Deep Image Processing Based Periodically Leaves Diseases Detection and Classification through Wireless Visual Sensors Network (WVSN)

Mazhar Noor [1], Naveed Abbas [1], Muhammad Wasim [1], Amerah Alghanim [2], Narmine ElHakim [2], and Amjad Rehman Khan [2,*]

[1] Department of Computer Science, Islamia College University Peshawar, Peshawar, Pakistan
[2] Artificial Intelligence and Data Analytics Lab, CCIS Prince Sultan University, Riyadh, Saudi Arabia
Email: mazharnoor.std@icp.edu.pk (M.N.); naveed.abbas@icp.edu.pk (N.A.); m.wasim@icp.edu.pk (M.W.); aghanim@psu.edu.sa (A.A.); nhakim@psu.edu.sa (N.E.); arkhan2030@gmail.com (A.R.K.)
*Corresponding author

Abstract—**Apples are one of the best sources of nourishment and are packed with various nutrients, including fiber, vitamins, minerals, and antioxidants which are essential for maintaining a healthy body and reducing the risk of chronic diseases. However, many diseases attack apple plants like “Scab”, “Rust”, and “Black rot”. These diseases are responsible for the decrease in the production and cultivation of apples. Identification of these diseases at an early stage can play an important role in their control before spreading into other parts of the plant. This job is challenging, especially in leaves even through an expert’s eye and many imaging methods are applied to identify these diseases from images using machine learning algorithms. To automate the process in real-time monitoring of Apple farms, this paper presents a framework for detecting diseases in apple leaves using a Wireless Visual Sensor Network (WVSN). A WVSN utilizes Convolution Neural Networks (CNN) for cloud-based classification. The framework will periodically capture the images directly from the apple farms through wireless nodes and send the data to the cloud through a gateway for further processing where the trained model classifies the diseases accurately. We tested our proposed model on the developed dataset to benchmark it against other state-of-the-art studies and subsequently deployed it in Apple farms to ensure the best results. The proposed framework gained an accuracy of 96.96% on the developed dataset and 95.1% in the real time with Apple farm images.**

Keywords—**agriculture, apple leaves diseases, real-time monitoring, wireless visual sensors network, rural economy, AlexNet**

I. Introduction

The wide usage of fruits in our daily life has increased its demand and apples are one of these. From juices to drugs and beauty products apples consumption is increasing day by day. Apple trees are cultivated around the world making it a valuable agricultural crop which is a source of income for many individuals and creates opportunities throughout the supply chain. They help economic growth by participating in global trade as a valuable export good. At the same time many diseases attack apple plants where some directly affecting the fruit, while others spread from the leaves to other parts of the plant. The cause of these diseases depends on different weather conditions [1]. These diseases cause a major effect on production loss, resulting in economic loss as well. Some of the common diseases that attack apple leaves are Apple Scab, Apple Black Rot, and Cedar Apple Rust.

Apple scab is caused by Venturia Inadequacies that has a catastrophic impact on the apples and mostly occurs in the Mediterranean regions where rainfall occurs frequently in spring [2]. Almost 16 to 20 times different types of fungicides are used to control this fungus that in severe cases resulted in 70% of production loss [3]. Black rot is caused by “Alternaria Alternata” which is a postharvest disease [4] and its symptoms are “frogeye” leaf spots that are small light brown lesions surrounded by dark brown borders [5]. Black rot also spread to trunk, branches and fruit and costs a whole apple tree [6]. Apple rust mostly appears on the upper surface of the leaf in the mid-summer, these are the bright red, orange, and yellow spots that develop on the apple leaf as depicted in Fig.1. This fungal disease spreads into nearby apple fields and can lead to premature defoliation as the signs and symptoms of this disease appear after the point at which treatment should have been started [7].

Fig. 1. Sample images from dataset.

Manuscript received October 30, 2023; revised November 17, 2023; accepted February 1, 2024, published July 23, 2024.

doi: 10.12720/jait.15.7.862-872

A. Problem Definition

It is a challenging task to recognize these diseases in the early stages as it affects the yield and the cost of production which suffer the farmers. Automating the detection of these diseases by using modern technologies like machine learning, Deep Learning (DL) algorithms and IOAT can help in detecting and accurate classification of these diseases at early stage and enhance the quality of agriculture as well as reduce the cultivation costs by avoiding pesticides and chemicals that are harmful to humans [8].

We proposed a real-time monitoring system (WVSN) to overcome this problem of disease spread. When they start appearing in the leaves. The network detects and classifies the disease category so that farmers will be able to take precautionary action and save the harvest. The following are the contributions of this research.

- Development of a synthetic dataset that consists of 20,000 images: 17,000 are diseased and 3,000 are healthy images.
- Presented a real-time monitoring system using WVSN.
- Proposed a framework to detect and classify apple leaves diseases.
- A cloud-based classification is carried out by sending real-time data directly from the farm.
- Performed multiple tests on state-of-the-art models with our dataset.

A Wireless Visual Sensor Network (WVSN) was implemented by Ali [9] to detect fungus and deficiencies in plant leaves in greenhouses. Using machine learning techniques, they distributed camera sensors throughout the greenhouse. When a fungus is detected the sensor node sends a message of humidity check through WVSN. Their approach obtained 94% success in the detection of fungus. Kamath [10] used WVSN for precision agriculture to monitor crop and detect the presence of weed immediately when they appear.

B. Objectives

- To gather and develop a synthetic dataset that consists of the most common apple leaves diseases.
- To develop a framework of Wireless Visual Sensors Network
- To analyze the pre-trained DL models with the developed datasets and framework.
- To propose a best-fit deep learning framework for detection and classification of Apple leaves diseases.

II. Literature Review

Image processing techniques are widely used in plant diseases detection over time and by advancements in “Neural networks” and “deep learning models” the accuracy of detection and classification has been comparatively enhanced. From traditional methods to Deep Neural Networks (DNN) various work is done in the agriculture sector for disease identification we have discussed some of these below.

Traditional machine learning algorithms are not the solution for modern-day problems as enhanced neural networks have overcome the limitations of ML algorithms. In recent years, there has been a major advancement in the development of techniques for evaluating leaf diseases. The following sections provide an overview of some significant contributions to this field’s study.

The research carried out by Zhang *et al.* [11]contained 90 samples of multi-class diseased apple leaves. Using a genetic algorithm to recognize patterns in apples their approach scored above 90% accuracy. They used the images for training models which were captured in a controlled environment, and their system cannot identify diseased apple leaves from real-time data. A leaf-disease detection solution was provided by Bashish and Braik *et al.* [12]. Their method consists of K-mean clustering in segmenting image and color transformation techniques in RGB images of diseased leaves [13], texture extraction, and finally pre-trained neural networks. An accuracy of 93% in detecting diseased apple leaves was achieved in the experiment. The accuracy of their model can be improved, and better segmentation methods can be developed.

Sannakki and Rajpurohit [14] devised a technique called “Classification of Pomeg Ranate Diseases Based on Back Propagation Neural Network”. They use color and texture as characteristics for classification after segmenting the defective area. For this, they used a neural network classifier. Their technique has the remarkable benefit of extracting the chromaticity layers by converting the image to Lab color space, which yields a 70% categorization accuracy. However, this strategy has the drawback of being limited to a few crops. Also, the algorithm achieved a less accuracy as compared to current advanced techniques of DL methods [15].

Revathi [16] used a typical recognition shallow architecture in their study, which involved collecting features using Histogram of Oriented Gradients (HOG) vectors and then classifying them using an SVM algorithm. The tests were carried out on two different leaf datasets: the Flavia leaf dataset and the Swedish leaf dataset. In their evaluation, the performance of the traditional method was compared to modern methods. They used a small dataset of Swedish leaf where the models work well but not tested in real-time scenario.

In Camargo’s study [17], cotton leaf images are used for detecting the damaged area by applying Enhanced Particle Swarm Optimization (EPSO). In their proposed method preprocessing is carried out on the initial dataset and then analysis was conducted based on extracting features like color, edge, and texture variance to correct skew divergence. The final data was passed to CIG-MRAN algorithm, Remarkably, the proposed methodology achieved an accuracy of 98% in classifying six types of diseases in cotton leaves. This study has some limitations, first the dataset they used for training and testing is not discussed well. Additionally, it lacks specific details about the algorithms or models employed. And lastly it is not used for real-time monitoring.

Camargo and Smith [17] used Maze plant images for recognizing disease to help farmers in automating the disease's detection process. They extracted features like texture, shape, and color for some of the common diseases such as "Brow stripe", "Downy Mildew", "Stem border", etc. Testing two algorithms for classification by extracting HIS values achieved recognizable accuracy. In the final step when detection and classification are carried out the framework sends the SMS alert to the farmer. The major drawback of these studies is using image recognition technology that can identify single object at a time. The real-time monitoring for better results has been ignored, which can be a practical application for diseases detection techniques.

Therefore, Researchers started use of various techniques of Deep Learning (DL) for detection of crop diseases. To achieve the intended results, most of the strategies have employed picture recognition with a classifier [18]. The authors of conducted a thorough survey to ascertain fruit type and calculate yields [19]. To make choosing the best crop detection system simple to realize and use, all available research has been summarized. Due to their ability to detect and identify things, the authors advise using neural networks. Furthermore, neural networks can identify and pick up basic properties from visual inputs like forms and patterns. The authors also suggest using transfer learning in primary layers as a fundamental method for determining the best parameters' weights by adjusting hyper-parameters, including activation function, momentum, initialization, and learning rate.

Ferentinos [20] used multiple CNN architectures which included Google Net, VGG, and AlexNet for pairing plant images to recognize patterns. They used a database that consists of 58 diseases and is available online. From multiples tests and fine-tuning the models the VGG outperformed the other algorithms and scored 99.53% of the highest accuracy. Although the models were trained from the laboratory and real cultivation conditions, the paper does not provide detailed validation results in real cultivation condition. Also, real-time monitoring can make it possible to overcome the spread of these diseases at early stages of growing plants.

Omrani and Elham *et al.* [21] applied K-mean clustering to identify infected regions in apple leaves images. They used an SVM classifier for some of the diseases "Alternaria, apple black spot, and apple leaf miner pest". In this study, SVM has the highest accuracy of 96%. Jesus [22] used improved convolution neural network for detecting apple leaf diseases. For detection they used some advanced methods like YOLO object detection and performed feature reuse combined with DensNet improved the detection accuracy. They tuned the kernel size by keeping dimensionality reduction of features and they scored 99% accuracy with detection speed of 280FPS. However, the model was trained for limited data and the method has more applications in video which they kept in future work.

Chen *et al.* [23] utilized MobileNet-V2 pre-trained on ImageNet as the backbone network to boost the learning capability for minute lesion features. They improved the model even further by including an attention mechanism to understand the importance of inter-channel relationships and spatial spots in the input information. During model training, they also optimized the loss function and used transfer learning twice. The suggested method outperformed previous state-of-the-art algorithms, attaining a remarkable average identification accuracy of 99.67% on a public dataset.

Sun *et al.* [24] proposed a real-time detector for apple leaf diseases called MEAN-SSD. They used annotated images from the AppleDisease5 dataset publicly available. By using Mean-block and Inception modules they constructed the Mobile End Apple Net base SSD algorithm. They constructed a light CNN architecture usable on mobile phones and their experiment scored an accuracy of 83.12% with a processing speed of 12.53 FPS. Ramacharan *et al.* [25] classified 3 forms of plant diseases and 2 insect damage by utilizing DL-CNNs algorithms such as Inception V3, KNN, and SVM. Their proposed framework scored an accuracy of 93%.

To recognize and categorize diseases on maize plant leaves, an EfficientNet-based technique is presented in [26]. Artificial Intelligence (AI) Challenger dataset and a few web photos of maize diseases are used to extract a minimal dataset sample. To prepare a sample dataset, images are first cleaned, and screened before being enhanced with translation, scaling, and rotation transformations. There are 9,279 total photos gathered, 6496 of which are used for training and 2,783 for testing. Based on the EfficientNet model, transfer learning is applied to increase recognition speed and accuracy. Comparing the proposed model to EfficientNet, VGG-16 [27], Inception-V3, and Resnet-50 [28], it achieves an accuracy score of 98.85%.

Jesus *et al.* [22] conducted a survey on monitoring quality assessment for wireless visual sensor networks, their work unifies and standardizes the existing knowledge on quality assessment in WVSN. It provides a fair comparison for different networks or same network in different implementations. They mentioned some open issues in WVSN that could lead to future research directions. They did not provide solutions or recommendations for addressing these issues and future research is needed to explore it.

Ali *et al.* [29] proposed proper placements for WVSN to cover more areas and get quality images with minimum number of cameras implemented. The study presents a mathematical formulation and optimal solution for WVSN cameras. They implemented this structure for plants monitoring in real-time. We will enhance that structure by adding a DL in the cloud storage and integrating it to WVSN.

According to these studies, deep learning has been used extensively in the field of apple diseases detection and good results have been obtained. Although real-time applications of these methods are still missing, which is of high value in reducing Apple disease and spread control. Therefore, we suggest a framework that is applicable to Apple farms in real-world scenario and perform monitoring of the whole process of the field.

III. MATERIALS AND METHODS

At the beginning of the fungus attack on apple leaves if accurate pesticides and chemicals are used it can stop its spread and can reduce the effect of loss by a huge margin, therefore considering the less and accurate use of chemicals can save health, the farmer's time, and the cost of the apple production. To make it possible we proposed a system considering the following characteristics.

- Real-time monitoring of apple leaves by using a Wireless Visual Sensor Network (WVSN)
- Sending these images using a gateway to the cloud for further processing.
- Training a lightweight model in the cloud that can identify each category with high accuracy.
- Prediction of the type of disease in the new data by using training data.

A. Dataset Collection

Multiple datasets are available online for plant leaf disease, but most of the models trained for the data contained a smaller number of images. The AI-Challenger-Plant-Disease-Recognition dataset contains 2,462 images of six apple leaf disease [30]. Other datasets collected for this research work were from Plant Pathology 2020 FGVC7 Kaggle competition [31] that consists of 1,821 portrait and landscape images of size 2,048×1,368 or 1,368×2,048 [32]. The open available dataset "plant village" [33] contains three types of disease in apple leaves which we also used for our work, in the scab category it has 454, black rot 496, 220 rust, and 1,316 healthy leaves images [34]. Some of the images used in this work were collected manually from google search and a personal dataset collected from the apple farm of the Agriculture University, Peshawar using a Canon Digital High-Definition Single-Lens Reflex camera. The size of each image is 5,132×3,418. 300 images of infected leaves and 100 healthy were collected from a research farm. Which contains 100 samples of each class. The remaining 400 images per class are filtered from the online datasets making 500 samples in each category. The data is increased by using Data-Augmentation techniques (Rotating, Scaling, Flipping, Translation and Brightness) shown in Table I. All the images are rotated at 150 angle, flipped by using horizontal and vertical flip technique, scaled 180%, Translated at TranslationX and TranslationY and brightened using iaa.addToBrightness (50).

To overcome the real-time monitoring issue of apple leaves diseases we implemented a Wireless Visual Sensor Network (WVSN) that will capture data in real-time and detect and classify any disease found in the leaf of the apples.

B. Design of the Proposed System

The WVSN based monitoring system in apple farms has three stages (1) Detection of infected apple leaf from the tree (2) capturing image of the infected leaf and sending it to the cloud via gateway (3) classification of the diseased occurred in the leaf.

TABLE I. DATASET DETAILS

Developed Dataset				
Source	**Original Images Per Category**	**Total Samples**	**Data after Augmentation**	**Total Dataset**
Online dataset (Infected)	400×3	1,200	Rotation, Flipping, scaling, zooming,	15,500
Healthy	400	400		2,000
Research Farm (Infected)	100×3	300		1,500
Healthy	100	100		1,000
		2,000		20,000

1) Detection of the disease

When a disease attacks the leaf, it starts appearing on the surface of the leaf slowly and the pixel values changes accordingly. Various detection techniques are being proposed in literature, the histogram and correlation features are used for fall detection in Veeraputhiran's study [35]. Also, the bounding box is a good option to use, and various techniques are adopted for it. For our framework we used the SSD model which has faster detection speed than R-CNN as discussed in Jiang's research [36].

2) Sendind data to cloud

To communicate with the server, the nodes are connected in the network using Reliable Asynchronous Image Transfer (RAIT) protocol that sends images to the cloud through the configured gateway. When the node detects the disease in the leaf and draws a box around the infected area and captures an image of the leaf by using detection method discussed in above section. The captured images are sent to the cloud server for further processing as illustrated in Fig. 3.

3) Classification of the disease

When the new image is received to the cloud server from the WVSN it classifies the disease type. The trained AlexNet model with 4 classes can correctly perform classification from the trained neurons. It returns the disease type and accuracy level comparing the result with other classes.

C. Pre-processing

The data collected for this research are from multiple resources that need to be cleaned and prepared to get better results. In this section we discuss the pre-processing steps applied to our data including resizing, noise and background removal, image enhancement and augmentation.

1) Image resizing

The dataset contains images of different sizes as they are collected from various sources. Due to this reason a pre-determined size is required for each frame to be executed. As shown in the input layer at Table II the image size is reduced to 227×227 pixels.

2) Cropping area of interest

We cropped the area of interest from the image and removed the unwanted parts to make the pictures more suitable for computation.

TABLE II. DIAGRAMMATIC VIEW OF PROPOSED MODEL

Layer		Kernels	Kernel Size	Stride	Feature Map	Activation
Input	Image	1	-	-	227×227×3	-
1	Convolution	96	11×11	4	55×55×96	ReLU
	Max Pooling	96	3×3	2	27×27×96	ReLU
2	Convolution	256	5×5	1	27×27×256	ReLU
	Max Pooling	256	3×3	2	13×13×256	ReLU
3	Convolution	384	3×3	1	13×13×384	ReLU
4	Convolution	384	3×3	1	13×13×384	ReLU
5	Convolution	256	3×3	1	13×13×256	ReLU
	Max Pooling	256	3×3	2	6×6×256	ReLU
6	FC	-	-	-	9216	ReLU
7	FC	-	-	-	4096	ReLU
8	FC	-	-	-	4096	ReLU
Output	FC	-	-	-	1000	SoftMax

3) Noise removal

To counter noisy images, we applied some filters like average, median, salt and pepper techniques and made the images clearer.

4) Background removal

Most of the images in our dataset contained the same background as they were gathered from directories but the images, we collected from the apple farm contained different backgrounds. So, we used software and manually removed background from those images.

D. Raspberry Pi Camera

A low power high-definition camera with a flat flexible cable is connected to the camera serial interface connector by using python APIs to get and process images as can be seen in Fig. 2. Covering the camera and sensor node is important as they can get damaged from wind/water or any likewise environment. The Raspberry Pi 3 model is interfaced with v1 pi as image sensor because the model B supports Ethernet and Bluetooth 4.0 along with wireless technologies. Bluetooth is the connectivity channel for nodes with the gateway. The base station or gateway is connected to Wi-Fi for cloud communication.

Fig. 2. Integrated camera with Raspberry Pi.

The Raspberry Pi camera comes with a flexible cable used to be connected to CSI connector with the following specifications. The still resolution is 5 mega pixels, Sensor is Omni Vision OV5647, Focal length of 3.60 mm ± 0.01, Horizontal FOV 53.50 ± 0.13 degrees, Vertical FOV 41.41 ± 0.11 degrees and image formats are JPEG (accelerated), RAW, GIF, PNG.RGB888. The signal latency of 1 seconds observed in the system while communicating to the gateway.

Pseudocode 1 gives general flow of the data over the framework. The first stage of initializing the variables then monitoring methods connectivity and detection case forwarding through the network is pictured.

Pseudocode 1: Generic Flow of Data

```
// Initialize variables
var wvsn_nodes = [] // List of WVSN nodes
var gateway = CloudGateway () // Cloud gateway object
var trained_model = TrainedModel () // Trained model
object
// Main loop
while True:
  // Monitor apple farm with WVSN nodes for node in
    wvsn_nodes:
      data = node.get_data () // Get data from node
      // Check for diseased leaf
      if data.is_diseased_leaf ():
        // Send data to cloud for classification
        gateway.send_data (data)
  // Process data in the cloud
  if gateway.has_data ():
      data = gateway.get_data () // Get data from
gateway
      // Classify disease using trained model
      disease = trained_model.classify (data)
      // Return detected disease to WVSN node
      gateway. send_disease_to_node (disease,
      data.node_id)
```

Figs. 2 and 3 shows the complete framework of our proposed work, The WVSN collects images from the Apple farm on multiple nodes and sends these images to the cloud via a gateway installed in the farm for communication. These images are then processed and passed to the models trained on a cloud network that detect 3 types of diseases in this prior experiment of our project which is Apple scab, Apple Rust and Black Rot, and a healthy class [37]. The image sent by WVSN is classified into either one class and returned with the output whether it is a healthy leaf or a diseased one.

In the cloud, experiments were carried out on different models but before the training model we needed a good dataset to feed these models with more features to learn. Images for these models were collected from multiple resources, some datasets available in different repositories like Kaggle etc. were mixed with Google search results and other online resources. 500 images in each category were filtered with the healthy class which in total makes the dataset of 2,000 samples including the dataset collected from the research farm.

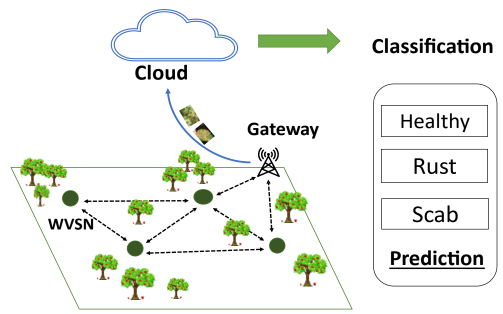

Fig. 3. Illustration of WVSN, sending data from field to cloud for classification.

The deep learning models used in this study are VGG-16, 2D CNN, and AlexNet. The first step of pre-processing techniques used on the dataset is to make it clean and easy for the model to learn more patterns. The pre-processing steps include scaling the images to a specific size and other augmentations are performed to rotate and flip these images for high learning features and duplicate the images in multiple angles, then zooming is applied with wide levels to make the model more durable and see the picture as humans see it from different angles and the same way Translation is used.

Once the pre-processing is completed, the data passes through the models one by one, with each model having different layers and parameters. These models are tuned for 40 epochs each as after multiple tests we noticed that all models were not converging after an average of 40 rounds. These Deep Neural Networks, when learning features and saving the weight of the parameters in the training span are then tested and evaluated with the new data to check their prediction and accuracy level. Each model is passed through the evaluation step with different techniques as shown in Fig. 4.

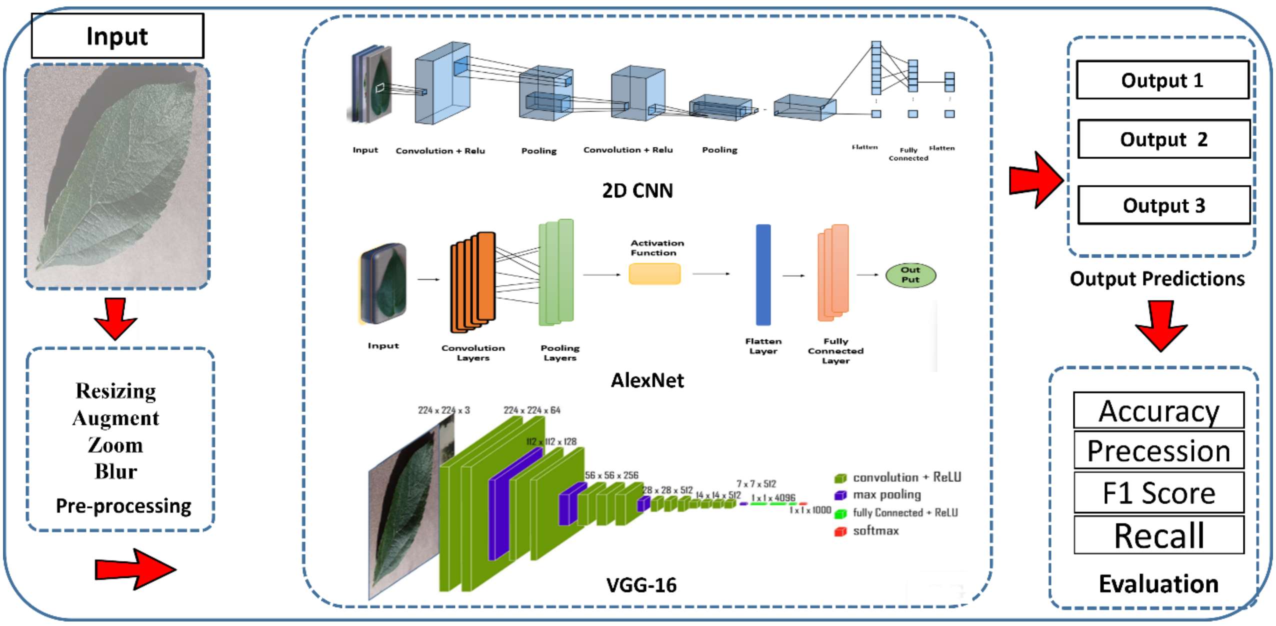

Fig. 4. Framework for multiple model's test and classification over cloud.

CNN is widely used in image processing as it is an easy-to-use feature extractor and is applied to each pixel region. It relies on the convolution layer that contains different numerical operations like convolution. The pixel values are contained in a 2D grid which is made up of parameters grid kernel. The outcomes of one layer are copied to the next convolution layer that produces more complex features. Using outcome algorithms like gradient decrease and backpropagation it gives the difference between real and predicted data.

Some of the strengths and limitations of these models are follows. Alex Net has performed classification of images very efficiently and widely used by researchers, also it is a faster algorithm in computation and memory efficient. The only limitation it carries is its application of high-resolution images. While the VGG-16 is best for small classification tasks and performed more accuracy in transfer learning, the reason of its inclusion in our research was the same task of transfer learning approach. The disadvantages it has are large numbers of parameters that increase computation cost and more prone to Vanishing Gradients problem.

The models were evaluated based on accuracy, precision tests, F1-Scores, and recall. The dataset was used in half for every model in the first test to check the time it takes to execute the dataset while in the second test, the whole dataset is passed to the model. The aim was to check which model fits well on the dataset we have collected and give better results so that we can adopt that model as our proposed for the classification in this project.

IV. Result and Discussion

We used each model for 2 iterations for a better understanding of time complexity and system requirements. The dataset we developed contains 20,000 images of four categories, Apple scab, Apple Rust, Apple Black Rot, and Healthy. The framework is executed in

Python 3 on a personal laptop in Windows 10 with dual processors of 1.6 and 2.3 GHz and RAM of 8 GB and an NVIDIA GEFORCE Graphics card. We also took some tests on Google Collab GPUs to save time in large-number epoch testing.

A. *Evaluation Parameters*

The model's performance was evaluated using Precision, Recall and F-measure parameters. The Eq. (1) gives accuracy of the classification where all the correct classifications are measured with accuracy. The correct predictions are divided by the total number of observations. Eq. (2) shows the recall and Eq. (3) represents precision parameters.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP+FP} \tag{2}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3}$$

where TP stands for True Positive, TN represents True Negative, FP means False Positive and FN represents False Negative. The details of these parameters are provided in Table III.

To adopt a model that is executable on a PC, we performed dual tests of the algorithms with small and larger datasets. Multiple epoch tests were performed to see the model's convergence on different levels. The data was split into 70% training and 30% for testing. For higher epoch tests we used Google Collab to save time and computation. But we noticed that our models were not learning more features from above 35 epochs, so we considered 40 epochs for all the models. From the results of testing all the models where Alex-Net outperformed the other two models and we considered it our best fit for the WVSN data. The proposed model is provided in Table II. The tested model's structure and their performance along with their predictions are discussed in the below section.

TABLE III. EVALUATION PARAMETERS VALUES

	Disease	Precision	Recall	F1-Score
2D-CNN	Scab	0.88	0.43	0.47
	Rot	0.59	0.80	0.71
	Rust	0.64	0.98	0.67
	Healthy	0.89	0.71	0.82
VGG-16	Scab	0.51	0.67	0.58
	Rot	0.62	0.56	0.59
	Rust	0.56	0.36	0.44
	Healthy	0.58	0.61	0.59
Alex Net	Scab	0.94	0.90	0.91
	Rot	0.86	0.99	0.97
	Rust	0.94	0.94	0.93
	Healthy	0.96	0.89	0.95

B. *2D-CNN*

The 2D-CNN, we used in this work contains six convolutions and three fully connected layers. The image size in the first convolution layer is (128×128×3), in the second conversion it reduces to 64, and in the third convolution 32. The kernel size used is 5×5 for the first and third, while the second convolution layer had a kernel of 3×3 and the activation function ReLU was the same for all. It started with an accuracy of 25% in training and in the first iteration with the half dataset it scored 56% accuracy while in the second test for the complete dataset, its accuracy increased a little to 63%. Fig. 1 includes the sample images of the dataset. The accuracy achieved with 2D-CNN was not up to the mark and reliable as it was not good on validation data, graphs can be seen in Fig. 5 illustrating that on testing data it was going towards overfitting, while Fig. 6 shows its confusion matrix.

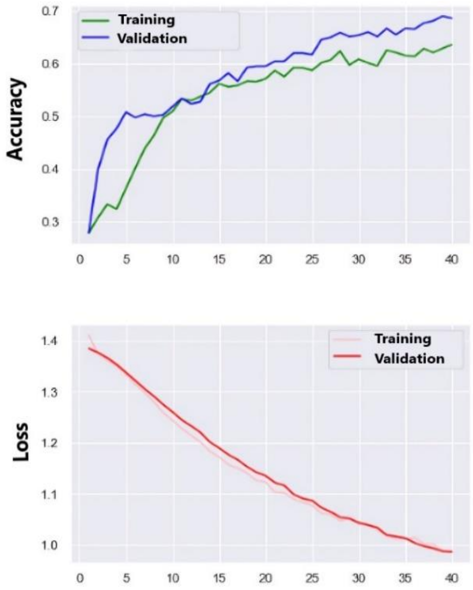

Fig. 5. Training and validation accuracy of 2D-CNN.

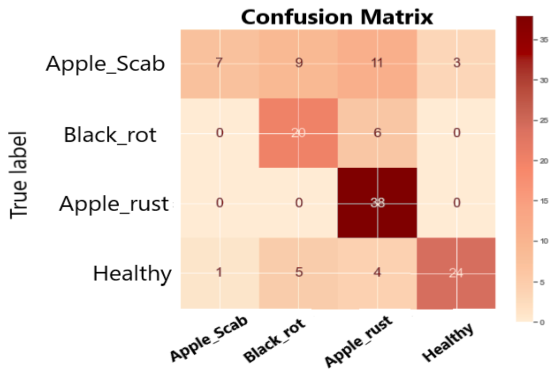

Fig. 6. Confusion matrix obtained from 2D-CNN.

The model conversion at the start was good enough on testing data which continued to its maximum of 63 almost but on testing data, it goes high over 65, one reason for this scenario is the data we provided for validation was from the same dataset so for the testing purposes, we used real-time images from the field, but still didn't get the desired results. To identify four classes or categories of diseases we trained the model which gives almost better results. Fig. 7 shows the output of the 2D-CNN in prediction, where each class has been labeled with the category predicted by the model and accuracy in that specific category.

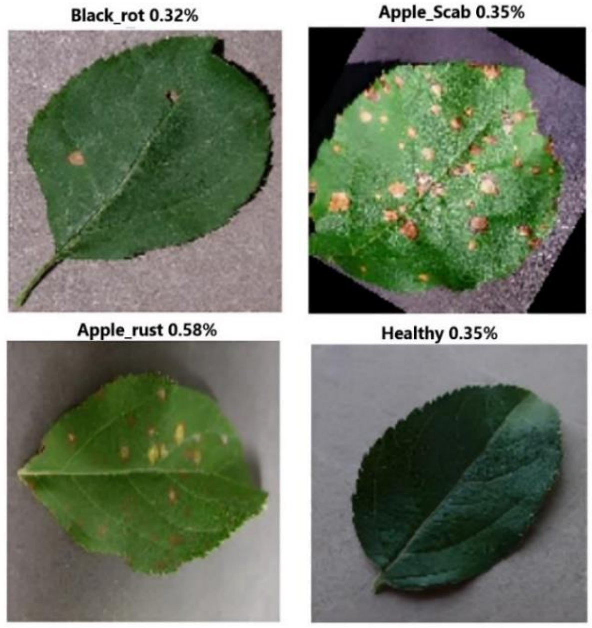

Fig. 7. Predicted output of 2D-CNN.

The maximum accuracy scored by any class prediction by 2D-CNN is 58% in the rust class. The lowest it scored was 32% in the apple black rot class. The overall prediction was correct by the model, but on the same data, it was not predicting the correct class to which the image belongs. 9 images in the black-rot category 11 in rust and 3 in healthy considered apple scab. In the same way, one scab, 5 rot, and 4 rust were considered healthy.

C. VGG-16

The architecture of VGG-16 was pre-tuned, and we used the default parameters defined for this model. The model consists of the pre-trained architecture of 13 layers of convolution and pooling while three are fully connected and one output layer. The accuracy and loss graph illustrates the result obtained from VGG-16 was not required, it failed on both training and validation data, from the accuracy graph it can be seen that on validation the accuracy goes higher, and the loss achieved in this model is too much, going above 1 which is not acceptable. The image provided to this model in the 1st convolutional layer is of size 224×224×3, at the start it gained 10% accuracy and converged better for the first 20 epochs after which its accuracy goes higher on testing data and did not perform well for the rest of the training. The maximum accuracy scored by this model is 47% on training data and 50% in testing as shown in Fig. 8. For the first iteration with half of the dataset, the model was not converging for more than 20% accuracy level that's where we noticed that VGG-16 is not a suitable algorithm for this problem. This model predicted 26% accurate results in Black-rot, 30% Scab, and 29% in Rust and Healthy. Fig. 9 shows confusion matrix obtained from VGG-16.

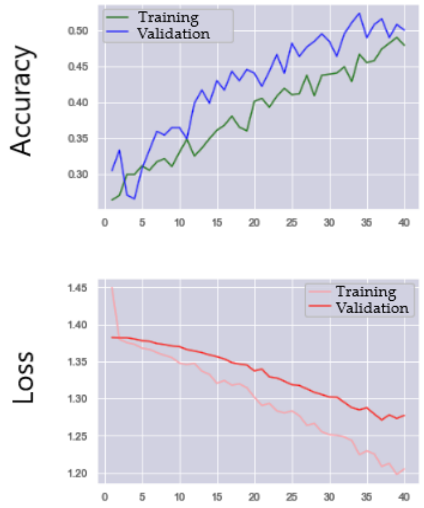

Fig. 8. Training and validation accuracy of VGG-16.

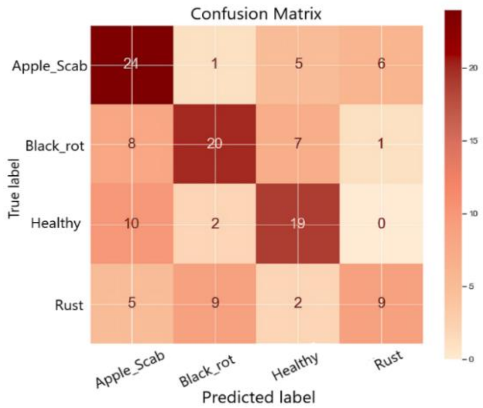

Fig. 9. Confusion Matrix Obtained from VGG-16.

D. Alex-Net (Proposed)

The performance of the Alex Net model in this project outperformed the other two algorithms we tested earlier, we tested this model twice once on half data and 20 epochs, and for the second test, we increased the number of epochs to 40. Its accuracy changed a little bit with the full dataset but was almost the same for 40 epochs on both data. The predefined model consists of 5 convolution layers and 3 dense with one output layer. At the start of the training this model its accuracy went from 61% in the first epoch and continued to increase in each epoch while completing all 40 it gained the highest accuracy of 91% in training and validation data as depicted in Fig. 10. From the loss graph, we can see it came to 0.2% in validation. The results of Alex Net crossed both other models and we achieved the accuracy level that we were performing tests for. Images of size 227×227×3 were given in the first convolution layer followed by a Max pooling layer. Each convolution is followed by pooling layers, and the kernel size for each convolution varies. The activation function used is ReLU and for the output layer, SoftMax is given as activation. The complete architecture is provided in Table II. The predicted images for validation are shown in Fig. 11 for each category and their accuracy level on the top.

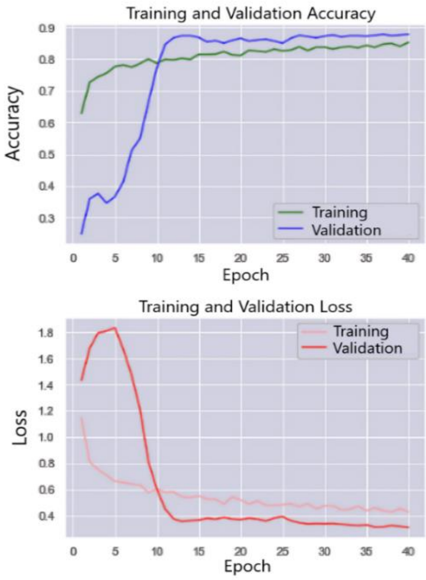

Fig. 10. Accuracy and loss graph of Alex-Net.

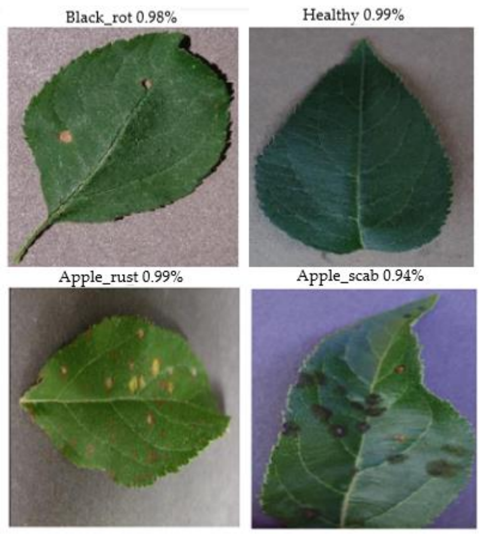

Fig. 11. Predicted results of Alex-Net.

From the visual result displayed by Alex Net, we can see that almost all the categories predicted with up to 99% accuracy on testing/validation data. Although Alex Net is a complex model and requires time to train, we performed the 20 epochs test on our personal laptop and for more testing with larger epochs and datasets Google Colab is used.

Fig. 12 shows the confusion matrix of the Alex Net model to see the classification results given in the real-time scenario. Here we can get from the graph that Alex Net correctly classified almost 90% of the categories. The missing is minor in scab where 2 are classified as scab and their true label is healthy class, same happened with one in healthy as Black_rot while in rust 2 are classified as rust which is a little bit complex and in terms of similarities in rust and rot images of apple leaf. The overall performance of the AlexNet model is commendable, as it demonstrates high accuracy in testing data/validation. Therefore, we propose this algorithm as the preferred model for the detection and classification of apple leaf images belonging to four different classes. These images are transmitted through a wireless visual sensor network to the cloud for further processing. In the cloud server, the trained Alex Net model will take the data sent from WVSN and make a prediction based on trained data to which class it belongs.

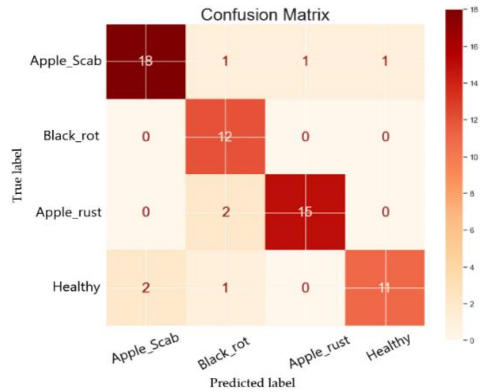

Fig. 12. Confusion matrix obtained from Alex-Net.

The reason why Alex-Net outperformed the other two models is their limitations. As discussed earlier that VGG-16 has the drawback of not working well with larger datasets and work well for small classifications. It was included because many researchers had adopted this algorithm for leaf diseases classifications as mentioned in the literature. Furthermore, our aim was to check these pretrained models on our framework and verify their results with our dataset. The Alex-Net with its deeper architecture and more learnable parameters allows it to learn complex features related to apple leaves diseases detection and classification as compared to 2D-CNN because of the less convolutions and few layers of 2D-CNN.

V. Conclusion

This study proposed a framework based on deep learning and Wireless Visual Sensor Network (WVSN) for actively monitoring apple farm in real time. The datasets used in this work were collected from multiple online repositories and some of the images were collected from the apple farm with a camera. A synthetic dataset is developed for training the models. This research focused on four classes in which three classes are diseased apple leaves namely “Apple Scab”, “Apple Rust”, “Apple Black Rot” and one “Healthy”. A (WVSN) node is installed in the apple farm for capturing real-time data by using a Raspberry pi camera that sends the real-time data to cloud for detection/classification.

Three different models were tested and evaluated for accurate detection and classification of these diseases in the cloud (“2D-CNN”, “VGG-16”, and “Alex-Net”). The models were tested in multiple iterations by providing half and complete dataset. For each iteration the epoch count was settled to 40 as the models were not converging after that. The result achieved from these models were 63% from 2D-CNN, 50% from VGG-16, and 91% by Alex-Net. Based on evaluation and accuracy Alex-Net is proposed as a best fit model for detection and classification of apple leaves diseases in the WVSN setup. In Fig. 13 the accuracy parameters of all models are illustrated.

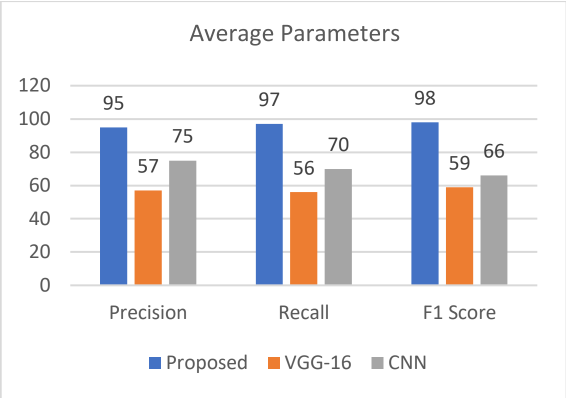

Fig. 13. Accuracy parameters of models.

A. *Real World Applications*

This framework can be used in smart agriculture systems for early detection and classification of apple leaf diseases that will allow farmers to take prompt action to control disease spread and minimize crop loss.

B. *Limitations and Future Work*

The limitation in this work includes lack of multi diseased leaves. In the future, we aim to increase the number of classes and multiple fruits along with vegetable diseases to be monitored in real-time. As the proposed deep learning model achieved good accuracy in detecting apple leaf diseases of 4 classes. However, further optimization of the model and hyperparameters may aid in detection of more diseases if provided with a good and clean dataset. Researchers can fine tune the model for multiple classes diseases of multiple plants and train it for detection and classification.

We will also include some other IOT sensors to check humidity and soil conditions. As the WVSN can actively monitor crops in real-time so these sensors can collect data from the farms directly and send a message when there is need of an action to perform.

Furthermore, the deployment and integration of our model on mobile devices is in the pipeline of future work. Taking humans in the loop like experts for feedback of the model's detection performance can help in providing a user-friendly interface to farmers where they can just simply scan any norm they found in the farm and send application sends the data to cloud for detection and classification.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, M.N., N.A, ARK; methodology, M.N., N.A, M.W., A,A, ARK; software, M.N., N.A, ARK; validation, M.N., N.A, M.W.; writing—original draft preparation, M.N., N.A, M.W., A,A, ARK; writing—review and editing, M.N., N.A, A.R.K, visualization, M.W., A,A, ARK; supervision, N.A, A,A, ARK; project administration, M.W., A,A, ARK; funding, ARK, A.A; all authors had approved the final version.

Acknowledgment

This research is supported by Artificial Intelligence and Data Analytics Lab (AIDA) CCIS Prince Sultan University, Riyadh, 11586, Saudi Arabia.

The authors would also like to thank Prince Sultan University, Riyadh Saudi Arabia for the support of APC of this publication.

References

[1] I. Ali, Z. Mushtaq, S. Arif, A. D. Algarni, N. F. Soliman, and W. El-Shafai, "Hyperspectral images-based crop classification scheme for agricultural remote sensing," *Comput. Syst. Sci. Eng.*, vol. 46, pp. 303–319, 2023.

[2] R. L. Ibrahim, Z. Yu, S. Hassan, K. B. Ajide, M. Tanveer, and A. R. Khan, "Trade facilitation and agriculture sector performance in Sub-Saharan Africa: Insightful policy implications for economic sustainability," *Frontiers in Environmental Science*, vol. 10, 962838, 2022.

[3] Y. J. Amuda, "Evaluation of agricultural policies and programmes for sustainable future farming intensification in Nigeria," *International Journal of Service Science, Management, Engineering, and Technology (IJSSMET)*, vol. 13, no. 1, pp. 1–13. 2022.

[4] Y. Ge *et al.*, "Effect of trisodium phosphate treatment on black spot of apple fruit and the roles of anti-oxidative enzymes," *Physiol. Mol. Plant Pathol.*, vol. 106, pp. 226–231, Apr. 2019. doi: 10.1016/J.PMPP.2019.03.003

[5] Warmund and Michele. (March 2023). Disease-resistant apple cultivars. *University of Missouri-Columbia* [Online]. Available: https://mospace.umsystem.edu/xmlui/handle/10355/91201

[6] M. Grabowski. (2019) Black rot of apple. *UMN Extension* [Online]. Available: https://extension.umn.edu/plant-diseases/black-rot-apple

[7] D. Strickland, J. Carroll, and K. Cox1. (March 2020). Cedar apple rust. *New York State Integrated Pest Management Program* [Online]. Available: https://ecommons.cornell.edu/handle/1813/43082.2

[8] S. S. Kumar and B. K. Raghavendra, "Diseases detection of various plant leaf using image processing techniques: A review," in *Proc. 2019 5th International Conference on Advanced Computing and Communication Systems, ICACCS 2019*, 2019, pp. 313–316. doi: 10.1109/ICACCS.2019.8728325

[9] A. Ali and H. S. Hassanein, "A fungus detection system for greenhouses using wireless visual sensor networks and machine learning," in *Proc. 2019 IEEE Globecom Workshops*, 2019. doi: 10.1109/GCWKSHPS45667.2019.9024412

[10] R. Kamath, M. Balachandra, and S. Prabhu, "Raspberry Pi as visual sensor nodes in precision agriculture: A study," *IEEE Access*, vol. 7, pp. 45110–45122, 2019. doi: 10.1109/ACCESS.2019.2908846

[11] C. Zhang, S. Zhang, J. Yang, Y. Shi, and J. Chen, "Apple leaf disease identification using genetic algorithm and correlation based feature selection method," *International Journal of Agricultural and Biological Engineering*, vol. 10, no. 2, 2017. doi: 10.3965/j.ijabe.20171002.2166

[12] D. A. Bashish, M. Braik, and S. Bani-Ahmad, "Detection and classification of leaf diseases using K-means-based segmentation and neural-networks-based classification," *Information Technology Journal*, vol. 10, pp. 267–275, 2011.

[13] K. P. Sinaga and M. S. Yang, "Unsupervised K-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80716–80727, 2020. doi: 10.1109/ACCESS.2020.2988796

[14] S. R. Maniyath *et al.*, "Plant disease detection using machine learning," in *Proc. 2018 International Conference on Design Innovations for 3Cs Compute Communicate Control, ICDI3C 2018*, 2018, pp. 41–45. doi: 10.1109/ICDI3C.2018.00017

[15] T. Q. Bao, N. T. T. Kiet, T. Q. Dinh, and H. X. Hiep, "Plant species identification from leaf patterns using histogram of oriented gradients feature space and convolution neural networks," *Journal of Information and Telecommunication*, vol. 4, no. 2, pp. 140–150, 2019. doi: 10.1080/24751839.2019.1666625
[16] P. Revathi, M. Hemalatha, "Classification of cotton diseases using cross information gain_minimal resource allocation network classifier with particle swarm optimization," *Journal of Theoretical and Applied Information Technology*, vol. 60, no. 1, pp. 184–188, 2022.
[17] A. Camargo and J. S. Smith, "An image-processing based algorithm to automatically identify plant disease visual symptoms," *Biosyst. Eng.*, vol. 102, no. 1, pp. 9–21, 2009. doi: 10.1016/J.BIOSYSTEMSENG.2008.09.030
[18] J. Ahmad *et al.*, "Visual features based boosted classification of weeds for real-time selective herbicide sprayer systems," *Comput. Ind.*, vol. 98, pp. 23–33, Jun. 2018. doi: 10.1016/J.COMPIND.2018.02.005
[19] A. Koirala, K. B. Walsh, Z. Wang, and C. McCarthy, "Deep learning—Method overview and review of use for fruit detection and yield estimation," *Comput. Electron. Agric.*, vol. 162, pp. 219–234, Jul. 2019. doi: 10.1016/J.COMPAG.2019.04.017
[20] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agric.*, vol. 145, pp. 311–318, Feb. 2018. doi: 10.1016/J.COMPAG.2018.01.009
[21] E. Omrani and B. Khoshnevisan *et al.*, "Potential of radial basis function-based support vector regression for apple disease detection," *Elsevier*, vol. 55, pp. 512–519, 2022.
[22] T. C. Jesus, D. G. Costa, P. Portugal, and F. Vasques, "A survey on monitoring quality assessment for wireless visual sensor networks," *Future Internet*, vol. 14, no. 7, 213, Jul. 2022. doi: 10.3390/FI14070213
[23] J. Chen, D. Zhang, A. Zeb, and Y. A. Nanehkaran, "Identification of rice plant diseases using lightweight attention networks," *Expert. Syst. Appl.*, vol. 169, 114514, 2021. doi: 10.1016/J.ESWA.2020.114514
[24] H. Sun *et al.*, "MEAN-SSD: A novel real-time detector for apple leaf diseases using improved light-weight convolutional neural networks," *Comput. Electron. Agric.*, vol. 189, 106379, 2021. doi: 10.1016/J.COMPAG.2021.106379
[25] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes, "Deep learning for image-based cassava disease detection," *Front Plant Sci.*, vol. 8, 1852, Oct. 2017. doi: 10.3389/FPLS.2017.01852/BIBTEX
[26] J. Liu, M. Wang, L. Bao, and X. Li, "EfficientNet based recognition of maize diseases by leaf image classification," *J. Phys. Conf. Ser.*, vol. 1693, no. 1, 012148, Dec. 2020. doi: 10.1088/1742-6596/1693/1/012148
[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd International Conference on Learning Representations, ICLR 2015*, 2015.
[28] K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
[29] A. Ali and H. S. Hassanein, "Optimal placement of camera wireless sensors in greenhouses," in *Proc. ICC 2021—IEEE International Conference on Communication*, Jun. 2021, pp. 1–6. doi: 10.1109/ICC42927.2021.9500727
[30] Y. Zhong and M. Zhao, "Research on deep learning in apple leaf disease recognition," *Comput. Electron Agric.*, vol. 168, 105146, Jan. 2020. doi: 10.1016/J.COMPAG.2019.105146
[31] R. Thapa, K. Zhang, N. Snavely, S. Belongie, and A. Khan, "The plant pathology challenge 2020 data set to classify foliar disease of apples," *Appl. Plant Sci.*, vol. 8, no. 9, Sep. 2020. doi: 10.1002/APS3.11390
[32] A. Yadav, U. Thakur, R. Saxena, V. Pal, V. Bhateja, and J. C. W. Lin, "AFD-Net: Apple foliar disease multi classification using deep learning on plant pathology dataset," *Plant Soil*, vol. 477, no. 1–2, pp. 595–611, Aug. 2022. doi: 10.1007/S11104-022-05407-3/FIGURES/15
[33] D. P. Hughes and M. Salathe, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," arXiv preprint, arXiv:1511.08060, 2015.
[34] M. Agarwal, R. K. Kaliyar, G. Singal, and S. K. Gupta, "FCNN-LDA: A faster convolution neural network model for leaf disease identification on apple's leaf dataset," in *Proc. 2019 International Conference on Information and Communication Technology and Systems*, 2019, pp. 246–251. doi: 10.1109/ICTS.2019.8850964
[35] A. Veeraputhiran and R. Sankararajan, "Feature based fall detection system for elders using compressed sensing in WVSN," *Wireless Networks*, vol. 25, no. 1, pp. 287–301, Jan. 2019. doi: 10.1007/S11276-017-1557-3/METRICS
[36] P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, "Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks," *IEEE Access*, vol. 7, pp. 59069–59080, 2019. doi: 10.1109/ACCESS.2019.2914929
[37] H. Ayaz, E. Rodríguez-Esparza, M. Ahmad, D. Oliva, M. Pérez-Cisneros, and R. Sarkar, "Classification of apple disease based on non-linear deep features," *Applied Sciences 2021*, vol. 11, no. 14, 6422, 2021. doi: 10.3390/APP11146422