

A Comparative Study on Plant Disease Detection and Classification Using Deep Learning Approaches

Banothu Balaji

Department of Computer Applications, National Institute of Technology, Tiruchirappalli, India

Email: balaji@nitt.edu

ORCID iD: <https://0000-0002-5174-3010>

T. Satyanarayana Murthy*

Department of Information Technology, Chaitanya Bharathi Institute of Technology, Hyderabad, India

Email: tsmurthy_it@cbit.ac.in

ORCID iD: <https://0000-0002-6320-8496>

*Corresponding Author

Ramu Kuchipudi

Department of Information Technology, Chaitanya Bharathi Institute of Technology, Hyderabad, India

Email: kramupro@gmail.com

ORCID iD: <https://0000-0002-1126-8550>

Received: 24 June, 2022; Revised: 24 August, 2022; Accepted: 27 October, 2022; Published: 08 June, 2023

Abstract: Agriculture is a big sector in nations like India, and it provides a living for many people. To improve crop productivity, it's very necessary to identify and classify plant diseases and prevent them from spreading further so that they do not affect the whole plant. Artificial intelligence (AI) and computer vision can help detect plant diseases that humans cannot always catch and overcome the shortcomings of continuous human monitoring. In this article, we aim to detect and classify diseases in tomato and apple leaves using deep learning approaches and compare the results between different models. Because tomatoes and apples are important components of the human diet, crop waste can result in losses for both farmers and ordinary people. These plant diseases have an immediate and negative impact on both the amount and quality of yield. Crop diseases must be identified and prevented as soon as possible to improve crop yield. Therefore, we need to monitor and analyze the growth stages of the plants so that the farmers can produce disease-free and with minimal losses to the crop. Furthermore, we used the sequential convolutional neural network (CNN) model followed by transfer learning models like VGG19, Resnet152V2, Inception V3, and MobileNet and compared the models based on accuracy. The performance of the models was evaluated using various factors such as dropout, batch size, and the number of epochs. For both, the datasets, the tomato, and apple MobileNet architecture performed better than the other existing models.

Index Terms: Convolution Neural Networks, Transfer Learning, classification, disease

1. Introduction

The automated detection of plant diseases from plant leaves is a significant achievement in the agricultural sector. Plant diseases are a global hazard to food security, but they may be particularly dangerous for small-scale farmers who rely on healthy crops to make a living. Climate change, pollinator reductions, plant diseases, and other factors that continue to affect food security have a negative impact on plant leaves. However, agriculturists and pathologists may be unable to recognize symptoms in plants by looking at diseased leaves due to the extensive production of numerous agricultural crops. Farmers in rural areas may be required to travel a great distance to see an expert, which consumes time and money, these are the drawbacks of existing traditional approaches. Additionally, professional supervision is required. The need to study width and depth is illustrated by the fact that recent technological developments have allowed Deep Learning Networks and their related applications to evolve and effectively address challenging tasks

like object recognition and image detection. The networks perform better when they take a wider and deeper approach. Smartphones in particular provide novel methods to assist in disease identification because of their powerful computers, sharp displays, and extensive built-in accessory sets, such as advanced high-definition (HD) cameras. Disease detection has become more accurate because of deep learning network advancements and rising smartphone usage based on automated image recognition is now technically possible for farmers. This study aims to characterize and identify illnesses that affect tomato and apple plant growth. One of the most widely grown crops in the world, tomato production has been steadily rising over time. Apples are a pricey fruit, so early disease diagnosis is critical to increasing productivity and avoiding significant losses. So, detecting plant diseases early and accurately enhances agricultural productivity and quality. As a result, the high throughput and accuracy of automated computing systems for the identification and diagnosis of plant diseases benefit farmers and agronomists.

This article addresses the study of plant disease detection and classification based on different neural networks and a comparison of different neural networks used for tomato and apple plant leaf diseases. The different models of deep learning used are CNN and its architectures, VGG19, and transfer learning approaches like InceptionV3, MobileNet, and ResNet152V2.

2. Literature Review

In-plant leaf disease diagnosis and image recognition deep learning algorithms are often employed to reliably identify diseased leaves, which is an exciting field of research. Plant diseases can cause significant crop losses at any moment. So, to address this problem there are so many methods are proposed. Still, some methods are not giving better classification accuracy results.

Hassan et al. [1] suggested CNN models for identifying and detecting plant disease from the leaves. They made use of information from 14 different plant species, 38 different categories of diseases, and unharmed plant leaves. They used CNN models like InceptionV3 [6], InceptionResNetV2 [7], MobileNetV2, and EfficientNetB0 for classification. They evaluated color, grayscale, and segmentation photos after dividing the dataset into train and test data in various ratios. The performance of the models was evaluated using several parameters like batch size, dropout, and epoch count. Using InceptionV3, InceptionResNetV2, MobileNetV2, and EfficientNetB0, they were able to attain disease classification accuracy rates of 98.42 percent, 99.11 percent, 97.02 percent, and 99.56 percent, respectively. The peanut dataset was utilized by Qi et al [2] for disease classification and detection. The author used machine learning models like Support Vector Machine, Random Forest, and Logistic Regression in addition to deep learning models like VGG16, DenseNet121, AlexNet, ResNet50, and Inception V3 for identification. He also tested deep-learning models with and without data augmentation and found that when employing data augmentation with a stacking ensemble, the deep-learning models performed better. The accuracy of a residual network with 50 layers (ResNet50) after ensemble by logistic regression was 97.59 percent, whereas the F1 score of a dense convolutional network with 121 layers (DenseNet121) was 90.50 percent. Following the logistic regression ensemble, the deep learning model utilized in this experiment's F1 score improved the most. ResNet50 and DenseNet121, two deep learning networks with more network layers, outperformed other deep learning networks in this experiment. Trivedi et al. [3] used a Convolutional Neural Network to identify and classify tomato illnesses (CNN). Preprocessing the incoming images was their first step, followed by the original image segmentation of the desired region. Second, the images are subjected to several CNN hyper-parameters. CNN also pulls borders, color, and texture from photographs. The outcomes show that 98.49% of the predictions made by the suggested model are correct. G. Geetha [4] used four phases to diagnose illness in tomatoes. The four steps include pre-processing, leaf segmentation, feature extraction, and classification. They started with pre-processing to lower the noise, and then they employed image segmentation to separate the leaf's impacted or damaged parts. They used the k-nearest neighbors (KNN) method, a complex machine-learning algorithm that is directed, supervised, and used to solve both classification and regression issues.

3. Methodologies

In computer vision and image categorization right now, deep learning is a popular issue. To improve classification and prediction accuracy, it builds a multilayer neural network, feeds it a large amount of label data, and then extracts the feature representation from the data.

3.1 Convolutional Neural Network

Deep neural networks can be used to carry out tasks like image identification, classification, and object detection. One such type of deep neural network is the convolutional neural network (CNN or ConvNet) (Parekh) [5]. "Image classification is the process of identifying the class or group of classes that best describes an input image. In CNN, we use a picture as an input, prioritize the image's many elements, and be able to tell one element from another. CNN requires significantly less pre-processing than other classification techniques. As shown in Fig.1, a CNN typically consists of three layers: a convolutional layer, a pooling layer, and a fully connected layer.

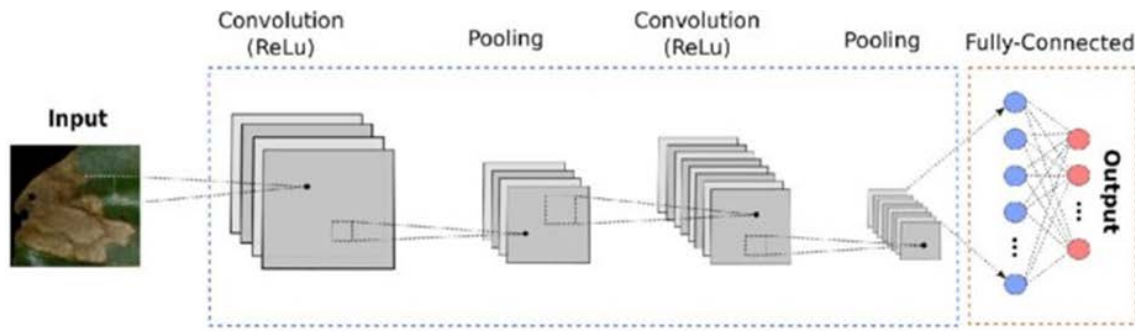


Fig. 1. CNN Architecture

Convolution: The main goal of convolution is to extract features from the input, such as edges, colors, and corners. The network begins to recognize more complex elements like shapes, digits, and facial regions as we learn more about them. This layer performs a dot product between two matrices: the limited section of the image and the set of learnable parameters (sometimes referred to as the filter or kernel).

Pooling: The main objective of this layer is to reduce the amount of computing power needed to process the data. By requiring fewer computers processing power to process the data, this is accomplished. It is accomplished by further reducing the highlighted matrix's dimensions. In this layer, we want to extract the most important features from a limited area of the neighborhood.

Fully connected layer: We will flatten our input image into a single column vector after converting it to a format appropriate for our multi-level fully linked architecture. Each training cycle transmits the flattened output to a feed-forward neural network using backpropagation. The model can classify and distinguish between dominant and specific low-level features in images throughout several epochs.

Table 1. Hyperparameters of the CNN model

Parameter	Description
No. of Convolution Layer	4
No. of Max Polling Layer	4
Dropout Rate	0.4, 0.3
Activation Function	Relu
Learning Rates	0.001
Epoch	30
Batch Size	32

Transfer learning in deep learning is the usage of a network that has already been trained on a new task. Because it enables the network to be trained with little input while maintaining high accuracy, transfer learning is a common deep learning technique. A machine generalizes to a new task using information from a previous one. In transfer learning, the learned network's final few layers are swapped out for new ones, including a fully connected layer and a SoftMax classification layer, which frequently has many classes in our article. Different dropout rates, learning rates, and batch sizes were used to test each model. The input size used is $224 * 224$.

3.2 VGG19

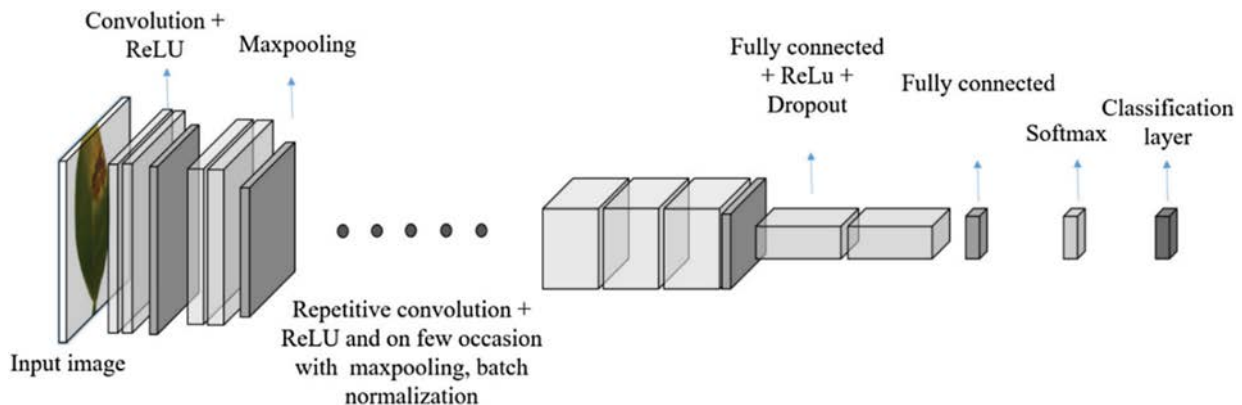


Fig. 2. VGG19 Architecture

The VGG convolutional neural network model was introduced in 2014 by Oxford University. When it was initially published, this idea quickly became well-liked because of how straightforward and practical it was. VGG19 [8,14] is a sophisticated CNN with layers that have already undergone training, and it has a thorough understanding of how to characterize an image's form, color, and structure. VGG19 has been trained on millions of images with challenging classification tasks and consists of 19 layers, including 16 convolutional layers, 3 fully connected layers, 5 MaxPool layers, and 1 SoftMax layer as shown in Fig.2. Instead of continuing to train VGG19 for my project, I froze its layers and added a thin 2-layer network on top to finish my categorization.

3.3 Inception-V3

The GoogLeNet framework is credited with first introducing the concept of "Inception." InceptionV3 improves model performance by creating a network that balances depth and breadth, as opposed to stacking network depth to do so. A convolutional neural network design from the Inception family called Inception-v3 carries label information down the network using Label Smoothing, Factorized 7×7 convolutions, and an additional classifier. The network's first two layers are a common convolutional layer and a max-pooling layer, which are followed by a stack of Inception modules with different topologies. Sizes of the convolutions range from 1×1 , 3×3 , and 5×5 . A 3×3 pooling layer is also present. The model benefits from using Inception modules by conserving a significant number of parameters, decreasing overfitting, and enhancing feature diversity. The network training process is easier as shown in Fig.3.

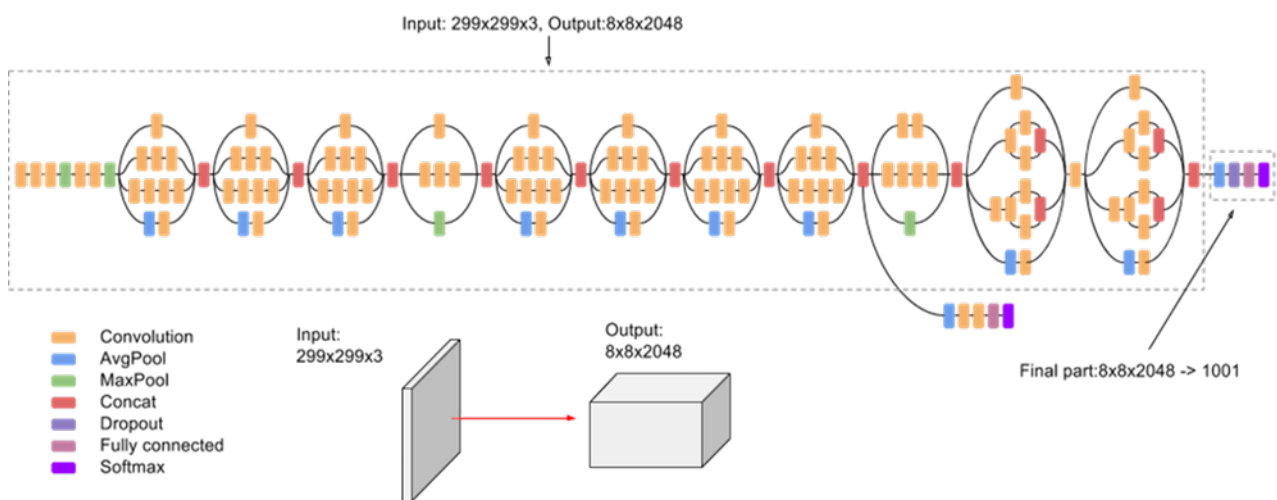


Fig. 3. Inception V3 architecture

3.4 Mobile NET

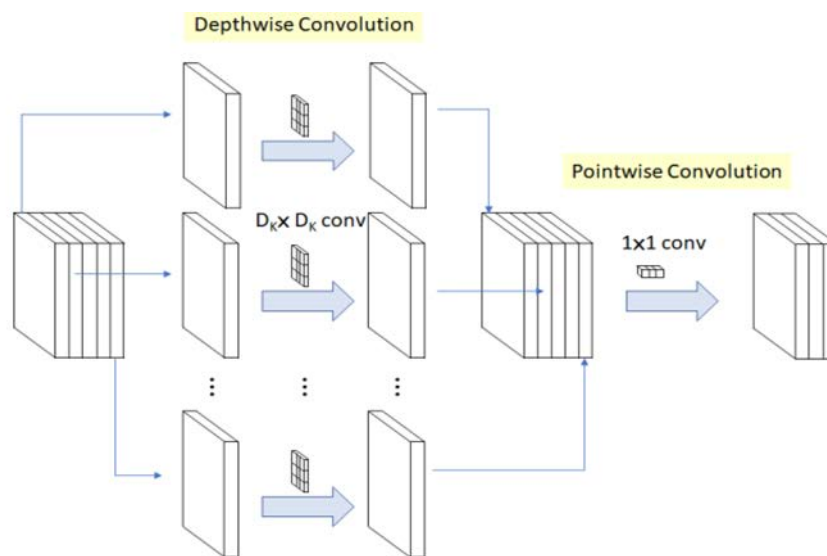


Fig. 4. Working of MobileNet

A very portable neural network is the Mobile Net [9-13,15]. This model has already undergone training. The depth-wise convolution method for Mobile Nets adds a single filter to each input channel as shown in Fig.4. The outputs of the depth wise convolution are then combined using the pointwise convolution as an 11 convolution. A conventional convolution filter produces a new set of outputs by integrating inputs in one step. The depth-wise separable convolution separates this into two layers, one for filtering and the other for combining. This factorization lowers the parameter while also reducing computation and model size.

3.5 ResNet152V2

He Kaiming proposed the residual network, a deep convolutional network. The precision of optimization improves as the network depth grows. Its main goal is to prevent gradient degradation brought on by increased network depth. Network performance can be enhanced by merely increasing network depth. By stacking the leftover blocks, the residual network is created. Identity blocks and convolution blocks are the two modules found in the remaining blocks. The identity block has the same input and output dimensions. Numerous identifying blocks can thus be joined as a result. The convolution block's function is to alter the feature vector's dimension because its input and output dimensions are dissimilar. Using a 33-size convolution kernel, the residual network converts the input image into a small but deep-feature map. The dimension of the output features increases as the network's depth increases, and the network learns more complex information.

Table 2. Comparison among different transfer learning models regarding depth and parameter size.

Model	Depth	Parameters (in million)	Size
VGG19	19	143.7	549 MB
Inception V3	189	23.9	92 MB
MobileNet	55	4.3	16 MB
ResNet152V2	307	60.4	232 MB

4. Datasets

We used the Tomato Leaf and Apple Leaf datasets as shown in Fig.5 and in Fig.6 respectively from the Kaggle website for this investigation. The validation dataset has 4304 healthy and unhealthy photos split into 10 types of tomato leaves, whereas the training dataset contains 18345 healthy and diseased leaf images. The apple dataset is organized into four categories and comprises 2716 training data and 484 validation data.

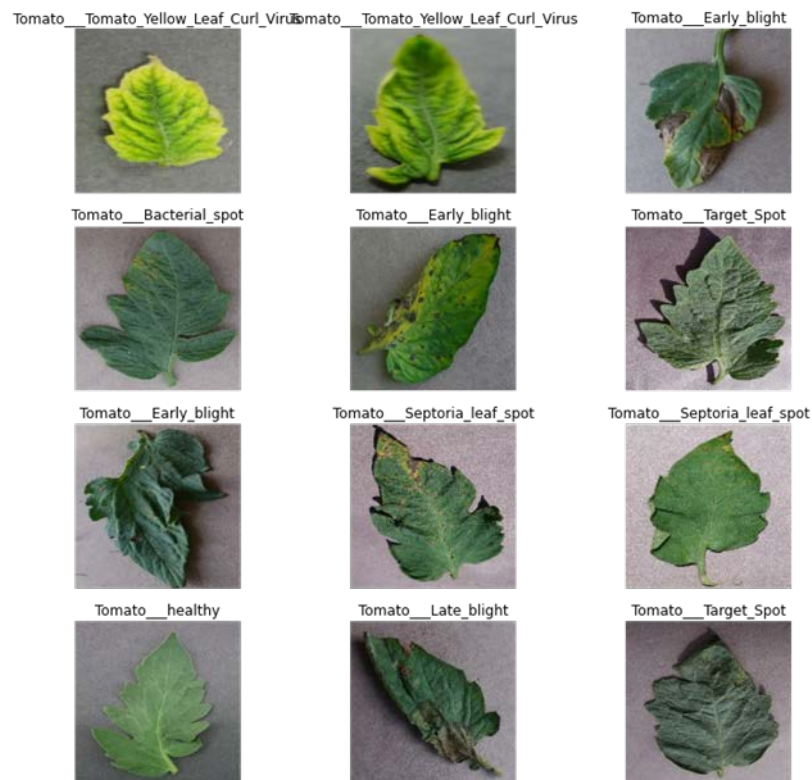


Fig. 5. Visualization of Tomato dataset

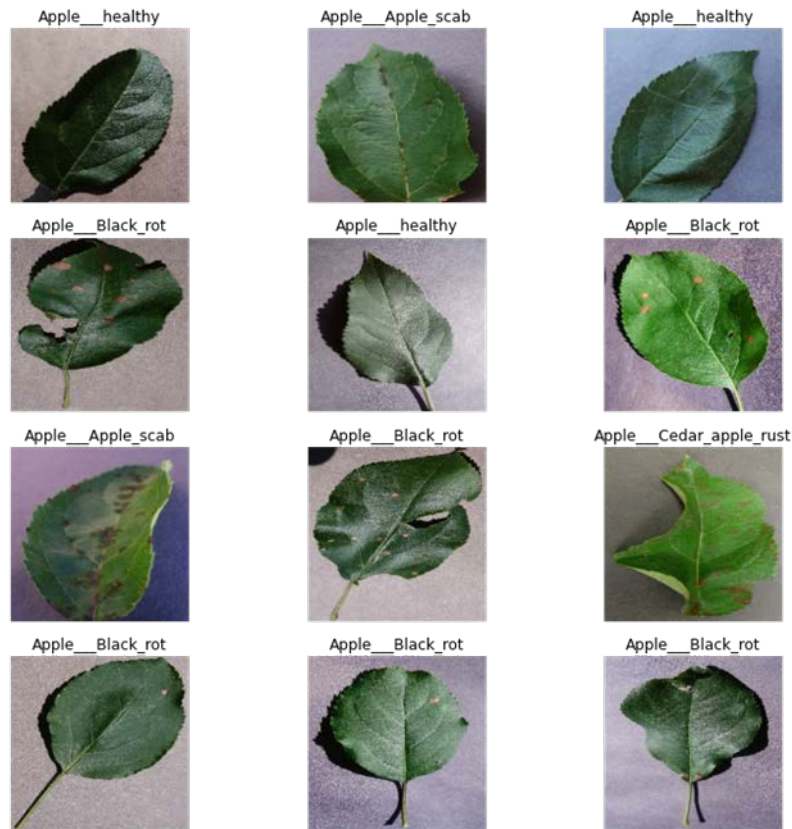


Fig. 6. Visualization of Apple dataset

5. Results and Discussions

The existing models and their architectures that were compared in this section are intended to aid in the identification and classification of tomato and apple leaf diseases in the agricultural sector. As a result, transfer learning approaches and CNN architectures are used to demonstrate the models' efficiency and sustainability. The preprocessed images were converted into NumPy arrays, and the transfer learning technique utilized in this study was used to retrain the pre-trained model with project-specific data, including enhanced images on the training set. Then, this was used to classify previously unknown data. A strong model that makes the best predictions was developed after testing different epochs, batch sizes, and learning rates. We created a four-layer CNN model from scratch and trained it for 30 epochs. The training accuracy for the tomato and apple plant leaves was 96% and 98.36%, respectively, suggesting that CNN models could benefit. The CNN model was quite accurate, although it required a long time to train the model compared to the transfer learning approaches.

Table 3. Accuracy comparison of all the models for the tomato dataset.

Models	Tomato Dataset				
	Training Accuracy	Validation Accuracy	Recall	Precision	F1-score
CNN	96%	90%	80%	81%	80%
VGG19	95%	90%	75%	78%	76%
Inception V3	92%	87%	81%	84%	82%
MobileNet	97%	93%	91%	92%	91%
ResNet152V2	95%	93%	90%	91%	90%

Table 4. Accuracy comparison of all the models for the apple dataset.

Models	Apple Dataset				
	Training Accuracy	Validation Accuracy	Recall	Precision	F1-score
CNN	98.36%	98.55%	92%	93%	92%
VGG19	99.78%	99.45%	99%	99%	99%
Inception V3	98.75%	97.52%	94%	94%	94%
MobileNet	99.45%	98.97%	100%	100%	100%
ResNet152V2	98.49%	98.14%	97%	97%	97%

As we can observe in Tables 3. and Table 4. MobileNet has given the highest accuracy of 97% and 99.45% for the tomato and apple leaf datasets, respectively. The model accuracy is enough to give an idea of the model but calculating the precision, recall, and F1 score provides a much better understanding of the models.

Fig. 6 (a),(b),(c), and (d) show the graphs of accuracy versus epochs, loss versus epochs, validation accuracy versus epochs, and validation loss versus epochs respectively of all the implemented models for the tomato dataset. In the accuracy versus epochs graph, the accuracy for all the implemented models increased with an increase in epochs. The accuracy of MobileNet is the highest which is 97% followed by the CNN, VGG19, ResNet152V2, and InceptionV3 models.

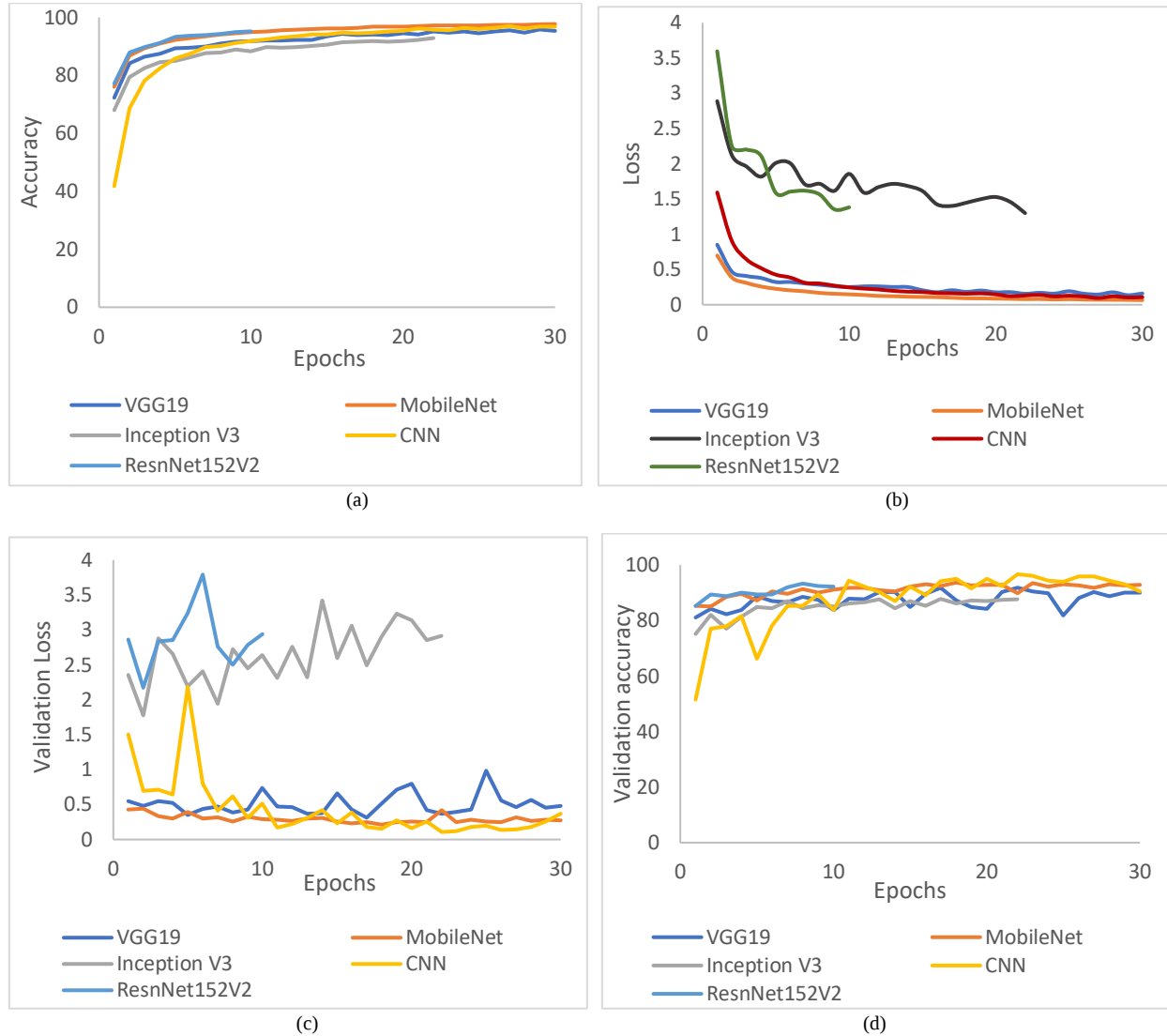


Fig. 6. (a) Accuracy vs epochs (b) Loss vs epochs (c) Validation Accuracy vs epochs (d) Validation Loss vs epochs for the tomato dataset

Fig. 7 (a), (b), (c), and (d) show the graphs of loss versus epochs, accuracy versus epochs, validation loss versus epochs, and validation accuracy versus epochs respectively, of all the different transfer learning models for the apple dataset. In the accuracy versus epochs graph, the accuracy for all the implemented models increased with an increase in epochs. The accuracy of MobileNet is the highest because it uses depthwise separable convolutions, which is 99% followed by the CNN, VGG19, ResNet152V2, and InceptionV3 models. Therefore, MobileNet gives the best accuracy as shown in Fig. 8, for both the tomato and apple datasets, because MobileNet is a lightweight model, it takes the least time to train the model compared to other transfer learning models.

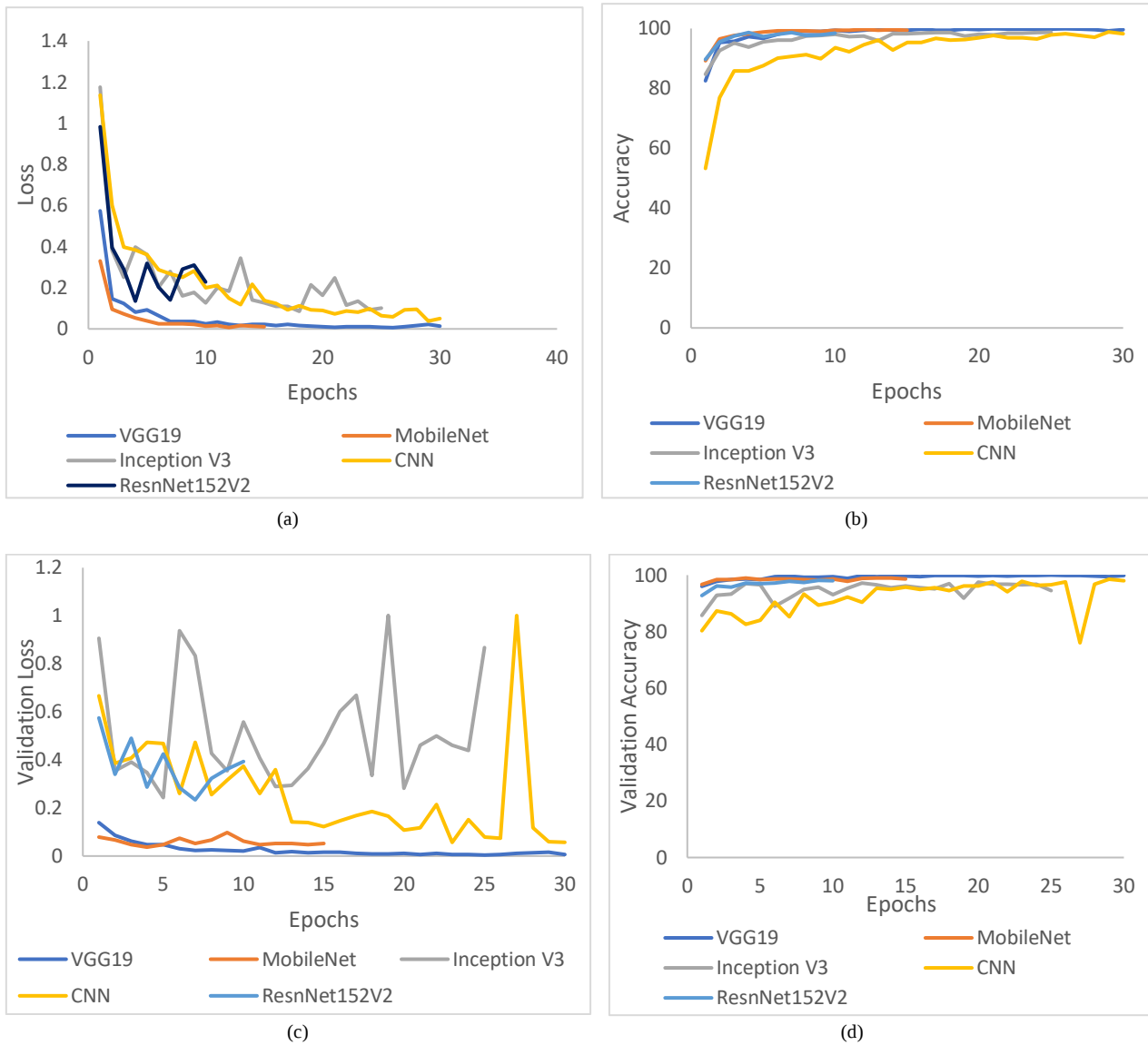
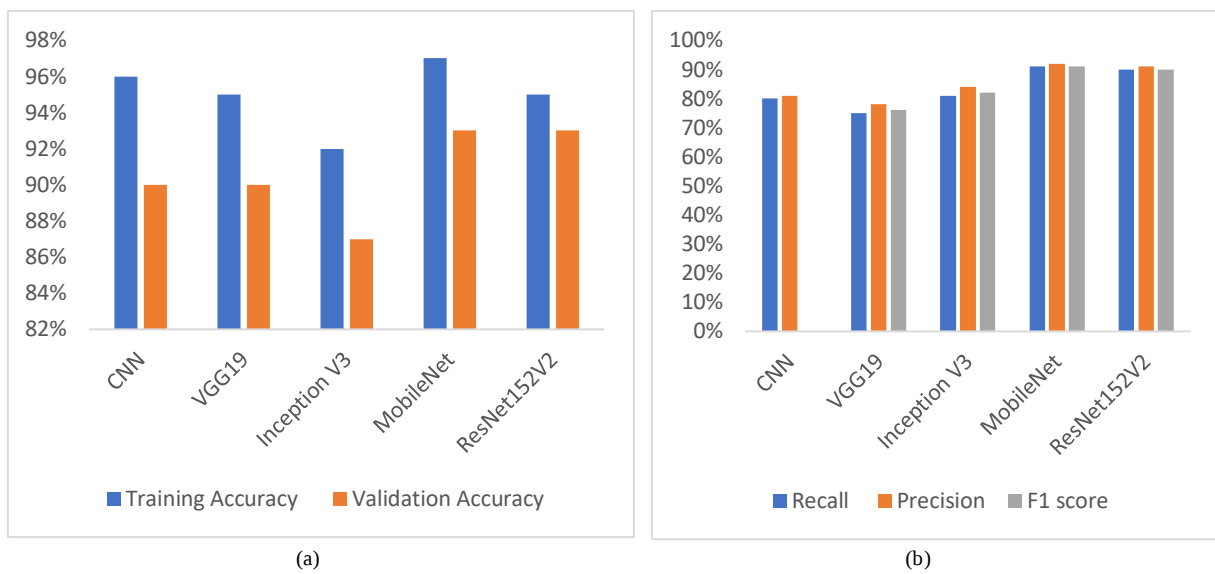


Fig. 7. (a) Accuracy vs epochs (b) Loss vs epochs(c) Validation Accuracy vs epochs (d) Validation Loss vs epochs for apple dataset



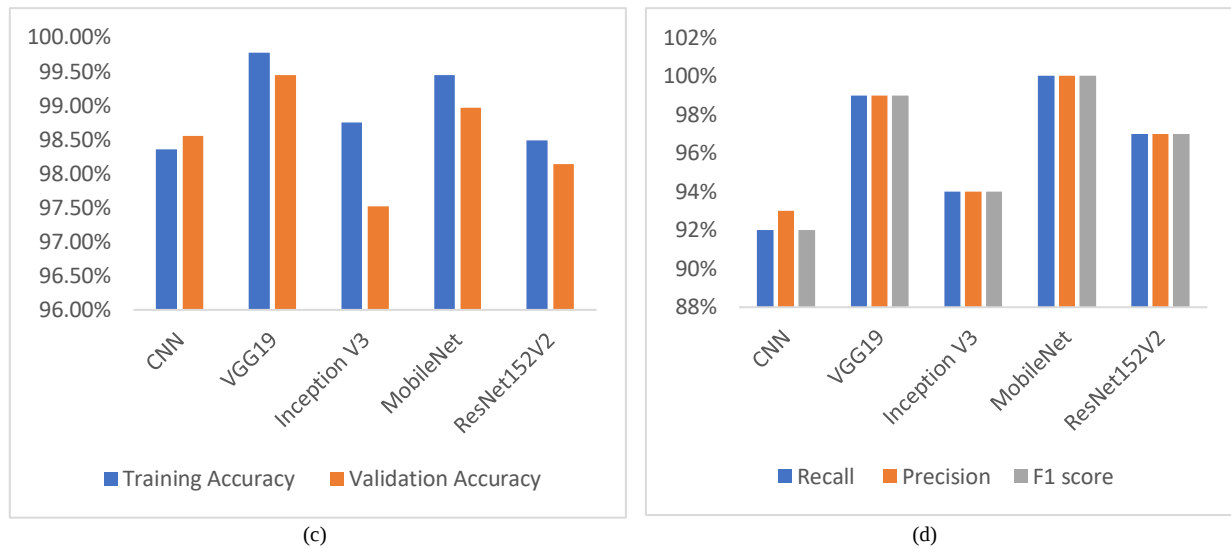


Fig. 8. (a) Comparing training and validation accuracy for the tomato dataset, (b) Comparing precision, recall, and F1-score for the tomato dataset (c) Comparing training and validation accuracy for the apple dataset, (b) comparing precision, recall, and F1-score for apple dataset

Prediction of tomato leaves



Fig. 9. Prediction using CNN model



Fig. 10. Prediction using VGG19

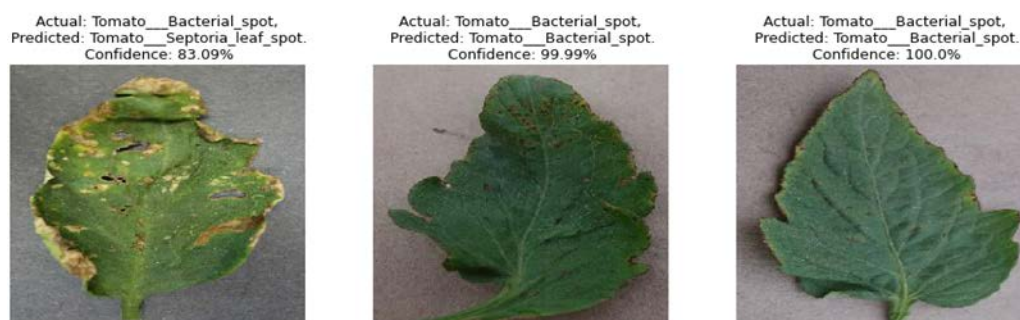


Fig. 11. Prediction using Inception V3

Prediction for apple dataset



Fig. 12. Prediction using CNN model

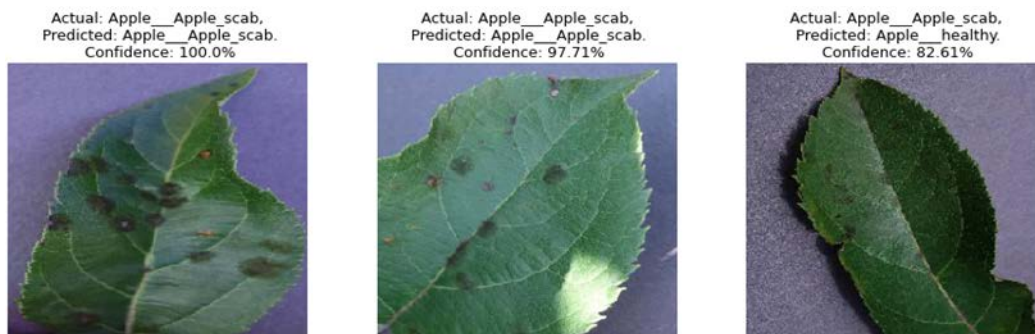


Fig. 13. Prediction using VGG19

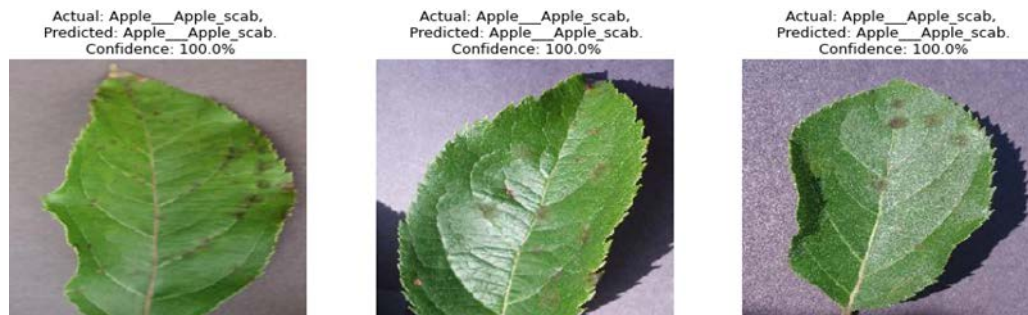


Fig. 14. Prediction using MobileNet Model

6. Conclusion and Future Work

Crop protection in farming is a difficult task. This requires a complete understanding of the crop being farmed, as well as any pests, diseases, or weeds that may be present. In this paper, we have used deep learning networks like transfer learning to build models and classify plant diseases. In this work, we have used both the tomato and apple datasets. MobileNet gave a pretty good accuracy compared to other existing methods, for both datasets of more than 97%. However, the sequential CNN model gave pretty good accuracy but using the transfer learning model was easy and took less time to train compared to the CNN model. When networks are trained with more parameters, their performance generally improves. Automatic plant disease thus becomes easier with the trained models. Thus, it becomes easier for new users because the network running in the background, taking input from the visual camera and informing the user of the output instantaneously so that the user may take appropriate action. As a result, preventative measures can be done earlier. This research could help with early and automatic disease identification in tomato and apple plants, thanks to the use of cutting-edge technologies like mobile phones, drone cameras, and the wide use of the internet. In this work, only tomato and apple leaves are taken from the Plant Village datasets. The investigation had to be limited to tomato and apple images due to a lack of sufficient Random-Access Memory (RAM) storage. In the future, it would be fantastic to test the CNN models on the entire plant village dataset, which contains a large diversity of plants, to bring detection and identification to a wider range of plants.

References

- [1] Hassan, S. M., Maji, A. K., Jasiński, M., Leonowicz, Z., & Jasińska, E. (2021, June 9). Identification of plant-leaf diseases using CNN and Transfer-Learning Approach. MDPI. Retrieved April 25, 2022, from <https://www.mdpi.com/2079-9292/10/12/1388>.
- [2] Qi, H., Liang, Y., Ding, Q., & Zou, J. (2021, February 23). Automatic identification of peanut-leaf diseases based on Stack Ensemble. MDPI. Retrieved April 25, 2022, from <https://www.mdpi.com/2076-3417/11/4/1950>.
- [3] Trivedi, N. K., Gautam, V., Anand, A., Aljahdali, H. M., Villar, S. G., Anand, D., Goyal, N., & Kadry, S. (2021, November 30). Early detection and classification of tomato leaf disease using high-performance deep neural network. MDPI. Retrieved April 25, 2022, from <https://www.mdpi.com/1424-8220/21/23/7987/htm>
- [4] G. Geetha *et al* 2020, Plant Leaf Disease Classification and Detection System Using Machine Learning, <https://iopscience.iop.org/article/10.1088/1742-6596/1712/1/012012>.
- [5] Parekh, M. (2019, July 16). A brief guide to Convolutional Neural Network (CNN). Medium. Retrieved April 25, 2022, from <https://medium.com/nybles/a-brief-guide-to-convolutional-neural-network-cnn-642f47e88ed4>.
- [6] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, Zbigniew Wojna, Rethinking the Inception Architecture for Computer Vision, Proceedings of the IEEE conference on computer vision and pattern recognition, pages: 2818-2826, 2016.
- [7] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. Proceedings of the AAAI Conference on Artificial Intelligence, 31(1). <https://doi.org/10.1609/aaai.v31i1.11231>.
- [8] Karen Simonyan and Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, International Conference on Learning Representations, 2015.
- [9] L. Sifre. Rigid motion scattering for image classification. PhD thesis, Ph. D. thesis, 2014.
- [10] Teki, Satyanarayana murthy., Varma, M.K., Harsha. (2021). Brain tumour segmentation using U-net based adversarial networks. Traitement du Signal, IETA Publisher, Vol. 36, No. 4, pp. 353-359. <https://doi.org/10.18280/ts.360408>.
- [11] T.Satyanarayana Murthy, N.P.Gopalan, Sasidhar Gunturu, " A Novel Optimization based Algorithm to Hide Sensitive Item-sets through Sanitization Approach", International Journal of Modern Education and Computer Science, Vol.10, No.10, pp. 48- 55, 2018.
- [12] T.Satyanarayana Murthy, N.P.Gopalan, Datta Sai Krishna Alla, "The Power of Anonymization and Sensitive Knowledge Hiding Using Sanitization Approach", International Journal of Modern Education and Computer Science, Vol.10, No.9, pp. 26-32, 2018.
- [13] T.Satyanarayana Murthy, N.P.Gopalan, "A Novel Algorithm for Association Rule Hiding", International Journal of Information Engineering and Electronic Business, Vol.10, No.3, pp. 45-50, 2018.
- [14] Jidesh, P., and B. Balaji. 2018. "Adaptive Non-local Level-set Model for Despeckling and Deblurring of Synthetic Aperture Radar Imagery." International Journal of Remote Sensing 39 (20): 6540–6556.
- [15] Teki, S.M., Banothu, B., Varma, M.K. (2019). An un-realized algorithm for effective privacy preservation using classification and regression trees. Revue d'Intelligence Artificielle, 33(4): 313-319.

Authors' Profiles



Balaji Banothu received his M.Tech. Degree in Computer Science and Data Processing from IIT Kharagpur, West Bengal, India in 2010 and he received a Ph.D. degree in the Department of Mathematical and Computational Sciences from the National Institute of Technology, Karnataka, India. At present, he is working as an Assistant Professor in the Department of Computer Applications at the National Institute of Technology, Tiruchirappalli. His current research area is Image restoration methods, Deep Learning.



T. Satyanarayana Murthy received his M.Tech. Degree in Computer Science and Systems Engineering from Andhra University, Visakhapatnam, India in 2010 and he received a Ph.D. degree in the Department of Computer Applications from the National Institute of Technology, Tiruchirappalli, India. At present, he is working as an Associate Professor in the Department of Information technology at the Chaitanya Bharathi Institute of Technology, Hyderabad. His current research areas are Data Mining, Deep Learning and CBIR.



Dr. Ramu Kuchipudi received his M.Tech. Degree in Computer Science and Engineering from JNTU, Hyderabad, India in 2010 and he received a Ph.D. degree in the Department of Computer science and Engineering, JNTU, Hyderabad India. At present, he is working as an Associate Professor in the Department of Information technology at the Chaitanya Bharathi Institute of Technology, Hyderabad. His current research areas are Wireless Sensor Networks, Network Security and Deep Learning.

How to cite this paper: Banothu Balaji, T. Satyanarayana Murthy, Ramu Kuchipudi, "A Comparative Study on Plant Disease Detection and Classification Using Deep Learning Approaches", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.15, No.3, pp. 48-59, 2023. DOI:10.5815/ijigsp.2023.03.04