Resolution-Performance Trade-off of Plant Disease Detectors using Convolutional Neural Networks

Thanaporn Patikorn[1], Saowalak Leelawongsarote*[1], Arunee Chaisri[2], Namooy Panya[2]

[1] Department of Computer Science, Faculty of Science and Technology
[2] Department of Science, Faculty of Science and Technology
Rajamangala University of Technology Suvarnabhumi
[1]Nontaburi, Thailand
[2] Phra Nakhon Si Ayutthaya, Thailand
*saowalak.l@rmutsb.ac.th

Abstract— Artificial intelligence has been utilized in various fields to drive modern innovations. For instance, in agriculture, computer vision has been employed to identify plant diseases by analyzing images of plant leaves. While state-of-the-art models can achieve high accuracy, their practical application is still limited. These models are specifically designed to detect diseases from close-up images of leaves. However, in reality, images of plants of interest, such as those captured by fertilizer spraying drones and security cameras, are often scenic and contain multiple trees, each with numerous leaves. Consequently, the extracted leaf images have significantly lower resolutions compared to the ones used in the models. In this study, we investigated the impact of this trade-off between the performance of plant disease detection models and input image resolution. The relationship between image dimensions and accuracy was investigated. The results showed that halving image width resulted in approximately 4.35% decrease in accuracy.

Keywords: computer vision; image classification; convolutional neural networks; plant disease detection

I. Introduction

Agriculture, from the household level to the industry level, has greatly benefited from modern technologies. Science and engineering have been integrated into new agricultural innovations. New innovations are invented every year as new technologies emerge and become more affordable. Many such inventions are now widely used, such as automated greenhouses [1], apple harvesting machines [2], and fertilizer drones [3]. With the significant advancements in artificial intelligence performance and cost, computer vision has also been employed for detecting leaf diseases, as demonstrated in [4]. In that study, convolutional neural networks (CNNs) were trained to identify plant species and diseases. Specifically, more than 50,000 leaf images from 14 different species, whether healthy or diseased, were used to train the models. The best result from that study showed that the model could accurately identify the specific type of leaf and the name of the disease (or determine if it was healthy) with an accuracy of 99.35%. The dataset was published in [5] under the Creative Commons license, allowing any interested researchers to use it freely.

The accuracy from [5] made it seemed like this was a solved problem. However, in practice, these models were far from immediately useful due to the difference between the images used to train the models and leaf images collected in the real-life scenarios. The dataset from [5] consisted of close-up images of leaves, where a single image only showed a single leaf on a solid-color background, taken close to the normal angle (90 degree), as shown in Fig. 1. However, in real life scenarios, such images were very labor-intensive and expensive to collect. There were leaf images that were easy and cheap to collect such as images from security cameras and drones. However, more often than not, such images were taken far from trees and leaves and contained dozens to hundreds of leaves in a single image. In addition, such images would never have a solid-color background and leaves being at non-normal angles more often than not. Even if each leaf was extracted, the resolution would be much smaller than that of images used to train the model in [4], which were all of 256x256 pixels, as shown in Fig. 2.

Figure 1. Examples of images from [5]: healthy apple leaf (left) and apple leaf with apple scab (right)

Figure 2. Leaf images resolution comparison from a sample image

In the field of computer vision, it was generally accepted that there was a trade-off between model performance and image resolution (within certain thresholds). For leaf disease detection, knowing this trade-off allowed for better cost-performance analyses and decision making. For example, farm managers could plan drone routes such that leaf images from drone cameras are of sufficient resolution to achieve acceptable disease detection accuracy, public park caretakers could move or direct security cameras such that it had enough resolution to detect plant diseases or pests that could cause pandemic, and notify appropriate parties.

Thus, in this study, models were constructed to confirm the existence of the aforementioned trade-off, as well as estimating the numerical impact of the resolution to the plant disease detection models.

II. Related Works

Convolutional neural networks (CNNs) [6] are a machine learning technique that has been widely used, especially for image classification, including protected avian species detection [7], wildlife species identification in national parks in Russia [8], and, as mentioned earlier, leaf disease detection [4].

CNNs, like most neural networks, require a large dataset and extended training time to sufficiently detect complex patterns. Countless techniques have been introduced to mitigate this issue. For instance, transfer learning [9] involves reusing the inner layers of large pre-trained models as a foundation, with little to no fine-tuning, and focusing training efforts on only the final decision layer. In the case of CNNs, these inner layers generally capture common, less complex patterns in images, such as straight lines and curves. By transferring such patterns and knowledge, new models do not need to be trained from scratch, thereby necessitating less training data and time. Several large pre-trained CNN models are well-known and widely used, such as EfficientNet [10], MobileNet [11], and ConvNeXt [12].

III. Methodology

A. Dataset

The dataset utilized in this study was the PlantVillage Dataset [5], comprising over 50,000 leaf images encompassing both healthy and diseased specimens. These images span 14 distinct plant species, namely apple, blueberry, cherry, corn, grape, orange, peach, bell pepper, potato, raspberry, soybean, squash, strawberry, and tomato. The number of diseases contained within the dataset varies across species; some contain only images of healthy leaves (e.g. squash), while others encompass 9 different diseases in addition to healthy samples (e.g. tomato), resulting in a total of 26 diseases across all species represented in the dataset.

In previous studies utilizing [5], the dependent variables predominantly focused on leaf species and disease names. However, in this study, the analysis was narrowed down to discerning whether the leaves were diseased or healthy. This approach was adopted to enable the constructed models to identify common visual patterns and characteristics distinguishing diseased leaves from healthy ones. The aim was to establish patterns that could be extrapolated to previously unseen plant species and diseases.

B. Data Preprocessing

To investigate the trade-off between resolution and performance, the original dataset was transformed into five new image sets of varying resolutions: 8x8, 16x16, 32x32, 64x64, and 128x128 pixels. These resolutions, combined with the original 256x256-pixel dataset, constituted the six image sets utilized in this study.

Given that the primary objective was to develop a model capable of detecting diseased versus healthy leaves in a general context, only species providing both diseased and healthy leaf images were considered for inclusion in the image sets. Specifically, the following species were included: apple, cherry, corn, grape, peach, bell pepper, potato, strawberry, and tomato—encompassing 9 out of the total 14 species.

Furthermore, each image set was randomly divided into training and test subsets, maintaining a 50:50 ratio without replacement. Sampling was executed in a manner where an identical image from different image sets (with varying resolutions) was consistently placed either within the training set for all image sets or within the test set for all image sets. This approach was adopted to guarantee equitable comparisons of models across the various image sets and to minimize potential biases in trade-off measurements. Additionally, during the training phase, 20% of the training set was allocated for the validation process.

C. Convolutional Neural Networks and Transfer Learning

Convolutional neural network (CNN) models have demonstrated remarkable accuracy in this context, as observed in [4]. This study similarly adopted an approach involving transfer learning, leveraging a large pre-trained CNN model for model training.

To be specific, the CNN models were trained using Keras, an open-source Python library designed for neural networks and deep learning, built upon the Tensorflow framework. The chosen pre-trained CNN model for transfer learning was EfficientNet [10], renowned for its efficiency in transfer learning scenarios. The precise model variant utilized was efficientnetv2-s-21k-ft1k, configured with an input image resolution of 384x384 pixels.

In the investigation of the resolution-performance trade-off, six distinct CNN models were trained, each corresponding to one of the six image sets. Subsequently, every model was evaluated on two subsets: the test subset sharing the resolution of its training subset, and the test subset aligning with the original resolution. The evaluation encompassed measurements of model accuracy, loss, and training time.

D. Handling Overfitting

In machine learning, the hallmark of a robust model lies in its ability to perform effectively on novel, unseen data, also known as generalization. However, it is not uncommon to encounter models that excel on the training set yet falter when presented with unfamiliar data. This predicament, where a model molds itself excessively to the training data, diminishing its capacity to generalize, is termed overfitting [13]. Artificial neural networks, including CNNs, are often susceptible to overfitting due to their substantial number of underlying parameters. Consequently, various strategies have been devised to mitigate overfitting. In this study, two approaches were employed: data augmentation and dropout, both using Keras, culminating in the final CNN model depicted in Fig. 3.

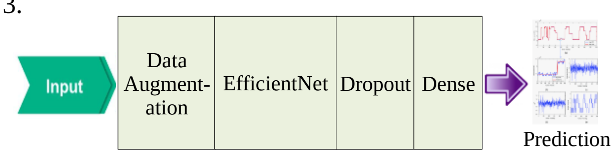

Figure 3. CNN model structure for plant disease detection

1) Data Augmentation

Data augmentation is a technique aimed at enhancing generalizability by modifying training data in a manner that preserves its correctness, significance, or patterns. The specific modifications vary depending on the nature of the data and the patterns to be retained. For instance, in the context of leaf disease detection, the label (diseased/healthy) remains unaffected when an image is horizontally flipped, rendering it a valid data augmentation technique. Within this study, training data augmentation encompassed horizontal flipping, image rotation, image translation, and image zooming.

2) Dropout

In neural network applications, many researchers theorized that overfitting may stem from nodes memorizing the training data or its distinctive characteristics. Dropout constitutes a strategy to counteract this behavior. During the training phase, instead of engaging all nodes across all layers, only a random subset is employed, while the remainder are "dropped out" of the network. In each subsequent weight update cycle, a distinct subset is chosen randomly for use versus dropout. This technique, known as dropout, compels the network to internalize broader patterns, as memorizing specific details becomes counterproductive when the corresponding nodes are subject to dropout [14].

IV. RESULT

The outcomes of training six distinct CNN models on six different image sets validated the resolution-performance trade-off, as illustrated in Table I.

TABLE I. THE TRAINING SET PERFORMANCE OF MODELS TRAINED ON 6 IMAGE SETS OF DIFFERENT RESOLUTIONS

Image Resolution (pixels)	Training Time (seconds)	Training Set		Validation Set	
		Loss	Accuracy	Loss	Accuracy
8x8	785.526	0.249	0.775	0.148	0.822
16x16	790.547	0.317	0.790	0.182	0.848
32x32	799.512	0.219	0.850	0.194	0.873
64x64	790.858	0.100	0.897	0.052	0.917
128x128	798.838	0.065	0.926	0.041	0.939
256x256	782.012	0.042	0.948	0.025	0.968

Subsequently, these models were assessed on both the test subset of the same resolution as their corresponding training subsets (Tab. II) and the test subset of the original resolution (Tab. III), with evaluations conducted for each plant species and overall accuracy. Notably, all models exhibited superior performance compared to the baseline accuracy (majority class = 72.2%). However, an exception was observed in the case of the 16x16 model, which displayed anomalously low accuracy for potato images at 16x16 pixels, and both potato and tomato images at 256x256 pixels (orange highlight in Tab. II and Tab. III)

TABLE II. MODEL ACCURACY WHEN TESTED ON THE TEST SUBSET OF THE SAME RESOLUTION AS THE TRAINING SUBSET

Model	**Accuracy**									
	Strawberry	**Tomato**	**Potato**	**Cherry**	**Corn**	**Peach**	**Apple**	**Grape**	**Bell Pepper**	**Overall**
8x8	0.754	0.860	0.908	0.689	0.843	0.897	0.588	0.826	0.628	0.812
16x16	0.837	0.832	0.658	0.895	0.833	0.900	0.626	0.877	0.741	0.813
32x32	0.711	0.913	0.948	0.962	0.717	0.891	0.592	0.908	0.745	0.853
64x64	0.869	0.929	0.970	0.996	0.943	0.945	0.812	0.932	0.822	0.919
128x128	0.989	0.953	0.980	0.989	0.979	0.933	0.943	0.993	0.861	0.956
256x256	0.997	0.967	0.991	0.994	0.970	0.933	0.965	1.000	0.897	0.968

TABLE III. MODEL ACCURACY WHEN TESTED ON THE TEST SUBSET OF THE ORIGINAL 256X256 RESOLUTION

Model	Accuracy									
	Strawberry	Tomato	Potato	Cherry	Corn	Peach	Apple	Grape	Bell Pepper	Overall
8x8	0.674	0.887	0.933	0.468	0.738	0.887	0.485	0.894	0.545	0.794
16x16	0.711	0.480	0.268	0.513	0.683	0.811	0.514	0.890	0.586	0.572
32x32	0.735	0.755	0.926	0.481	0.834	0.752	0.817	0.927	0.936	0.792
64x64	0.837	0.927	0.985	0.926	0.855	0.968	0.968	0.983	0.975	0.934
128x128	1.000	0.953	0.989	0.956	0.973	0.960	0.960	0.998	0.979	0.966
256x256	1.000	0.953	0.989	0.956	0.973	0.960	0.960	0.998	0.979	0.966

Interestingly, models trained using lower resolutions performed substantially worse on cherry leaf images of original resolution than on lower resolution (pink in Tab. III).

Excluding the aforementioned trade-off, the resolution-performance trade-off could be observed across plant species, excluding the aforementioned anomaly.

To uncover the trend within the trade-off, the model accuracy was subjected to linear regression analysis across four distinct scales of input resolution: 1) image width, which was identical to height, given square input images ($width$), 2) image area ($area$), 3) log based 2 of the image width ($\log_2 width$), and 4) log based 2 of log based 2 of the image width ($\log_2 \log_2 width$). This exploration yielded four linear regression equations, each accompanied by corresponding R^2 values of 0.333370, 0.230693, 0.422520, and 0.407801, respectively (Fig. 4). Among these equations, the linear regression model with $\log_2 width$ demonstrated the best fit, characterized by a coefficient (m) of 0.04349 and a y-intercept (c) of 0.63361, as shown in equation (1).

$$accuracy = 0.04349\,(\log_2 width) + 0.63361 \quad (1)$$

The base-2 logarithm signifies that alterations in width are evaluated within the context of powers of two i.e. doubling and halving. For instance, with a width of 32, the model projects an accuracy of 0.85106. Upon halving the width to 16, the accuracy diminishes to 0.80757, marking a decrement of 0.04349. Conversely, when the width is doubled to 64, the accuracy elevates to 0.89455, concurrently representing a decrease of 0.04349.

V. DISCUSSION AND FUTURE WORKS

The results vividly illustrate a robust positive correlation between image resolution and model accuracy, thereby confirming the presence of a resolution-performance trade-off within CNN models designed for plant disease detection. Specifically, considering square leaf images, a reduction of 1 in $\log_2 width$ was linked to a decrease of 0.043489 in accuracy. In simpler terms, if the image width is halved (resulting in a quartered area), the accuracy would diminish by approximately 4.35%.

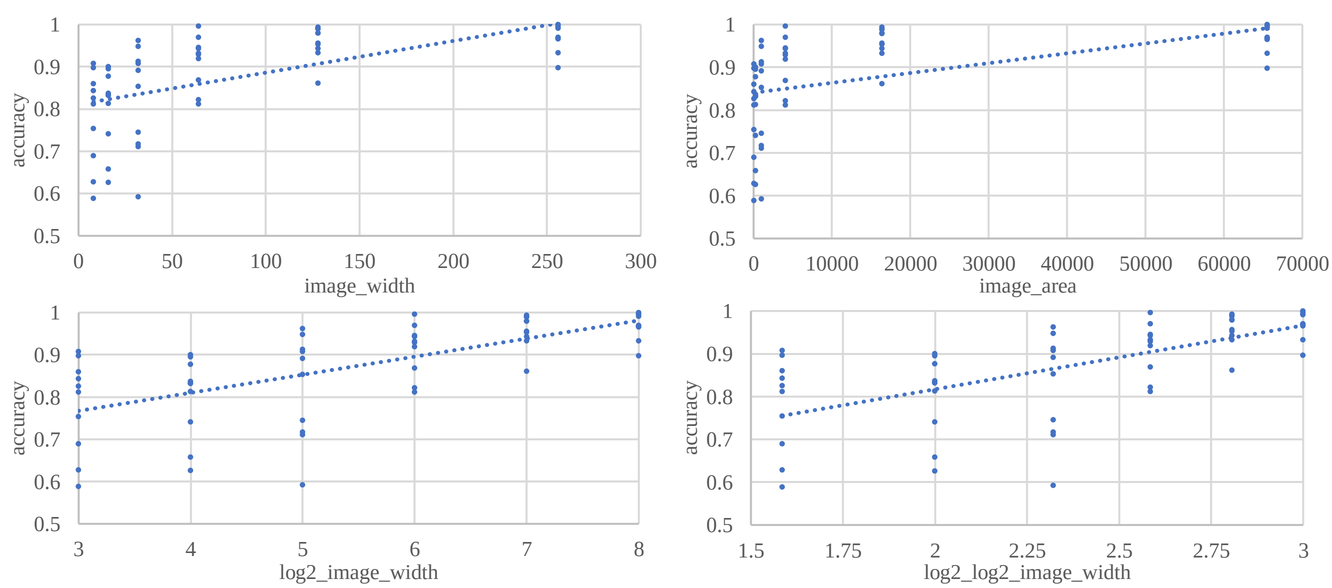

Figure 4. Linear regression comparison between different scales of resolution and accuracy: 1) $width$ (top left), 2) $area$ (top right), 3) $\log_2 width$ (bottom left), and 4) $\log_2 \log_2 width$ (bottom right)

Addressing the anomaly observed within the 16x16 model—where accuracy significantly dropped for tomato and potato images—it could potentially be attributed to overfitting or the emergence of image artifacts accentuated as resolutions declined. Further investigation is warranted, involving supplementary data augmentation techniques such as brightness/darkness adjustments and grayscale conversions.

Furthermore, for a comprehensive understanding, this trade-off should be subjected to further scrutiny using images derived from real-world scenarios, including those captured by security cameras and agricultural drones. Such images typically present a higher degree of complexity, encompassing elements like multiple leaves, overlapping foliage, non-square formats, leaves captured from non-normal angles, and lens imperfections. All of these variables hold the potential to exert considerable influence on model accuracy and, in turn, on the rate of accuracy decline as resolution diminishes.

Acknowledgment

We would like to thank the Research Unit of Smart Digital Technology Innovation, Faculty of Science and Technology, Rajamangala University of Technology Suvarnabhumi for various support throughout the research.

References

[1] T. Bhuvaneswari and J. T. H. Yao, “Automated greenhouse,” in *2014 IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*, 2014, pp. 194–199.

[2] M. E. De Kleine and M. Karkee, “A semi-automated harvesting prototype for shaking fruit tree limbs,” *Transactions of the ASABE*, vol. 58, no. 6, pp. 1461–1470, 2015.

[3] P. Mhetre, D. Soni, A. Nerkar, and H. Vishal, “Agriculture drone for fertilizer spraying,” *International Research Journal of Modernization in Engineering Technology and Science*, vol. 2, no. 6, 2020.

[4] S. P. Mohanty, D. Hughes, and M. Salathé, “Inference of plant diseases from leaf images through deep learning,” *Front. Plant Sci*, vol. 7, p. 1419, 2016.

[5] D. Hughes, M. Salathé, and others, “An open access repository of images on plant health to enable the development of mobile disease diagnostics,” *arXiv preprint arXiv:1511.08060*, 2015.

[6] H. Luo, C. Xiong, W. Fang, P. E. Love, B. Zhang, and X. Ouyang, “Convolutional neural networks: Computer vision-based workforce activity assessment in construction,” *Automation in Construction*, vol. 94, pp. 282–289, 2018.

[7] M. Favorskaya and A. Pakhirka, “Animal species recognition in the wildlife based on muzzle and shape features using joint CNN,” *Procedia Computer Science*, vol. 159, pp. 933–942, 2019.

[8] Y. Harjoseputro, I. Yuda, K. P. Danukusumo, and others, “MobileNets: Efficient convolutional neural network for identification of protected birds,” *IJASEIT (International Journal on Advanced Science, Engineering and Information Technology)*, vol. 10, no. 6, pp. 2290–2296, 2020.

[9] E. S. Olivas, J. D. M. Guerrero, M. Martinez-Sober, J. R. Magdalena-Benedito, L. Serrano, and others, *Handbook of research on machine learning applications and trends: Algorithms, methods, and techniques: Algorithms, methods, and techniques*. IGI global, 2009.

[10] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International conference on machine learning*, 2019, pp. 6105–6114.

[11] A. G. Howard *et al.*, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[12] S. Woo *et al.*, “Convnext v2: Co-designing and scaling convnets with masked autoencoders,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16133–16142.

[13] X. Ying, “An overview of overfitting and its solutions,” in *Journal of physics: Conference series*, 2019, vol. 1168, p. 022022.

[14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.