP and XGBoost. *Int. J. Syst. Assur. Eng. Manag.* **13**(1), 16–27 (2022)
22. M. Nikhitha, S. Roopa Sri, B. Uma Maheswari, Fruit recognition and grade of disease detection using inception v3 model, in *2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 1040–1043. IEEE (2019)
23. U.K. Lilhore, S. Simaiya, J.K. Sandhu, N.K. Trivedi, A. Garg, A. Moudgil, Deep learning-based predictive model for defect detection and classification in industry 4.0, in *2022 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pp. 1–5. IEEE (2022)
24. M. Poongodi, M. Hamdi, M. Malviya, A. Sharma, G. Dhiman, S. Vimal, Diagnosis and combating COVID-19 using wearable Ora smart ring with deep learning methods. *Pers. Ubiquitous Comput.* **26**(1), 25–35 (2022)
25. L. Pan, W. Zhang, Na. Zhu, S. Mao, Tu. Kang, Early detection and classification of pathogenic fungal disease in post-harvest strawberry fruit by electronic nose and gas-chromatography–mass spectrometry. *Food Res. Int.* **62**, 162–168 (2014)

26. U.K. Lilhore, U. Kumar, S. Simaiya, H. Pandey, V. Gautam, A. Garg, P. Ghosh, Breast cancer detection in the IoT cloud-based healthcare environment using fuzzy cluster segmentation and SVM classifier, in *Ambient communications and computer systems*, pp. 165–179. Springer, Singapore (2022)
27. C. Kaushal, M.K. Islam, S.A. Althubiti, F. Alenezi, R.F. Mansour, A framework for interactive medical image segmentation using optimized swarm intelligence with convolutional neural networks. *Comput. Intell. Neurosci.* (2022)
28. N. Kumari, R. Belwal, Hybridized approach of image segmentation in the classification of fruit mango using BPNN and discriminant analyzer. *Multimed. Tools Appl.* **80**(4), 4943–4973 (2021)
29. N.M. Ibrahim, D.G.I. Gabr, A.U. Rahman, S. Dash, A. Nayyar, A deep learning approach to intelligent fruit identification and family classification. *Multimed. Tools Appl.* **81**, 1–16 (2022)
30. H.T. Rauf, B.A. Saleem, M.I. Lali, M.A. Khan, M. Sharif, S.A. Bukhari, A citrus fruits and leaves dataset for detecting and classifying citrus diseases through machine learning. *Data Brief* **26**, 1043 (2019)
31. C. Kim, H. Lee, H. Jung, Fruit tree disease classification system using generative adversarial networks. *Int. J. Electr. Comput. Eng. (2088-8708)* **11**(3), 2508 (2021)
32. N.K. Trivedi, A. Anand, U.K. Lilhore, K. Guleria, Deep learning applications on edge computing, in *Machine Learning for Edge Computing*, pp. 143–168. CRC Press
33. Y. Ye, H. Zhou, H. Yu, R. Hu, G. Zhang, J. Hu, T. He, An improved efficientNetV2 model based on visual attention mechanism: application to identification of cassava disease. *Comput. Intell. Neurosci.* (2022)

Publisher's Note

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

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)
