Classification of Apple Tree Leaves Diseases using Deep Learning Methods

Ashwaq Alsayed, Amani Alsabei, Muhammad Arif
s44280168@st.uqu.edu.sa, s44280161@st.uqu.edu.sa, mahamid@uqu.edu.sa
Computer Science Department, Umm Al-Qura University, Makkah, Saudi Arabia

Summary
Agriculture is one of the essential needs of human life on planet Earth. It is the source of food and earnings for many individuals around the world. The economy of many countries is associated with the agriculture sector. Lots of diseases exist that attack various fruits and crops. Apple Tree Leaves also suffer different types of pathological conditions that affect their production. These pathological conditions include apple scab, cedar apple rust, or multiple diseases, etc. In this paper, an automatic detection framework based on deep learning is investigated for apple leaves disease classification. Different pre-trained models, VGG16, ResNetV2, InceptionV3, and MobileNetV2, are considered for transfer learning. A combination of parameters like learning rate, batch size, and optimizer is analyzed, and the best combination of ResNetV2 with Adam optimizer provided the best classification accuracy of 94%.
Keywords: *Deep Learning, Classification, Apple Tree Leaves Diseases*

I. Introduction

There is always a great challenge for agricultural products to feed the growing population of the world. The sustainability of farming products against diseases is an essential part of increasing the yield of the products. Plants and trees face biotic stresses and adversarial environmental and weather conditions [1]. To cope with these challenges, farmers use different techniques, including pesticides, fertilizers, irrigation policies, etc. Fruit agriculture provides food for human beings and the enjoyment of different tastes. Apple agriculture loses millions of dollars each year due to biotic and abiotic challenges. Throughout the growing season, various insects, fungi, bacteria, and viruses threaten apple orchards [2]. Apple leaves are exposed to many diseases such as Frogeye leaf spot, Fire blight, apple scab, Powdery mildew, and Alternaria leaf spot. The fungus Venturia inaequalis causes Apple scab, and it is one of the most dangerous fungal diseases that affect apples in temperate climates [3]. It attacks both leaves and fruits of an apple tree. The Dark tan with velvety spores and corky lesions characterizes scab disease on the infected fruit, as shown in Figure 1. Similarly, this fungal disease produces yellowish-green spots on the upper surface of the leaves and darker spots on the lower surface of the leaves. Basidiomycotina fungus causes cedar-apple rust. Small, pale yellow dots on leaves are the first signs of the illness, as shown in figure2. Early identification of plant disease is essential to start the treatment at the right time and increases production. Computer Vision (CV) and Machine Learning (ML) algorithms are used to detect disease in early stages, reducing the disease spread and increasing the cure rate. In some diseases, signs occur too late for successful treatment, while in others, they appear in the early phases of the disease [4].

Figure 1. Apple Scab

Figure 2. Cedar Apple Rust

Manuscript received July 5, 2021
Manuscript revised July 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.7.37

Powdery mildew, rust, and black rot may affect fruits that have to be inspected manually. Manual inspection is sluggish, vulnerable to mistakes, and requires a lot of human resources and time. The quality of the fruit influences customer choices. Most fruits are packaged by hand, which can degrade consistency and add to labor costs. Automated strategies must be simple, work well in low light, satisfy a range of client needs, and manage new fruit varieties. Low-quality fruit should be shipped to customers who want to juice or pickle it; the rotting pieces are extracted, and the rest is sliced or pressed. Fruits of superior quality demand a premium. A system that can differentiate between different types of fruit and their quality is necessary [5]. Deep learning [6] is a method of machine learning in which a machine model learns to perform classification tasks based on images, text, or sound. Deep learning models can achieve high classification accuracy, outperforming humans in many cases. Models are trained using multi-layer neural network architectures. Deep learning techniques have been successfully implemented in many real-life applications and have recently been reached in the agricultural domain. Pre-trained GoogleNet and AlexNet models are used to derive deep features for a plant disease classification technique. A convolutional neural network with data augmentation is used to detect and classify tomato diseases [7]. This paper is organized as follows: Section II presents an overview of literature work, Section III explains materials and methods used in the article. Section IV summarizes results and discussions, and finally, conclusions and future work are listed in section V.

II. RELATED WORK

Plant diseases may lead to a shortage of food production and is a significant threat to plant growth and crop yield. Many researchers have conducted studies for the prediction of plant disease. Due to human errors and time-consuming processes in manual inspection, classical image processing and machine learning techniques are used for automatic detection. In recent years, machine learning techniques have been widely used for the detection of apple diseases. An algorithm for classifying plant diseases using texture-based features is proposed in [8]. The images are converted into grayscale, and features are extracted after segmentation. Support Vector Machine (SVM) classifier achieved an accuracy of 94% on a smaller dataset of 500 images containing 30 different types of plants.

Similarly, in [9], the authors proposed an approach for apple leaf disease detection using image processing and pattern recognition techniques. They have used a dataset of plant leaves containing 90 images containing different classes and achieved a classification accuracy of 90%. A fast, effective, and accurate approach for detecting apple plant diseases is proposed in [10]. Their proposed methodology consists of four steps: computing the color transformation, segmentation of these images using a k-means clustering algorithm, computing texture features, and finally, a pre-trained neural network model is used for the classification.

Classification accuracy of 93% is achieved on a dataset of 150 images containing six classes. Local Binary Patterns (LBP) with a machine learning-based decision support system is used to detect grapes plant diseases (downy mildew and black rot) in [11]. They have used a high pass filter to extract features and then trained a decision support system using these features. Deep Learning approaches have also been proposed for the detection of plant fruit diseases. In [12], they offered a deep learning-based detection end-to-end algorithm that extracts and classifies six types of apple leaves diseases. Their proposed approach achieves an accuracy of 97.18%. Another image processing-based plant disease detection approach is presented in [13]. They used a publicly available dataset of 54,306 images containing 38 classes of 14 crop species with 26 diseases and trained a CNN-based model. Their proposed model achieved an overall classification accuracy of 85%. In [14], the authors proposed a framework for detecting four common diseases of apple leaves. They preprocessed the images and used the transfer learning method to train the model. It is noted that the neural network-based approaches achieve more accurate results than classical image processing techniques (overall classification accuracy of 97%). In another study [15], the authors proposed a mathematical model based on a deep learning approach to detect plant disease with improved accuracy, efficiency in training, and generality. First, a region proposal network (RPN) was used to detect leaves and localize them even in a complex environment. Then the segmented leaves images from the RPN model are used to fine-tune the transfer learning model. Their model's output was evaluated using plant diseases such as black rot, rusts, and plaque diseases. Classification accuracy of 83.57% accuracy is achieved on the test dataset. Petrellis et al. [16] developed a mobile application that can identify prevailing disease signatures such as leaf color, spots, the shape of spots, etc. The detection process was accomplished by drawing histograms of the number of pixels to the number of color levels for each color channel (RBG). The difference between the histogram patterns in the healthy leaves and leaves with diseases, particularly the start, peak, and endpoints, gave insights in building a model that could identify deficiencies in plant leaves. The model achieved more than 90% for grape diseases such as Downy Mildew, Pierce, Esca, and Powdery Mildew.

Shrestha et al. [17] developed a model that contained five convolutional layers, batch normalization layers, max-pooling layers, and dense layers as output. In total, there were over 58 million trainable parameters. Images of 15 different plant diseases were trained on the model, and classification accuracy on the dataset was 88.8%.

III. MATERIALS AND METHODS

The primary purpose of this research is to classify the diseases in the apple leaves. To achieve this, pre-trained CNN models will be fine-tuned to classify four diseases. Plant Pathology 2020 - FGVC7 data is used in this paper [2], [18]. Four pre-trained models will be explored, namely, VGG16, ResNetV2, Inception_v3, and MobileNetv2. The disparity in the number of samples within classes may reduce the classification performance. Data augmentation is used, including flips, scale, crop, and illumination, to generate additional samples to overcome the class imbalance. According to [19], the optimizer used to train a model has a significant impact. Hence, four different optimizers: Stochastic Gradient Descent (SDG), Adadelta, RMSProp, and Adam, are studied.

Figure 3 describes the overall flow diagram of the framework. Images from the dataset are preprocessed and used to retrain a pre-trained deep learning model. After parameter optimization, the model can be used to classify the four classes of apple leaves diseases.

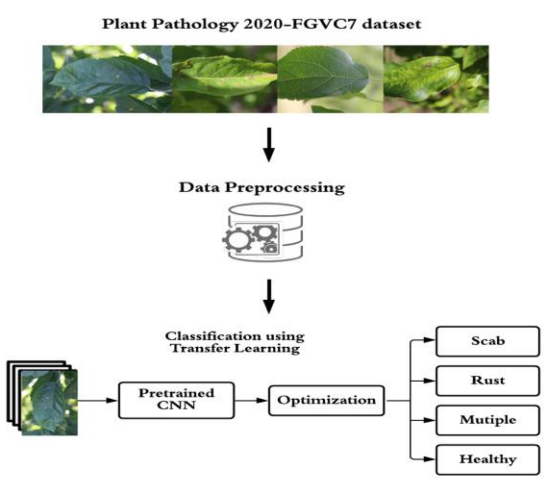

Figure 3. The proposed architecture of the methodology

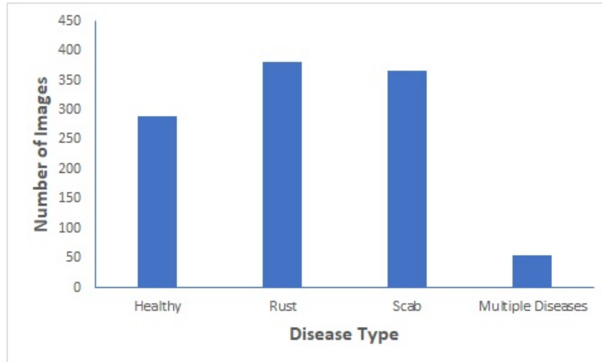

Figure 4. Classes distribution in Plant Pathology 2020 Dataset

A. Dataset

The dataset is gathered from Kaggle, collected by [2], and available on [18]. Plant Pathology 2020-FGVC7 is an online dataset consisting of 1821 images of apple scab, cedar apple rust symptoms, complex disease patterns (leaves with more than one disease in the same leaf), and healthy leaves. Classes distribution is shown in Figure 4. The healthy class contains 289 images, the rust class has 382 images, the scab class consists of 367 images, and the multiple diseases class includes 54 images.

B. Data Preprocessing

In the data preprocessing step, the whole dataset was split into three datasets at a ratio of 65:15:20 (65% as the training dataset, 15% validation dataset, and 20% testing dataset). The images are resized and normalized depending on the model (each model has its default image size, mean and standard deviation).

Table I
PROPERTIES OF THE PRE-TRAINED MODELS

Model	Parameters	Layers	Size (MB)
VGG-16	138 million	23	528
ResNetV2	44 million	-	171
Inception-V3	23 million	159	92
MobileNetV2	3.5 million	88	14

C. Transfer Learning

The technique of building a network from scratch is not commonly used. It is due to the significant amount of time and data needed for training the deep architecture. Transfer learning is a machine learning method in which a pre-trained model is supposed to be retrained partially on a small available dataset for a new application [20].

D. Pre-Trained Models

Four pre-trained models are considered in this paper, and a short description of each model is given below.

ResNetV2, a shorter abbreviation of Residual Networks, is already used in many computer vision tasks [21]–[23]. The main idea of the ResNet is to use identity connections that skip some layers. It contains 50 layers having 23 million parameters. After the initial release of ResNet [24], an improved model was built and called ResNetV2. The significant difference between ResNet and ResNet V2 is how the layers were arranged in the residual block. In ResNet V2, a batch normalization with a RELU activation layer was placed just before the 2D convolutional layer.

VGG16 [25] is a popular model used for various image classification tasks. It was trained on 14 million+ images that contained 1000 classes. VGG16 is generally seen as an improvement of the AlexNet model, which was considered the most popular image classifier. VGG16 is made of 5 blocks of convolutional layers and three fully connected layers.

Inception V3 [26] is another prominent CNN model that finds application in object detection and feature extraction. It was built from the popular GoogleNet, and the architecture of Inception V3 is made of 48 layers which can be loaded with its pre-trained weights for various image classification problems.

MobileNetV2 by Google is a deep convolutional neural network that can perform better on mobile devices [27]. It contains two types of blocks, residual blocks with the stride of 1 and residual blocks with the stride of 2 for downsizing. The architecture of MobileNetv2 comprises 2D convolutional layers and residual bottleneck layers. Residual blocks provide a skip connection from the start to the end of a convolutional block. Table I compares the number of layers and number of parameters for each model.

E. Optimization Algorithms

The deep CNN model is trained by iteratively updating the parameters of all layers in the network, with the optimizer method playing a pivotal role. A gradient descent algorithm is a popular option for neural network optimization. The parameters are updated in the opposite direction of the objective function gradient to minimize an objective function. The desired output and projected output are compared at each iteration, and the error is back-propagated. Cross entropy is one of the most often used output evaluation metrics. When the desired and estimated outputs are precisely the same, the cross-entropy value is close to zero, stopping criteria for the optimizer. In this paper, four optimization methods, namely, Stochastic Gradient Descent (SGD), RMSprop, Adadelta, and Adam, were explored for four pre-trained models.

The Stochastic Gradient Descent method typically converges smoothly, and it can also be used online. The objective function may fluctuate significantly because SGD may make bigger initial steps [28]. The SGD weight update rule is given in Equation 1.

$$\theta_{t+1} = \theta_t - \eta d_t \qquad (1)$$

where d_t denotes the objective function's gradient based on θ at time t, and is the learning rate [29].

AdaDelta optimizer is a stochastic optimization technique for SGD that enables different learning rates for individual dimensions. It monotonically decreases the learning rate value. AdaDelta reduces the window of combined past gradients to a fixed size w rather than counting all past squared gradients. The number of gradients is recursively represented as the decaying average of all previously squared gradients [28].

RMSprop optimizer works similarly to the gradient descent algorithm with the addition of a momentum term. Gradients are calculated differently between RMSprop and gradient descent. Beta is the measure of momentum, and it is usually set to 0.9 [30]. The weight update rule is given in Equation 2.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{R[d^2]_{t+\epsilon}}} d_t \qquad (2)$$

Adam optimizer is a combination of RMSprop and Stochastic Gradient Descent. Learning rates are modified as done in RMSprop optimizer and momentum term by taking the a moving average of the gradients similar to SGD with momentum [31].

F. Performance Measures

Performance of the Apple leaves classifier is evaluated based on overall classification accuracy, Precision, recall, and F-measure. Precision is defined as the number of truly positive instances (TP: True Positive) divided by the total number of positive instances (including True Positive and False Positive).

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (3)$$

Recall or Sensitivity is defined as the number of instances truly classified as positive (TP: True Positive) divided by the total Positive instances including TP and FN (False Negative).

$$\text{Recall} = \frac{TP}{TP+FN} \quad (4)$$

F-measure is the combination of Precision and recall and defined as,

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (5)$$

IV. RESULTS AND DISCUSSIONS

The dataset is divided into a training dataset (65%, 1164images), validation dataset (15%, 292 images), and testing dataset (20%, 365 images). A combination of four pre-trained models with four different optimizers is used in the initial experimentation. The dataset is divided into a training dataset (65%, 1164images), validation dataset (15%, 292 images), and testing dataset (20%, 365 images). A combination of four pre-trained models with four different optimizers is used in the initial experiments on the training dataset. The validation dataset is used to test these models in all the experiments. Batch size is fixed to eight, and Table II summarizes the results for sixteen experiments for different learning rates on the validation dataset. For the VGG16 model, the SGD optimizer performed better if the learning rate is 0.01. Overall, Adam optimizer produced the best classification accuracy of 0.901 for a learning rate equals to 0.006. In the ResNetV2 model, the Adam optimizer performed better than other optimizers for learning rates of 0.001 (classification accuracy is 0.9368). The performance of InceptionV3 and MobilenetV2 models is not good as compared to the ResNetV2 model. The best classification accuracy of InceptionV3 was 0.8983 for the RMSprop optimizer at the learning rate of 0.001, and it is 0.8598 for the MobilenetV2 model at the learning rate of 0.003 when the optimizer is RMSprop. Hence, we have found that a learning rate of 0.001 produced the highest classification accuracy. In the next experiment, we have fixed the learning rate equals to 0.001 and studied the effect of batch size on the classification accuracy.

Table III summarizes the classification accuracies for all the models when the learning rate is fixed, and batch size is varied. For the VGG16 model, a batch size of 32 (Adam optimizer) and batch sizes of 8 and 16 in the case of the RMSprop optimizer produced the same classification accuracy of 0.9065. For the ResnetV2 model, the classification accuracy of 0.9368 is achieved (batch size of 8 and Adam optimizer). For InceptionV3 and MobilenetV2 models, the batch size of 8 is the best with the RMSprop optimizer. Hence, overall the best classification accuracy was 0.9368 for the ResNetV2 model (batch size of 8 and Adam optimizer).

Based on tables II and III, the best combination of learning rate and batch size is 0.001 and 8, respectively, and the best model is ResNetV2. The pre-trained ResNetV2 model is trained and tested on the testing dataset with a learning rate equals to 0.001 and a batch size of 8. The values of the Precision for all the classes are high and more than 0.90. Recall of the multiple diseases class is very low as compared to the rest of the classes.

Table II
PERFORMANCE ANALYSIS FOR ALL MODELS AND OPTIMIZER METHODS WITH FIXED BATCH SIZE OF 8

Models	Optimizers	Learning Rate			
		0.01	0.001	0.003	0.006
VGG16					
	SGD	0.8873	0.8324	0.8241	0.8434
	Adam	0.8186	0.8681	0.8626	0.9010
	AdaDelta	0.6318	0.4560	0.5549	0.6126
	RMSprop	0.2994	0.9065	0.8131	0.7527
ResNetV2					
	SGD	0.3269	0.8901	0.8846	0.6675
	Adam	0.8956	0.9368	0.9230	0.8626
	AdaDelta	0.7280	0.2417	0.3406	0.5302
	RMSprop	0.2967	0.9093	0.7967	0.8049
Inception _v3					
	SGD	0.8379	0.7087	0.8076	0.8324
	Adam	0.8736	0.8626	0.8791	0.8681
	AdaDelta	0.4780	0.3489	0.3956	0.4368
	RMSprop	0.8461	0.8983	0.8928	0.8598
Mobilenetv2					
	SGD	0.8021	0.7060	0.7417	0.7857
	Adam	0.8296	0.7664	0.8214	0.8379
	AdaDelta	0.4532	0.3708	0.4093	0.4120
	RMSprop	0.8269	0.8049	0.8598	0.8296

Table III
PERFORMANCE ANALYSIS FOR ALL MODELS AND OPTIMIZER METHODS WITH FIXED LEARNING RATE OF 0.001

Models	Optimizers	Batch Size			
		8	16	32	64
VGG16					
	SGD	0.8324	0.7912	0.7472	0.6785
	Adam	0.8681	0.8791	0.9065	0.8626
	AdaDelta	0.4560	0.4423	0.4203	0.4093
	RMSprop	0.9065	0.9065	0.8983	0.8489
ResNetV2					
	SGD	0.8901	0.9038	0.8846	0.8956
	Adam	0.9368	0.9203	0.9258	0.9285
	AdaDelta	0.2417	0.2335	0.2445	0.2417
	RMSprop	0.9093	0.9010	0.9203	0.8269
Inception _v3					
	SGD	0.7087	0.7390	0.6401	0.4615
	Adam	0.8626	0.8489	0.8296	0.8214
	AdaDelta	0.3489	0.3379	0.3076	0.3021
	RMSprop	0.8983	0.8846	0.8736	0.8681
Mobilenetv2					
	SGD	0.7060	0.6346	0.5467	0.5
	Adam	0.7664	0.7554	0.7307	0.6813
	AdaDelta	0.3708	0.3489	0.3406	0.3324
	RMSprop	0.8049	0.8021	0.7829	0.7692

Table IV
PERFORMANCE ANALYSIS OF RESNETV2 MODEL

	Healthy	Multiple	Rust	Scab
Precision	0.964	1.0	0.928	0.9819
Recall	0.9819	0.2	0.9914	0.9914
F1-score	0.9732	0.333	0.9586	0.966

Table V
CONFUSION MATRIX OF RESNETV2 MODEL

	Healthy	Rust	Scab	Multiple diseases
Healthy	109	0	2	0
Rust	1	116	0	0
Scab	1	0	116	0
Multiple diseases	2	9	5	4

Table VI
COMPARISON WITH PUBLISHED RESULTS

Study	Model	Accuracy
[9]	ML Algorithms	0.90
[10]	Clustering + ANN	0.93
[8]	SVM	0.94
This Study	ResNetV2 + Adam	0.947
[15]	RPN + Transfer Learning	0.837
[17]	CNN	0.88

It shows that the multiple disease class instances are confused with other classes. The confusion matrix for all four classes is shown in Table V. It is evident from the table that most of the healthy, Rust, and Scab classes are classified correctly. In contrast, the majority of instances of multiple disease classes are confused with other classes. One reason for such a result is the lack of enough instances of the multiple disease class. The overall classification accuracy of the model is 94.7%.

Table VI compares the performance of our trained model with other published results. It can be seen that the classification accuracy of our trained model is better or comparable with the published results. Our transfer learning framework from a pre-trained model can perform with high classification accuracy with more instances and more classes.

V. CONCLUSIONS & FUTURE WORK

Agriculture is considered an engine of developing economics and industrialized countries. It is the source of food consumed by human beings and animals. Plant disease causes considerable losses to the agriculture sector, which results in less production. In this study, a convolution neural network was trained to detect diseases in plant leaves using transfer learning approaches. Early diagnosis and detection might control the spread of disease in the early stages to mitigate the losses resulting from the disease. The ResNet V2 model with a selected learning rate made predictions with an accuracy of 94.7%. The effect of different optimizers was also studied in this work, and the Adam optimizer is effective in the transfer learning of the ResNetV2 model. It is recommended that increasing the number of instances may further improve classification accuracy. In future work, we will study adding more classes and more images to improve the performance.

REFERENCES

[1] E. Gimenez, M. Salinas, and F. Manzano-Agugliaro, “Worldwide research on plant defense against biotic stresses as improvement for sustainable agriculture,” Sustainability, vol. 10, no. 2, p. 391, 2018.

[2] R. Thapa, N. Snavely, S. Belongie, and A. Khan, “The plant pathology 2020 challenge dataset to classify foliar disease of apples,” arXiv preprint arXiv:2004.11958, 2020.

[3] W. E. MacHardy, D. M. Gadoury, and C. Gessler, “Parasitic and biolog- ical fitness of venturia inaequalis: relationship to disease management strategies,” Plant disease, vol. 85, no. 10, pp. 1036–1051, 2001.

[4] S. S. Abu-Naser, K. Kashkash, and M. Fayyad, “Developing an expert system for plant disease diagnosis,” 2010.

[5] I. M. Nasir, A. Bibi, J. H. Shah, M. A. Khan, M. Sharif, K. Iqbal, Y. Nam, and S. Kadry, “Deep learning-based classification of fruit diseases: An application for precision agriculture,” CMC-COMPUTERS MATERIALS & CONTINUA, vol. 66, no. 2, pp. 1949–1962, 2021.

[6] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT press, 2016.

[7] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, “A robust deep-learning- based detector for real-time tomato plant diseases and pests recognition,” Sensors, vol. 17, no. 9, p. 2022, 2017.

[8] S. Arivazhagan, R. N. Shebiah, S. Ananthi, and S. V. Varthini, “Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features,” Agricultural Engineering International: CIGR Journal, vol. 15, no. 1, pp. 211–217, 2013.

[9] Z. Chuanlei, Z. Shanwen, Y. Jucheng, S. Yancui, and C. Jia, “Apple leaf disease identification using genetic algorithm and correlation based feature selection method,” International Journal of Agricultural and Biological Engineering, vol. 10, no. 2, pp. 74–83, 2017.

[10] D. Al Bashish, M. Braik, and S. Bani-Ahmad, “Detection and classification of leaf diseases using k-means-based segmentation and neural network based classification,” Information technology journal, vol. 10, no. 2, pp. 267–275, 2011.

[11] H. Waghmare, R. Kokare, and Y. Dandawate, “Detection and classification of diseases of grape plant using opposite colour local binary pattern feature and machine learning for automated decision support system,” in

2016 3rd international conference on signal processing and integrated networks (SPIN), pp. 513–518, IEEE, 2016.

[12] A. I. Khan, S. Quadri, and S. Banday, "Deep learning for apple diseases: classification and identification," International Journal of Computational Intelligence Studies, vol. 10, no. 1, pp. 1–12, 2021.

[13] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," Frontiers in plant science, vol. 7, p. 1419, 2016.

[14] B. Liu, Y. Zhang, D. He, and Y. Li, "Identification of apple leaf diseases based on deep convolutional neural networks," Symmetry, vol. 10, no. 1, p. 11, 2018.

[15] Y. Guo, J. Zhang, C. Yin, X. Hu, Y. Zou, Z. Xue, and W. Wang, "Plant disease identification based on deep learning algorithm in smart farming," Discrete Dynamics in Nature and Society, vol. 2020, 2020.

[16] N. Petrellis, "Mobile application for plant disease classification based on symptom signatures," in Proceedings of the 21st Pan-Hellenic Con- ference on Informatics, pp. 1–6, 2017.

[17] G. Shrestha, M. Das, N. Dey, et al., "Plant disease detection using cnn," in 2020 IEEE Applied Signal Processing Conference (ASPCON), pp. 109–113, IEEE, 2020.

[18] Kaggel, Plant Pathology 2020 - FGVC7, 2020. https://www.kaggle.com/c/plant-pathology-2020-fgvc7/data.

[19] Y. Wang, J. Liu, J. Mišić, V. B. Mišić, S. Lv, and X. Chang, "Assessing optimizer impact on dnn model sensitivity to adversarial examples," IEEE Access, vol. 7, pp. 152766–152776, 2019.

[20] Z. Guo, Q. Chen, G. Wu, Y. Xu, R. Shibasaki, and X. Shao, "Vil- lage building identification based on ensemble convolutional neural networks," Sensors, vol. 17, no. 11, p. 2487, 2017.

[21] T. Leelaruji and N. Teerakawanich, "Short term prediction of solar irradiance fluctuation using image processing with resnet," in 2020 8th International Electrical Engineering Congress (iEECON), pp. 1–4, IEEE, 2020.

[22] D. Sarwinda, R. H. Paradisa, A. Bustamam, and P. Anggia, "Deep learning in image classification using residual network (resnet) variants for detection of colorectal cancer," Procedia Computer Science, vol. 179, pp. 423–431, 2021.

[23] X. Liu, Y. Zhou, J. Zhao, R. Yao, B. Liu, D. Ma, and Y. Zheng, "Multiobjective resnet pruning by means of emoas for remote sensing scene classification," Neurocomputing, vol. 381, pp. 298–305, 2020.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778, 2016.

[26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818–2826, 2016.

[27] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510–4520, 2018.

[28] Sebastian Ruder, An overview of gradient descent optimization algorithms, 2016. https://ruder.io/optimizing-gradient-descent/.

[29] S. Bera and V. K. Shrivastava, "Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification," International Journal of Remote Sensing, vol. 41, no. 7, pp. 2664–2683, 2020.

[30] Towardsdatascience, A Look at Gradient Descent and RMSprop Optimiz- ers, 2018. https://towardsdatascience.com/a-look-at-gradient-descent- and-rmsprop-optimizers-f77d483ef08b.

[31] Towardsdatascience, Adam — latest trends in deep learning optimiza- tion., 2016. https://towardsdatascience.com/adam-latest-trends-in-deep- learning-optimization-6be9a291375c.