ISSN: 1803-3814 (Printed), 2571-3701 (Online)
https://doi.org/10.13164/mendel.2022.2.041

A Novel Light-Weight DCNN Model for Classifying Plant Diseases on Internet of Things Edge Devices

Trong-Minh Hoang[1,✉], **Tuan-Anh Pham**[1], **Van-Nhan Nguyen**[1]

[1]**Telecommunication Faculty No1, Posts and Telecommunications Institute of Technology, Vietnam**

hoangtrongminh@ptit.edu.vn✉, anhpt.b17vt021@stu.ptit.edu.vn, nhannv.b18vt314@stu.ptit.edu.vn

Abstract

One of the essential aspects of smart farming and precision agriculture is quickly and accurately identifying diseases. Utilizing plant imaging and recently developed machine learning algorithms, the timely detection of diseases provides many benefits to farmers regarding crop and product quality. Specifically, for farmers in remote areas, disease diagnostics on edge devices is the most effective and optimal method to handle crop damage as quickly as possible. However, the limitations posed by the equipment's limited resources have reduced the accuracy of disease detection. Consequently, adopting an efficient machine-learning model and decreasing the model size to fit the edge device is an exciting problem that receives significant attention from researchers and developers. This work takes advantage of previous research on deep learning model performance evaluation to present a model that applies to both the Plant-Village laboratory dataset and the Plant-Doc natural-type dataset. The evaluation results indicate that the proposed model is as effective as the current state-of-the-art model. Moreover, due to the quantization technique, the system performance stays the same when the model size is reduced to accommodate the edge device.

Keywords: *Deep Convolution Neuron Networks, Edge Computing, Multi-leaf disease image, Plant-Doc dataset.*

Received: 25 October 2022
Accepted: 17 November 2022
Online: 28 November 2022
Published: 20 December 2022

1 Introduction

Today, intelligent agriculture based on the Internet of things and artificial intelligence platforms is a key trend in technological innovation. Automated systems for information gathering, processing, and decision-making have helped farmers reduce their workload and increase productivity by utilizing the knowledge of other users. Computer vision rapidly identifies crop diseases, limits risks, and makes accurate estimations. [15] [25] [19].

Nowadays, agriculture requires cutting-edge technologies such as IoT [24] and AI [13] to maintain agricultural productivity. There are numerous examples of AI and ML being applied in the agri-food industry [5] because of the growing importance of AI in the food industry due to its capacity to reduce food waste, increase production security, and improve the cleaning of machines, diseases, and pests. The use of machine learning, a subset of artificial intelligence, has the potential to overcome several obstacles in the development of knowledge-based farming systems [3]. There is a large amount of research on the many machine-learning algorithms used in agriculture's diverse application sectors. Consequently, it is essential to establish the appropriate technique for assuring precision and consistency in a particular agricultural application.
Due to the development of machine learning models, image processing can now detect plant diseases quickly and efficiently. Some popular convolution neural network architectures today, including AlexNet, GoogLeNet, and ResNet, have been widely applied. Recent progress in plant disease detection research utilizing the CNN model for image identification applications has produced good results. On the other hand, Deep networks are learned by estimating the neural network's parameters to enhance mapping during training. Recent advances in mathematics and computer science have substantially improved this computationally intensive method. Consequently, the Deep convolutional neural network (DCNN) approach can give substantial advantages for problems involving inconsistent data, such as the plant disease data set [14]. Some laboratory examinations of data sets will produce results that differ from those obtained in nature. In this work, we examine the performance characteristics of the DCNN model versus the state-of-the-art model using data collected from the real world.

The paper is structured as follows: The following section briefly describes the related works. In Section 3, two sets of testing datasets and models are brief introductions. The proposed DCNN model and the results of performance evaluation and comparison will be presented in Section 4. Finally, the conclusions and future study directions are given.

2 Relate work

Because of the increasing exploration of IoT devices, Edge computing and AI solutions, particularly deep learning, are merged in a robust approach to providing edge intelligence. The limited resource availability of edge devices is one of the most difficult challenges to overcome. It prevents the implementation of accurate, sophisticated computations for deep learning algorithms. Consumption of computational resources or processing power is an additional issue that must be considered when deploying in a real-world scenario [21]. As a result, a vast portion of current research focuses on developing AI based on edge models.

Today, DNN is utilized as an effective inference engine in various applications. Outstanding is the customer-friendly services that, among other technologies, use voice assistants, machine translation, and image retrieval to provide individual convenience [10]. Despite this, the DNN model, the current core component of artificial intelligence, requires a substantial amount of processing and is becoming "wider" (more filters in a layer) and "deeper" (more layers) with billions of parameters and millions of floating point operations (FLOP). These 'deeper' models provide greater precision at the tradeoff of increased computational complexity. In addition to latency restrictions, cloud computing is associated with security and privacy concerns. In industrial development solutions and autonomous vehicles, the problem of response delay is exceptionally harmful [4], [2]. Hence, edge computing solutions can tackle this problem when edge computing systems are located close to data generation to process data locally and quickly [18] [29]. This geological distance unavoidably resulted in a "computational gap" between DNN models and edge systems with weaker capabilities. To implement DNNs on the actual edge device, two hardware- and software-based strategies have been and are being developed. Some hardware architectures, such as application-specific integrated circuits (ASICs), tensor processing units (TPUs), or FPGA-based acceleration approaches, are tailored to implement DNN models. This strategy is frequently expensive and does not support a large variety of DNN models. The software-based approach, on the other hand, offers flexibility and is less costly to design than hardware. The goal is to create DNN models suitable for utilization with edge systems while providing the appropriate performance and increasing accuracy. Typical proposals in this field concentrate on building lightweight DNN models, compressing models, and searching for new neural architectures. The study aims to develop an efficient and adaptive DCNN model with high accuracy. Moreover, the proposed model is reduced dimension by quantization method to be tailored with edge devices.

3 Reivew of models and dataset

This section introduces some standard ML models used for edge devices and 02 popular data sets in agriculture.

3.1 Applicated DNN models

SqueezeNet [9]. This machine learning (ML) model is used to build a low-complexity calculation module (Fire module), including convolution math distributed in squeeze and expand layers. Additionally, several 3x3 convolutions were replaced with 1x1 convolutions to reduce weight.
MobileNet [11]. This machine learning model replaces traditional convolution with depth-wise convolution to achieve more efficient outcomes and lower computational expenses. Depth-wise decomposes a standard (k x k x n) convolution into a (k x k x 1) depth-wise convolution and a (1 x 1 x n) point-wise convolution. Each input channel is convolutional with depth-wise and point-wise convolution operators that linearly aggregate depth-wise results to generate a channel/feature map. Depth-wise separable convolution can drastically cut computation costs, decreasing edge device inference time. MobileNetV2 incorporates a linear bottleneck and inverted residual blocks to improve performance and accuracy. MobileNetV3 integrates NAS and NetAdapt to provide a more accurate and efficient network topology.
ShuffleNet [28]. This model's strategy is to use convolution in groups and combine channels to reduce computational costs while maintaining accuracy. ShuffleNetV2 recommends split channels to improve performance.
GhostNet [7]. In this model, some features in the convolutional layers are highly correlated, leading to the Ghost modules' creation. It firstly uses conventional convolution to extract some intrinsic features and then creates additional features from the extracted intrinsic features using less expensive linear processes.
EfficientNet [23]. The advantage of this model is based on the influence of the three size ratios in the DNN model to scale appropriately to achieve better accuracy and fewer parameters.
In [11] showed some performance comparisons of the above popular models. In the previous study [1], we also evaluated the performance of the MobileNet V3 model on a specific dataset. It indicates through experimental findings that it can be deployed on typical edge devices. MobileNet V3 has an accuracy of 96.58%, a quick Inference/Initialization time of 127 ms and 11 ms, respectively, and uses a total of 7.4 MB of RAM, making it the most efficient option for a real farm. To match edge processors, data compression is a standard method for purposes such as: Eliminating redundant numbers or excessive parameters, quantizing to store DNN network weights or intermediate results in low-bit registers; or parametric distillation from the large model to the small model [11]. In this study, we apply the quantization method.

3.2 Dataset

In agriculture, disease detection on plants remains a complex process without the assistance of information technology. And computer vision technologies have proven particularly effective in the detection of disease. Due to differences in picture attributes, however, the datasets utilized by various DNN models generate diverse practical outcomes. To demonstrate the efficacy of the algorithms, the models mentioned above frequently employ the laboratory dataset (PlantVillage) and relatively few experimental models on the naturally collected dataset (Cropped-PlantDoc). To demonstrate the performance of the proposed model, we will apply it to the Cropped-PlantDoc dataset and compare it to other studies. Simultaneously, quantization approaches are utilized to highlight the concept's merits that can be effectively implemented on IoT edge devices. **PlantVillage dataset.** An extensive, validated dataset of photos of damaged and healthy plants is needed to develop accurate image classifiers for diagnostic applications of plant disease. Such a dataset did not exist until recently, and even smaller datasets were not openly accessible. The PlantVillage project has collected thousands of images of healthy and diseased plants that are freely accessible to the public. All images in the PlantVillage database were captured at experimental research stations and laboratories, with various brightness, environment, and other user-specified settings. Eventually, the terminal (smartphone user) will charge a picture in multiple "random" conditions. More than 50,000 of these images are currently hosted on www.plantvillage.org and are accessible to the public via American colleges (Penn State, Florida State, Cornell, and others). The dataset contains 54,303 images of healthy and unhealthy leaves, categorized by species and disease into 38 categories. Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Bell Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, and Tomato are covered. It comprises illustrations of 17 fungal diseases, four bacterial diseases, two molds (oomycete), two viral diseases, and one mite-borne disease. There are images of healthy leaves on 12 crop species that are not visibly damaged by disease.

Cropped-PlantDoc. Cropped-PlantDoc is a dataset for visually detecting leaf diseases. The collection contains a total of 2,598 data points on 13 plant species and 17 diseases, the labeling of which required around 300 hours of human effort. It is the first dataset to include data from natural, uncontrolled environmental conditions. This dataset was created by downloading images from Google and Ecosia in the farm environment. The dataset collected about 20,900 images using scientific and common names and belonged to 38 classes of the dataset. Plant Doc provides realistic images of healthy and diseased plants to create a publicly available dataset.
Cropped-PlantDoc Dataset. To show the difference between our dataset and PlantVillage, we built another Cropped-PlantDoc (C-PD) dataset by cropping the image using the box information envelope. Like PlantVillage, the idea is cut to only the leaf, but these images are low quality, small in size, and have different backgrounds. The total number of leaf shapes after trimming 2,598 photos to 9,216, i.e., 9,216 bound to boxes.

4 System model and experimental results

Unlike most existing studies, authors propose feeding a DCNN CIE Lab instead of RGB color coordinates [17]. Authors modified an efficient Inception V3 architecture [20] to include one branch specific for achromatic data (L channel) and another department specific for chromatic data (AB channels). This modification takes advantage of the decoupling of chromatic and achromatic information. Besides, splitting branches reduces the number of trainable parameters and computation load by up to 50% of the original figures using modified layers. They achieved a state-of-the-art classification accuracy of 99.48% on the Plant Village dataset and 76.91% on the Cropped-PlantDoc dataset. In this session, we will use the MobilenetV3 model with the plan doc dataset to compare the accuracy, then perform the quantization of the MobilenetV3 model and the author's [17] proposed model.

4.1 Our utilized model

MobileNetV3 was adapted to mobile phone CPUs through hardware recognition network architecture (NAS) searches supplemented by the NetAdapt algorithm and improved by new architecture advances. MobileNets [26] is a series of lightweight deep neural networks based on Depthwise Separable Convolutions. It was followed by the improved version of Version1, MobileNetV2. MobileNetV2 [16] continues to use Depthwise Separable Convolutions in addition to the following proposals: Linear bottlenecks and Inverted Residual Blocks (shortcut connections between bottlenecks). MobileNetV3 [8] achieves better performance with less FLOP with new improvements over its predecessors with the following new block architecture:

In contrast to the previous version of MobileNet, which was designed manually, MobileNetV3 relies on AutoML [30] [27] [6] to find the best possible architecture in the search space suitable for mobile computer vision tasks. To make the most of the search space, the author [8] pointed out two techniques sequences implemented in the MnasNet [22] and NetAdapt [26]. First, search for a rough architecture using MnasNet, using the enhanced learning feature to select the optimal configuration from a set of discrete choices. Then, refine the architecture using NetAdapt, an additional technique that cuts out unused activation channels to a small extent. It is possible to create large or small models to provide the best possible performance under different conditions. Later the author [8] also pointed

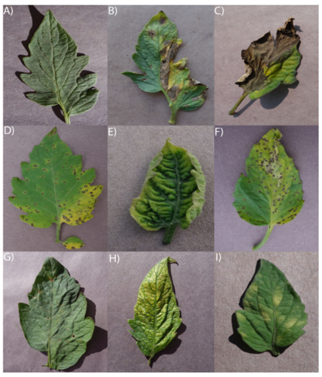

Figure 1: Example of potato images in Planvillage dataset

Figure 2: Example images of PlantDoc Dataset

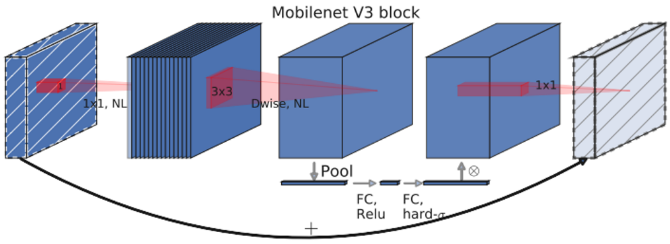

Figure 3: Blocks included in the MobiletnetV3 model

out how to improve the network by redesigning computationally costly layers and modifying the non-linear function to hard-swish (h-swish) based on Swish's nonlinear function so that it could overcome the most significant limitation of the Swish function being that it is very inefficient when computing on mobile hardware. The comparison of the soft and complex versions of sigmoid and swish nonlinearities is shown in figure 4. In his experiments, the author [8] found the complex version of all these functions to have no discernible difference in accuracy but multiple advantages from a deployment perspective. First, optimized implementations of ReLU6 are available on virtually all software and hardware frameworks. Second, in quantized mode, it eliminates potential numerical precision loss caused by different implementations of the approximate sigmoid. Finally, in practice, h swish can be implemented as a piece-wise function to reduce the number of memory accesses driving the latency cost down substantially.

4.2 Training Results

After training with Tesla P80 GPU with 16GB VRAM along with 12GB RAM using Google Colab, we train to achieve the best model with the following training results: First, we trained and tested both Color-Aware Two-Branch models with the LAB color system and compared them with the efficient DNN model MobileNetV3 on standard data sets taken in the laboratory with full conditions and lighting, the results achieved early model convergence to more than 99% in just a few epoch, the graph shows that efficient DNN model is mobile-net is optimized quite well and stable. Similarly, we trained and tested the Color-Aware Two-Branch model with the LAB color system and compared it with the efficient DNN model MobileNetV3 on the data set captured and collected closer to reality. In other words, it will be more challenging to train the model, and the results achieved MobileNetV3-Small model also soon converged to more than 99%

in just about 5 to 10 epochs. The graph shows that the efficient DNN model Mobilenet is optimized quite well and stable. In contrast to the Color-Aware Two-Branch model, convergence took up to 200 epochs to catch up with MobileNetV3.

4.3 Parameters and comparison results

The results are collected and indicated in the table above. With laboratory data, both models achieve high accuracy of 99.54% and 99.48%, respectively. But in addition to accuracy, there is also an essential factor for the lightweight model Parameters. We can see the difference when MobileNetV3-Small only needs 1.5M compared to 5M of Shuler's-Two-Branch, which still achieves better accuracy. For the data to be close to reality, to ensure accuracy, we replaced MobileNetV3-Small with MobileNetV3-Large. And the Color-Aware Two-Branch model selected an L-AB ratio of 50-50, corresponding to the highest accuracy rate that the author [17] proposed. When testing, the accuracy of both models has decreased compared to testing with previous data. However, the MobileNetV3 we used achieved 77.71% higher accuracy than Schuler's 76.91% with the same number of Parameters.

4.4 Experimental results on the reduced model by quantization

From the table above, we can see that with the efficient Model MobilenetV3-Large, after each quantum method, precision will reduce, especially when the compression ratio is higher, leading to a more profound accuracy reduction. Specifically, when the model is compressed to float-16-bit format, the accuracy is reduced to 0.70, and with full-int-8-bit, the accuracy is only 0.41. Similar to Precision, MobileNetV3's Recall is not only reduced but also reduced further when with full-int-8-bit quantization, Recall is only 0.22. In contrast to MobiletNetV3-Large, both Precision and Recall are virtually unchanged after each quantization process for the Color-Aware Two-Branch model. The results are similar for the Accuracy and F1-score parameters shown in Figure 7. The reason for this reduction is that the MobilenetV3 model, when designed, aimed to be optimized for the edge device, so continuing to perform quantization of the model will directly reduce the model's accuracy.

5 Conclusion

To choose a model suitable for edge devices but still ensure accuracy when classifying images of pests and diseases in agriculture, the paper presents the use of the MobilenetV3 model for laboratory dataset as well as dataset closer to reality, then compared with the best one and got better results in terms of both accuracies as well as the number of parameters used. In addition, the model quantization method is also applied to compare how the model accuracy changes after quantization. However, the MobilenetV3 model after quantization has significantly reduced accuracy, while the State-of-the-art model [17] is almost unaffected.

References

[1] Anh, P. T., and Duc, H. T. M. A benchmark of deep learning models for multi-leaf diseases for edge devices. In *2021 International Conference on Advanced Technologies for Communications (ATC)* (2021), IEEE, pp. 318–323.

[2] Baruah, S., Lee, P., Sarathy, P., and Wolf, M. Achieving resiliency and behavior assurance in autonomous navigation: An industry perspective. *Proceedings of the IEEE 108*, 7 (2020), 1196–1207.

[3] Benos, L., Tagarakis, A. C., Dolias, G., Berruto, R., Kateris, D., and Bochtis, D. Machine learning in agriculture: A comprehensive updated review. *Sensors 21*, 11 (2021), 3758.

[4] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316* (2016).

[5] Eli-Chukwu, N. C. Applications of artificial intelligence in agriculture: A review. *Engineering, Technology & Applied Science Research 9*, 4 (2019), 4377–4383.

[6] Gong, X., Chang, S., Jiang, Y., and Wang, Z. Autogan: Neural architecture search for generative adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 3224–3234.

[7] Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., and Xu, C. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 1580–1589.

[8] Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision* (2019), pp. 1314–1324.

[9] Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., and Keutzer, K. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360* (2016).

[10] Kepuska, V., and Bohouta, G. Next-generation of virtual personal assistants (microsoft cortana, apple siri, amazon alexa and google home). In *2018 IEEE 8th annual computing and communication workshop and conference (CCWC)* (2018), IEEE, pp. 99–103.

[11] Liu, D., Kong, H., Luo, X., Liu, W., and Subramaniam, R. Bringing ai to edge:

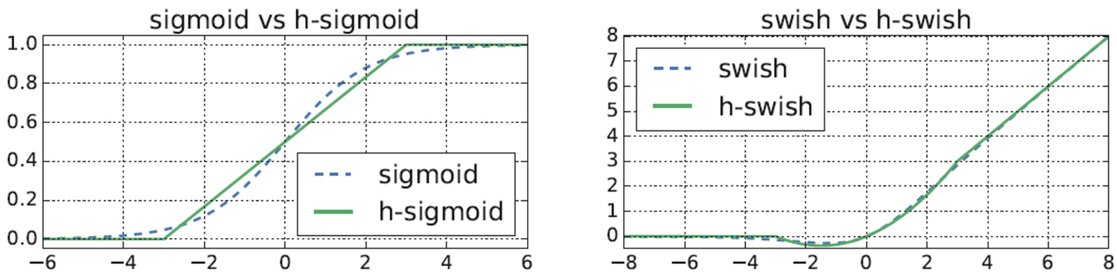

Figure 4: Non-linear cases 'Sigmoid' and Swish and variations

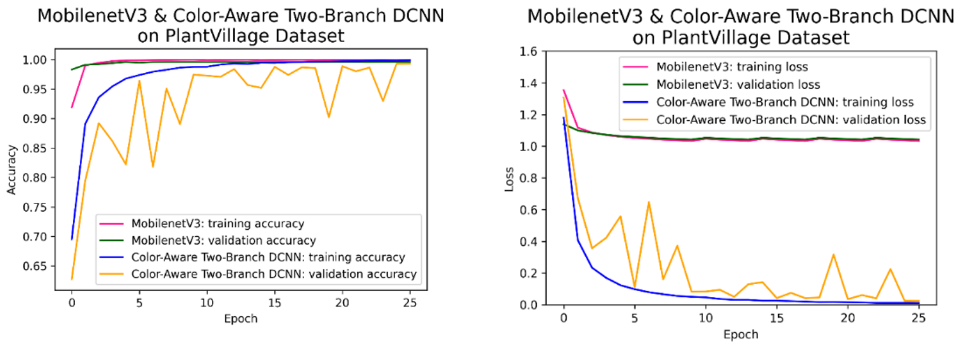

Figure 5: Graph of loss and accuracy, validation of two-model training on Plant Village dataset

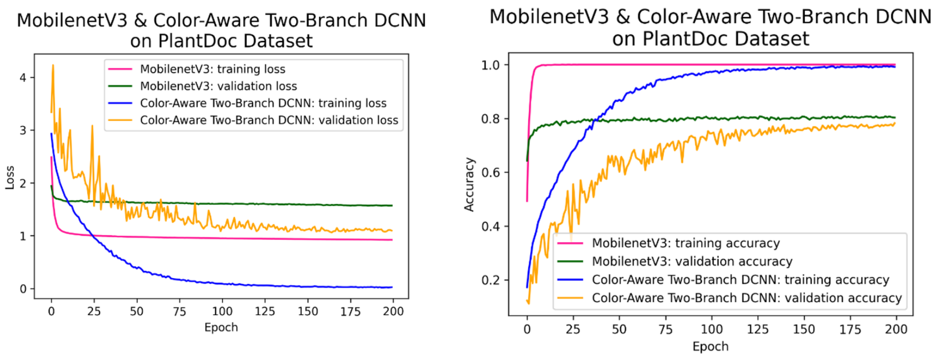

Figure 6: Graph of loss, accuracy, and validation of two-model training on Plant-Doc dataset

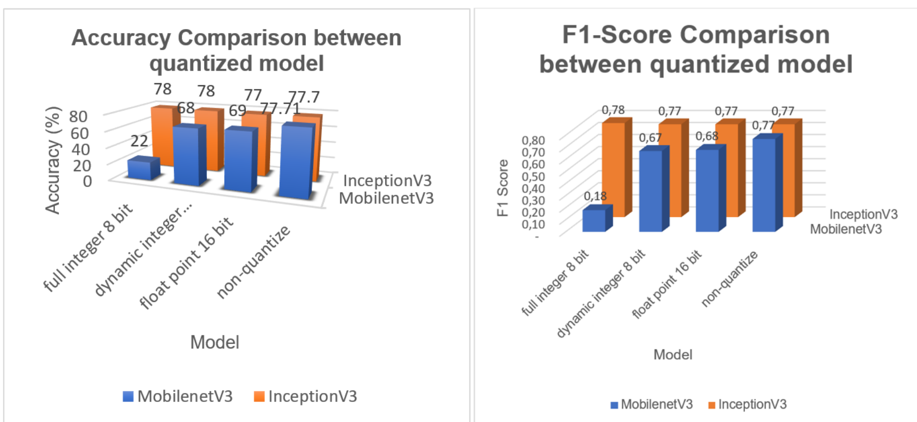

Figure 7: Comparison of Accuracy and F1-Score of Pre and Post Models Quantized at each level

Table 1: A comparison of common DNN model metric performances.

Model	Year	Parameters	FLOPs/MACs	Accuracy
SqueezeNet[9]	2016	1.24M	——	60.4%
Mobilenet[11]	2017	4.2M	/569M	70.6%
ShuffleNet[28]	2018	3.4M	/292M	71.5%
MobilenetV2[16]	2018	3.4M	/300M	72%
MobilenetV2-large[16]	2018	6.9M	/585M	74.7%
MobilenetV3-small[8]	2019	2.5M	/56M	67.4%
MobilenetV3-large[8]	2019	5.4M	/300M	75.2%
ShuffleNetV2[12]	2018	2.3M	146M/	69.4%
EfficientNet-B0[23]	2019	5.3M	390M/	77.1%
GhostNet[18]	2020	5.2M	141M/	73.9%

Table 2: Training model parameters using Plant Village Dataset

Author	Architecture	Color space	Parameters	Accuracy	F1
MobileNet V3 Small	Efficient CNN	RGB	1.5M	99.54%	0.9931
Schuler - Two-Branch	20%L + 80%AB	L–AB	5M	99.48%	0.9923

Table 3: Training model parameters using PlantDoc Dataset

Author	Architecture	Color space	Parameters	Accuracy	F1
MobileNet V3-Large	Efficient CNN	RGB	5M	77.71%	0.77
Schuler - Two-Branch	50%L + 50%AB	L–AB	5M	76.91%	0.76

Table 4: Results after performing quantization with training model with PlantDoc dataset

Model	Precision		Recall	
quantization	Our model	Schuler	Our model	Schuler
full integer 8 bit	0.41	0.78	0.22	0.78
dynamic integer 8 bit	0.68	0.78	0.68	0.78
float point 16 bit	0.70	0.77	0.69	0.77

From deep learning's perspective. *Neurocomputing* (2021).

[12] Ma, N., Zhang, X., Zheng, H.-T., and Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)* (2018), pp. 116–131.

[13] Mark, R. Ethics of using ai and big data in agriculture: The case of a large agriculture multinational. *The ORBIT Journal 2*, 2 (2019), 1–27.

[14] Negi, A., Kumar, K., and Chauhan, P. Deep neural network-based multi-class image classification for plant diseases. *Agricultural Informatics: Automation Using the IoT and Machine Learning* (2021), 117–129.

[15] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision 115*, 3 (2015), 211–252.

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

[17] Schuler, J. P. S., Romani, S., Abdel-Nasser, M., Rashwan, H., and Puig, D. Color-aware two-branch dcnn for efficient plant disease classification. *MENDEL Journal 28*, 1 (2022), 55–62.

[18] Shi, W., Cao, J., Zhang, Q., Li, Y., and Xu, L. Edge computing: Vision and challenges. *IEEE internet of things journal 3*, 5 (2016), 637–646.

[19] Sood, S., and Singh, H. Computer vision and machine learning based approaches for food security: A review. *Multimedia Tools and Applications 80*, 18 (2021), 27973–27999.

[20] Stachurski, J., and Toda, A. A. An impossibility theorem for wealth in heterogeneous-agent models with limited heterogeneity. *Journal of Economic Theory 182* (2019), 1–24.

[21] Strubell, E., Ganesh, A., and McCallum, A. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243* (2019).

[22] Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., and Le, Q. V. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 2820–2828.

[23] Tan, M., and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (2019), PMLR, pp. 6105–6114.

[24] Vangala, A., Das, A. K., Kumar, N., and Alazab, M. Smart secure sensing for iot-based agriculture: Blockchain perspective. *IEEE Sensors Journal 21*, 16 (2020), 17591–17607.

[25] Yang, J., Sharma, A., and Kumar, R. Iot-based framework for smart agriculture. *International Journal of Agricultural and Environmental Information Systems (IJAEIS) 12*, 2 (2021), 1–14.

[26] Yang, T.-J., Howard, A., Chen, B., Zhang, X., Go, A., Sandler, M., Sze, V., and Adam, H. Netadapt: Platform-aware neural network adaptation for mobile applications. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), pp. 285–300.

[27] Yang, Z., Wang, Y., Chen, X., Shi, B., Xu, C., Xu, C., Tian, Q., and Xu, C. Cars: Continuous evolution for efficient neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 1829–1838.

[28] Zhang, X., Zhou, X., Lin, M., and Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 6848–6856.

[29] Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., and Zhang, J. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE 107*, 8 (2019), 1738–1762.

[30] Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 8697–8710.