Deep-CNN for Plant Disease Diagnosis Using Low Resolution Leaf Images

Ashiqur Rahman, Md Hafiz Al Foisal, Md Hafijur Rahman, Md. Ranju Miah and M. F. Mridha1[0000-0001-5738-1631]

[1]Bangladesh University of Business & Technology, Dhaka, Bangladesh
ashikak270@gmail.com,alfoisal940700@gmail.com, developer.hafij@gmail.com, ranjumiah826@gmail.com, firoz@bubt.edu.bd

Abstract. Plant diseases present a threat to food security, but early recognition is problematic in many areas due to a lack of necessary facilities. Plant disease diagnosis is critical in agriculture since diseases frequently restrict plant production capacity. Manual strategies to identify plant diseases, on the other hand, are often temporal, challenging, and lengthy. As a consequence, agricultural automation with automated identification of plant diseases is widely preferred. In the modern time, for the advancement of computer vision, detecting diseases utilizing leaf representation of a specific plant has already been implemented. Despite these challenges, most of the implemented models could only identify diseases of a particular plant using high-resolution images, which is quite expensive from a farmer's position. Because of the variation in leaf colours, aspect ratios, and congested backgrounds, detecting plant disease by low-quality images is difficult. With the advancement of deep convolutional neural networks (DCNNs), the field of object recognition from low-resolution images has seen significant progress. This paper explores an efficient plant disease identification model that combines multiple plant diagnosis for low-resolution images. The model inherits a multilabel classification system to classify both the plant and the specific disease simultaneously. We gathered data for the study and analysis from online articles, including leaf images of tomatoes, corn, and apples. For our research, we have used various stand-ard convolutional neural network (CNN) architectures such as Xception, ResNet, DenseNet, and MobileNet to get better performance in this task. The result comparison looks like this: Xception, ResNet, DenseNet performs better only on High-resolution images, and MobileNet performs well in low-resolution images.

Keywords: Plant disease, Convolutional neural networks, Deep learning, Transfer learning.

1 Introduction

A disease occurring in food crops is a significant threat that has a significant economic effect, including its productivity. Plant diseases are responsible for 10–16 percentage of crop damage worldwide every year. Plant diseases have an impact on the growth of individual plants, and early detection is crucial. So if disease outbreaks are not identified in advance, food disruption will grow. Plant disease study is associated with the examination of visually noticeable patterns on plants and especially on leaves. It takes quite a lot of fieldwork, such as looking at any leaf or plant to find symptoms to detect diseases at an early stage. This method could be sped up using advanced computational methods to analyze real-time images of plants or leaves and recognize the disease. Numerous Machine Learning (ML) frameworks were used to detect and identify plant diseases, such as SVM classifiers, KNN classifiers, ANN classifiers, etc.

The advancement makes far towards deep learning approaches, which is also proven to be a novel approach for further research. Deep learning methods have been developed to detect any objects or similar patterns and have seen great success due to their reliability. Researchers tried many Deep Learning approaches to build a system that can detect plant diseases more frequently with real-time input. Convolutional neural networks (CNNs) were already introduced by DL researchers that overcame the image-related plant disease recognition challenges. The development of the CNN approach has become more reliable and shows improvement after getting in touch with the transfer learning strategy. In transfer learning, the CNN model is initially trained on a pre-trained dataset. The pre-trained model recognizes related image patterns out of the same or different dataset ranges. This strategy helps to prevent overfitting issues with the small datasets.

Several works have been done on high-resolution image datasets when analyzing some literature on detecting or diagnosing plant disease. Most farmers have no access to high-resolution cameras and cannot afford to detect diseases of plants. Consequently, a system for diagnosing and recognizing plant diseases on low-resolution images that would be both effective and highly accurate must be developed. And so, we are determined to develop a framework that can effectively detect the patterns of low-resolution images. We have chosen an already experimented dataset for this work, and we will work on three different plants as Apples, tomatoes, and corn. The overall contribution can be summed up as:

- We have investigated previously done researchers to determine the general knowledge behind the architectures and their inabilities in deep learning.
- We gathered images of 3 plants (Tomato, Apple, and Corn) 14 diseases to construct our study and evaluate the deep-CNN architecture.
- We tested six image recognition baseline strategies, including Densenet, Inception, Mobilenet, ResNet, VGG, and Xception, and explored MobileNet compelling low-resolution plant leaf images.

The continuation of this research is organized in the following manner: The previous literature is presented in Section 2. The dataset is introduced in Section 3. Section 4 describes the overall structure of deep CNN. Section 5 evaluates the model and measures the outcome. Finally, Section 6 brings the chapter to a conclusion.

2 Related Work

Numerous works on automated plant diagnostic techniques and detection using deep learning techniques have developed in recent years. Chen, J.[1] introduced transfer learning of the convolutional neural network as a method for identifying plant diseases using pre-trained models learned from large datasets and transferring them into the particular task in their very own dataset. This method advances in this area by not using random weights; alternatively, they utilized ImageNet, a pre-trained large labeled dataset. Shruthi, U.[2] showed a general phase of the plant disease detection process and a comparison study with some of the most commonly used machine learning classification methods. They compared SVM classifiers, ANN classifiers, KNN classifiers, Fuzzy classifiers, and Deep Learning approaches, and CNN outperformed them all. Ferentinos, K. P. [3] presented an established deep learning model for detecting diseased plants from an openly available dataset that a particular Convolutional Neural Network develops. The VGG Convolutional Neural Network was the most efficient.

Barbedo, J. G. A.[4] investigated different diseases which affect the same leaf but in different parts. To detect the disease, the researchers divided the leaf criteria into isolated symptoms taken individually and defects that were part of clusters taken as a group. This study demonstrated how dataset limitations could influence any techniques used to classify plant disease. And this solution leads them to get more reliable results. Kawasaki, Y.[5] presented an approach to achieve high classification efficiency using a Convolutional Neural Network and the K-fold Cross-Validation technique and obtained an average accuracy. Durmuş, H.[6] shared research on plant disease detection using deep learning, which intended to detect diseases in real-time data collected by the robot. They worked hard to investigate cases such as physical changes in leaves that were observed by RGB cameras. They tested their network on AlexNet and SqueezeNet. Francis, M.[7] Presented a Convolutional Neural Network model developed to detect and classify plant diseases using apple and tomato leaf images of normal and infected plants. They detected the overfitting problem, eliminated that dropout value, and evaluated their output on the GPU. They attempted to provide the researchers with creative insight to create an integrated plant disease identification method. This paper addressed the approaches used to identify plant diseases utilizing images of leaves and some segmentation and feature extraction algorithms used during plant disease detection, namely the boundary and spot detection algorithm, k-means clustering, and Otsu Threshold algo-rithm[8].

Ramesh, S.[9] proposed to identify anomalies that arise on plants in their natural environment. To remove occlusion, they shot photographs with a plain backdrop. In terms of precision, the method was applied to other machine learning models. The classifier was tested using 160 images of papaya leaves and the Random forest classifier. Mohanty, S. P.[10] proposed a deep convolutional neural network trained on a public dataset of 54306 images to recognize 14 crop species and 26 diseases. The proposed method achieved an efficiency of 99.35 percentage on the test set. Venkataramanan, A. [11] presented a Deep Learning technique for identifying plant diseases by analyzing the leaf of a particular crop. The classifier is trained in multiple stages to remove possibilities at each point. An object detector, YOLO v3, extracted the leaf and analyzed it through ResNet18 models. Many strategies for recognizing disease symptoms were examined, and how researchers were using the PlantVillage dataset on DL models to assess results. A systematic study is needed to explain the factors influencing plant disease detection, including dataset classes and scale, learning rate, visibility, etc. Also, DL techniques must be enhanced to identify pathogens during their entire life cycle[12]. Sharma, P.[13] proposed a work investigating a possible solution for automatic detection in plants by training convolutional neural network (CNN) models with segmented image data. They compared the F-CNN model and S-

CNN model and tested more than double the performance in this work. Sladojevic, S.[14] proposed a novel approach for plant disease recognition model based on leaf picture classification, distinguishing 13 different plant diseases among healthy leaves and distinguishing plant leaves from their surroundings. This study[15] employs a minor variation of the convolutional neural network model known as LeNet to recognize and identify diseases in tomato leaves to achieve comparatively better outcomes for detecting tomato plant diseases.

3 Dataset

Three openly prepared plant disease datasets are applied throughout the investigations. The images were gathered from Kaggle. It carries pictures of 3 separate plants where every plant has a different number of related diseases. Table 1 reveals how the database is categorized in terms of plant varieties and illnesses. Besides, the number of images employed in every class is presented in Table 1. Figure 1 illustrates the three affected apple leaves, namely, Apple scab, Black rot, and Cedar apple rust, with a healthful leaf image. Also, in Figure 1, dysfunctional images of corn are displayed: Cercospora leaf spot, Common rust, and Northern leaf blight. Nine disorders of tomatoes, including Early blight, Late blight, Leaf mold, Septoria leaf spot, Spider mites, Target spot, Tomato mosaic virus, and Yellow leaf curl virus, are exhibited in Figure 1.

Table 1. The table summarises the plants and varieties of disease that the accumulated dataset includes.

Plant Name	Condition	Samples
Tomato	Healthy	2175
	Early Blight	2175
	Late Blight	2175
	Leaf Mold	2175
	Septoria Leaf Spot	2175
	Spider Mites	2175
	Target Spot	2175
	Tomato Mosaic Virus	2175
	Yellow Leaf Curl Virus	2175
Corn	Healthy	1925
	Cercospora Leaf Spot	1925
	Common Rust	1925
	Northern Leaf Blight	1925
Apple	Healthy	2180
	Apple Scab	2180
	Black Rot	2180
	Cedar Apple Rust	2180

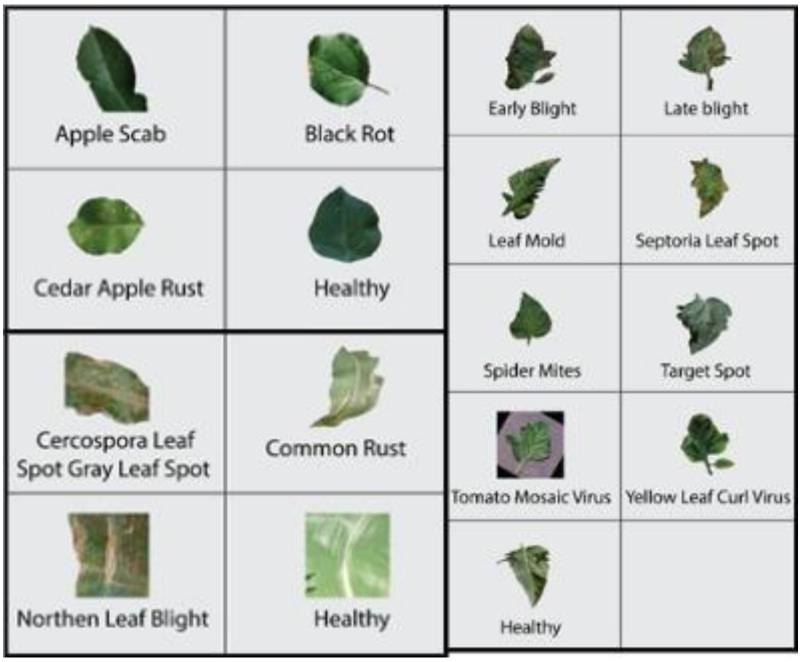

Fig. 1. The Figure shows the disorder images of apple (top left), corn(bottom left) and tomato (right).

4 Methodology

To diagnose plant diseases, various CNN architectures are introduced and compared. The basic concept and layers of different CNN architectures are instructed in the following sections.

4.1 Image Preprocessing

Normalization is a process in image analysis that alters the range of pixel intensities [16]. Data normalization assures that each input data has an equal distribution. Since the CNN architectures need the input images to be of the same form, each image data is restructured into 128 by 128 pixels, resulting in faster CNN convergence. As a consequence, each channel of the reconfigured leaf images is normalized as follows [17]:

$$Normalize(D) = \begin{bmatrix} d_{1,1} & \cdots & d_{1,m} \\ \vdots & \ddots & \vdots \\ d_{n,1} & \cdots & d_{n,m} \end{bmatrix} / 255 \tag{1}$$

Here D is the single-channel leaf image matrix, n refers to the number of rows, and m means the number of columns of the leaf image matrix.

4.2 Baseline Architecture

Convolutional Neural Networks are renowned for their ability to identify patterns in images. This field allows and enables machines to understand the world in the same way that individuals do and then apply that knowledge to several things and processes such as Image Recognition, Image Analysis, classification, etc. The first step is to feed the image's pixels in the form of arrays to the neural network's input layer (used to classify images). The hidden layers extract features by conducting various measurements. Convolutional neural networks are feedforward neural networks commonly used to interpret visual images by analyzing information and are referred to as a ConvNet [18].

To recognize and identify objects in an image, a convolutional neural network is used. A convolution neural network has many hidden layers that enable the extraction of information from images. The essential layers are given below with a brief discussion:

We emphasize analyzing reviewed CNN architecture for detecting plant disease on low-resolution images in this article. The CNN architecture is depicted in Figure 2, with the input layer (images of the leaves from the dataset), convolutional layers, a dense layer, and an output layer.

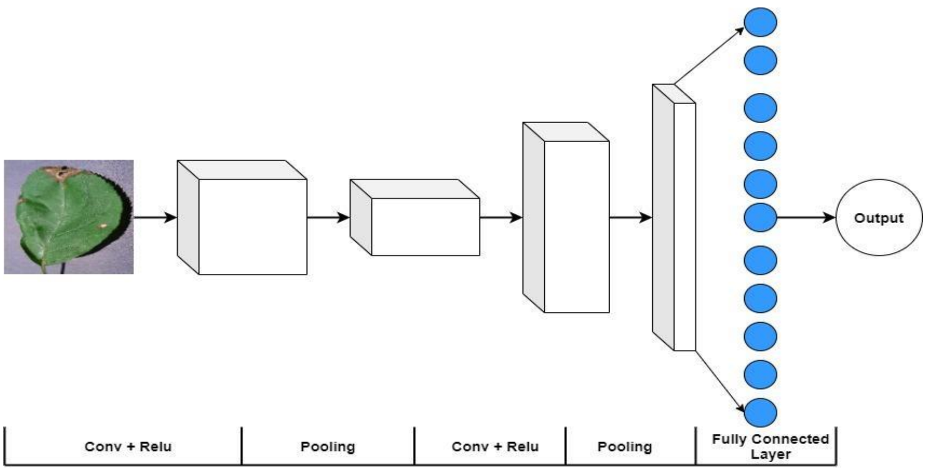

Fig. 2. The neural network architecture of plant diagnosis from leaf images.

Input layer: In the input layer, the original images of different plants from the dataset are input. But the authentic images are reconstructed with different widths, heights in the shape of 120 * 120 * 3 before given to the architecture as inputs.

Convolutional layers: This is the first step in extracting useful information from images. The convolution process is performed by plenty of filters (matrix of values known as weights that have been trained to detect unique features) in a convolution layer. CNNs utilize filters, also identified as kernels, to determine what features, such as edges, are present in images. The filter iteratively checks the images to see if the function it is designed to detect is available. The mathematical procedure can be found in[19]. The Batch Normalization process converts the images into a standard shape based on the mean and variance of a particular batch. It also improves the reliability of the network and leads to faster convergence.

ReLU Layer: The rectified linear unit (ReLU) is an abbreviation. After the function maps have been removed, they must be moved to a ReLU layer. This is a non-linear activation function, and its sole aim is to incorporate non-linearity into the network. The ReLU function is straightforward; values less than or equal to zero become zero, while all positive values remain unchanged.

Pooling layer: Pooling is a type of downsampling that reduces the dimensions of a feature map. Reduce the redundancy in the input function to boost up the training process and decrease the amount of storage used by the network. To generate a pooled feature map, the rectified feature map is now passed through a max-pooling pooling layer. Max pooling dramatically decreases the size of the image, lowering the amount of system memory and the number of operations conducted further in the network.

Fully Connected Layer: The following step in the process is known as flattening (convert all the resultant 2-Dimensional arrays into a single long continuous linear vector). With the fully connected operation of a neural network, the input image is flattened into a feature vector and transferred through a network that predicts the output possibilities. This fully connected layer is followed by several dense layers of neurons, which eventually produce accurate observations.

For this analysis, the MobileNet [19] architecture provides the highest performance. Apart from the first layer, the MobileNet architecture is based on depth wise separable convolutions. The first layer is a dense or full convolutional layer. After each sheet, batch normalization and ReLU non-linearity are applied. On the other hand, the output layer is a fully-connected layer with no non-linearity that flows into the softmax for classification. Stridden convolution is being used for both depthwise convolutions and the first fully convolutional layer in downsampling. When depthwise and pointwise convolution are considered as different layers, the overall number of layers for MobileNet is 28.

4.3 Transfer Learning

In general, transfer learning refers to a method in which a model trained solely on a single problem is used in any manner on a second related issue. It is a common method in computer vision since it helps us to create accurate models while saving time. Instead of beginning the learning process from scratch, transfer learning begins with patterns learned while solving a different set of problems.

The first step is to obtain the pre-trained model that will be used to solve the specified task. To go for one of the frameworks, such as ResNet or Xception, to initialize the base model. The base model would generally have much more units in the final output layer than desired and would need to remove the final output layer. Then, add a final output layer that is task-compatible. Freezing the layers from the pre-trained model is crucial because all previous learning would be lost if not. The following move introduces additional trainable layers, which will convert existing features into predictions on the new dataset. After training the new layers on the dataset, we will have a model that can make predictions on the dataset. And we can improve the performance by fine-tuning. Fine-tuning is accomplished by unfreezing the base model or a part of it and training the whole model on the data set at a shallow learning rate. The low learning rate would improve the model's efficiency on the new dataset while avoiding overfitting.

5 Evaluation

This section is designed with some subsections, including evaluation metrics, experimental setup, experiments, and comparison that we have done for this specific task. The datasets on which the tests were performed are given in Table 1.

5.1 Evaluation Metric

According to the following equations, we evaluated various baseline architectures by using the confusion matrix's accuracy, precision, and recall evaluation metrics.

Accuracy: By following the equation, we can find the accuracies of the model.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2}$$

Recall: The equation which helps to gain an efficient recall can be stated as

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

Precision: This equation will give the precision of the model, and it can be stated as

$$Precision = \frac{TP}{TP+FP} \tag{4}$$

5.2 Experimental Setup

We used Python for data collection, pre-processing, tests, and model evaluation [20]. And Keras [21] is used to implement the neural network architecture also for deep learning models. We used Numpy [22], which is a Python library for performing simple mathematical operations. Then TensorFlow [23] is used to produce neural network GPU execution.

5.3 Experiments and Comparisons

Experimented the proposed system using various pre-trained Keras-implemented transfer learning models like Mobilenet, MobileNetV2, Xception, ResNet50, ResNet50V2, ResNet101, DenseNet121, DenseNet169, DenseNet201, InceptionV3, VGG16, VGG19, and so on in this section. As the pretrained learned features were applied in the new diagnosis task, so it certainly identifies the diseases from the infected plants. Thus, we present the effectiveness of each model to determine which had the best accuracy, recall, and precision. Table 2 shows the dimensions of the developed system. We could see that MobileNet performs admirably, with 99.9 % accuracy, 99.08% precision, and 99.87% recall. This is the present state of the performance, as far as we can learn. Here, we applied different architectures such as MobileNet, ResNet, DenseNet, Inception, VGG, Xception, and their latest version, producing better performance. However, MobileNet provides the best performance for disease diagnosis in various plants. The concept of transfer learning was greatly used to run the experiment using these CNN architectures.

Table 2. This table represents the accuracy, precision, and recall of different architectures.

Model	Accuracy	Precision	Recall
Xception [27]	98.7	98.6	98.56
ResNet50 [28]	96.8	93.41	94.13
ResNet50V2 [29]	96.90	97.42	96.67
ResNet101 [25]	97.28	98.82	98.65
DenseNet121 [24]	97.87	94.1	95.54
DenseNet169 [24]	97.36	97.86	96.63
DenseNet201 [24]	97.08	97.88	96.67
InceptionV3 [25]	95.55	95.9	94.53
VGG16 [26]	94.56	92.53	93.45
VGG19 [26]	91.05	93.54	92.13
MobileNet [30]	**99.79**	**99.08**	**98.87**

6 Conclusion

This section incorporates and justifies a multi-plant diagnosis approach based on various image classifier baselines for low-resolution images. We used a transfer learning strategy to train and evaluate our method systematically. We collected a publicly introduced dataset that consisted of different images, and we evaluated three different plants, apples, corns and tomatoes, with several different diseased leaf images. We find that separate and distinct convolution and skipped connections greatly enhance plant disease detection efficiency. The paper focused on image processing using CNN for its excellent feature extraction capability and superiority over conventional systems in terms of real-time, precision, and resilience. We evaluate and observe different architectures such as Xception, ResNet, DenseNet and MobileNet, and in comparison, MobileNet performs better for low-resolution plant leaf images. This work does have a lot of potential for expansion. Changing the architecture of CNN will help to reduce feature extraction losses. We are confident that perhaps the objectives of this research work will lead the road for intensive studies on image processing methods in the future and improve the knowledge and usefulness of object classification based on CNN.

References

1. Chen, J., Chen, J., Zhang, D., Sun, Y., & Nanehkaran, Y. A. (2020). Using deep transfer learning for image-based plant disease identification. Computers and Electronics in Agriculture, 173, 105393.
2. Shruthi, U., Nagaveni, V., & Raghavendra, B. K. (2019, March). A review on machine learning classification techniques for plant disease detection. In 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS) (pp. 281-284). IEEE.
3. Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture, 145, 311-318.
4. Barbedo, J. G. A. (2019). Plant disease identification from individual lesions and spots using deep learning. Biosystems Engineering, 180, 96-107.
5. Kawasaki, Y., Uga, H., Kagiwada, S., & Iyatomi, H. (2015, December). Basic study of automated diagnosis of viral plant diseases using convolutional neural networks. In International Symposium on Visual Computing (pp. 638-645). Springer, Cham.
6. Durmuş, H., Güneş, E. O., & Kırcı, M. (2017, August). Disease detection on the leaves of the tomato plants by using deep learning. In 2017 6th International Conference on Agro-Geoinformatics (pp. 1-5). IEEE.
7. Francis, M., & Deisy, C. (2019, March). Disease detection and classification in agricultural plants using convolutional neural networks—a visual understanding. In 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN) (pp. 1063-1068). IEEE.
8. Khirade, S. D., & Patil, A. B. (2015, February). Plant disease detection using image processing. In 2015 International conference on computing communication control and automation (pp. 768-771). IEEE.
9. Ramesh, S., Hebbar, R., Niveditha, M., Pooja, R., Shashank, N., & Vinod, P. V. (2018, April). Plant disease detection using machine learning. In 2018 International conference on design innovations for 3Cs compute communicate control (ICDI3C) (pp. 41-45). IEEE.
10. Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. Frontiers in plant science, 7, 1419.
11. Venkataramanan, A., Honakeri, D. K. P., & Agarwal, P. (2019). Plant disease detection and classification using deep neural networks. International Journal on Computer Science and Engineering (IJCSE), 11(08), 40-46.
12. Saleem, M. H., Potgieter, J., & Arif, K. M. (2019). Plant disease detection and classification by deep learning. Plants, 8(11), 468.
13. Sharma, P., Berwal, Y. P. S., & Ghai, W. (2020). Performance analysis of deep learning CNN models for disease detection in plants using image segmentation. Information Processing in Agriculture, 7(4), 566-574.
14. Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. Computational intelligence and neuroscience, 2016.
15. Tm, P., Pranathi, A., SaiAshritha, K., Chittaragi, N. B., & Koolagudi, S. G. (2018, August). Tomato leaf disease detection using convolutional neural networks. In 2018 Eleventh International Conference on Contemporary Computing (IC3) (pp. 1-5). IEEE.
16. Kabir, M. M., Ohi, A. Q., & Mridha, M. F. (2020). A Multi-Plant Disease Diagnosis Method using Convolutional Neural Network. arXiv preprint arXiv:2011.05151.
17. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1097-1105.
18. Kabir, M. M., Ohi, A. Q., Rahman, M. S., & Mridha, M. F. (2020, December). An Evolution of CNN Object Classifiers on Low-Resolution Images. In 2020 IEEE 17th International

Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET) (pp. 209-213). IEEE.
19. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
20. Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual CreateSpace. Scotts Valley, CA.
21. Chollet, F. (2015). Keras: Deep learning library for theano and tensorflow. URL: https://keras. io/k, 7(8), T1.
22. Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: a structure for efficient numerical computation. Computing in science & engineering, 13(2), 22-30.
23. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. In 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16) (pp. 265-283).
24. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
25. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).
26. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
27. Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).
28. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
29. He, K., Zhang, X., Ren, S., & Sun, J. (2016, October). Identity mappings in deep residual networks. In European conference on computer vision (pp. 630-645). Springer, Cham.
30. Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017, February). Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 31, No. 1).