

# Herbal plant recognition using deep convolutional neural network

Izwan Asraf Md Zin<sup>1</sup>, Zaidah Ibrahim<sup>2</sup>, Dino Isa<sup>3</sup>, Sharifah Aliman<sup>4</sup>, Nurbaity Sabri<sup>5</sup>,  
Nur Nabilah Abu Mangshor<sup>6</sup>

<sup>1,2,4,5,6</sup>Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Malaysia

<sup>3</sup>CONNECT Initiative, Crops for the Future, Malaysia

---

## Article Info

### Article history:

Received Jan 10, 2020

Revised Mar 16, 2020

Accepted Apr 20, 2020

### Keywords:

Convolutional neural network

Data augmentation

Deep learning

Herbal plant recognition

---

## ABSTRACT

This paper investigates the application of deep convolutional neural network (CNN) for herbal plant recognition through leaf identification. Traditional plant identification is often time-consuming due to varieties as well as similarities possessed within the plant species. This study shows that a deep CNN model can be created and enhanced using multiple parameters to boost recognition accuracy performance. This study also shows the significant effects of the multi-layer model on small sample sizes to achieve reasonable performance. Furthermore, data augmentation provides more significant benefits on the overall performance. Simple augmentations such as resize, flip and rotate will increase accuracy significantly by creating invariance and preventing the model from learning irrelevant features. A new dataset of the leaves of various herbal plants found in Malaysia has been constructed and the experimental results achieved 99% accuracy.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



---

## Corresponding Author:

Izwan Asraf Md Zin,  
Faculty of Computer and Mathematical Sciences,  
Universiti Teknologi MARA,  
40450 Shah Alam, Selangor, Malaysia.  
Email: izwanzin@gmail.com

---

## 1. INTRODUCTION

Malaysia's high humidity and warm climate provide optimal growth to numerous species of fauna contributing to 12,500 species of seed plants with more than 740 endemic species [1]. There were 2013 herbal plants recorded in peninsular Malaysia with the additional survey conducted recently registered another 68 species in Gunung Ledang, Johor [2, 3]. In Malaysia, herbal plants study focuses more on the usage and treatments due to commercial value. Cataloguing herbaceous plants can be frustrating as they first need to be identified, followed by discussions with local folks on the usage and method of applications. Not to mention that the availability of the publicly accessible comprehensive database more often caused repetitive works. This process can be shortened by deploying automated herbal plant identification using deep learning.

Rapid technological advancement has significantly assisted in accelerating the ideas into reality. Mobile phone and digital camera can now capture useful information with higher quality image as well as the location or coordinates of where the picture was taken. Although the later may be underutilised, it can be beneficial if it's being utilised in activities such as plant mapping and geotagging. Conventional plant identification method can be time-consuming as it requires in-depth knowledge as well as careful examination of the plant phenotypes. Plant identification can be done in multiple methods such as colours, flowers, leaf, textures and structures. The conventional method of plant identification follows generic plants

hierarchy such as vascular or nonvascular, spore-producing or seed-producing and finally non-flowering or flowering. These methods can be complicated due to the enormous numbers of plant species and also similarities of the plants when it reaches the family level. Additionally, it requires a botanist to identify the exact plants which is frustratingly time-consuming due to the method of using plant morphology as identification keys [4-5]. A botanist usually examines one or more characteristics of a plant such as leaf shape, bark and petal before concentrating on the unique feature of the plant to deduce the plant species [6-7]. Series of questions are normally needed before a botanist could confirm the species of the plant. Leaf is commonly chosen as the primary input in plant identification as every plant possesses these characters unlike other parts such as flower and barks. Plants are also easily identified by using leaf due to the distinguishable feature such as shapes, veins and blades. However, there are always instances where the plant shares similarities especially when it comes to family level thus making identification process to be challenging even for experienced botanists [8-11]. Naturally, this would call for greater urgency in seeking more efficient plant identification procedure which can be used in plant identification as well as conservation plan and disease management [12-13]. Deep learning models such as convolutional neural network (CNN) provides excellent aid in plant identification as it extracts hierarchical representation of the input data and greater feature extraction such as size patches on different branches [14-16]. Unlike the conventional method, CNN would significantly shorten the time in plant identification and this can be further enhanced with GUI deployment that could show additional information about the identified plants.

## 2. METHOD

This study focuses on the application of deep learning in herbal plants identification that are commonly available and used for traditional medicinal purposes. Random herbal plants were selected and captured at a herbal nursery at Jalan Kebun, Shah Alam in January 2019. Twelve plants were photographed with at least ten images each using a mobile phone (Huawei P20 Pro) in different angles to increase the varieties of the photos. Each image was captured using phone camera default settings which were; ten Megapixels with the dimension of 2736 pixels x 3648 pixels. Based on the images captured, ten images were selected for each plant for CNN model creation. Additionally, these images underwent pre-processing steps as explained below. Figure 1 shows the sample of the images that have been captured.



Figure 1. Samples of captured images

### 2.1. Data pre-processing

Data pre-processing was performed using IMBatch® application due to its capability in handling multiple tasks for image processing. Data augmentation methods were performed using the configurations explained below and results of the augmentation can be seen in Figure 2. Running a CNN model against original sizes requires more computational power, therefore, the images were resized to 160 pixels x 120 pixels with 95% jpeg loss less transformation from the original format to ensure image features are retained while at the same time significantly reducing the processing power requirement. Each image was rotated by

ten-degree for the first three images and 30 degrees subsequently to increase the training data. This process was repeated 16 times from previously rotated images to create an additional number of 160 images. Thus, a total number of 170 images for each plant were generated during this process. Since the image rotation process performed previously created different sizes of images, each image was resized again to 160 pixels x 120 pixels to maintain a standard format across all pictures. The samples were grouped into two categories where one dataset contains only 10 images for each plant and another dataset contains 170 images generated during the pre-processing phase for each plant. The model was tested against these datasets and later was fine-tuned to achieve better performance.

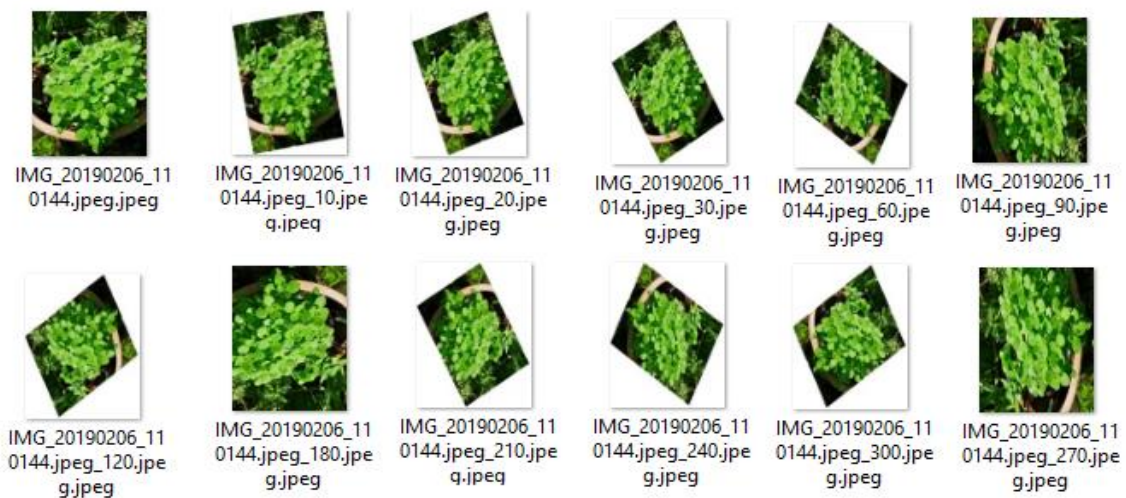


Figure 2. Sample plant images for dataset 2

Proposed CNN model was executed using Matlab© 2018b with GPU of 1080TI and 3584 CUDA cores to utilise the hardware for higher processing power. Table 1 shows the proposed parameters for CNN architecture used while Table 2 describes each model that was used in the test.

Table 1. The proposed parameters for CNN architecture

Layer	Parameter
Convolve layer	3 x 3
Pooling Layer	2 x 2 with Maxpooling
Activation function	ReLU where $f(x)=\max(x,0)$
Classification layer	Softmax

Table 2. Model description

Model	Dataset	Class	Total sample	Layers	Convolve	Pooling	Epoch
Model 1	1	12	120	5	3 x 3	Max	30
Model 2	1	12	120	7	3 x 3	Max	30
Model 3	1	12	120	9	3 x 3	Max	30
Model 4	1	12	120	11	3 x 3	Max	30
Model 5	1	12	120	9	3 x 3	Max	30
Model 6	1	12	120	9	5 x 5	Max	30
Model 7	1	12	120	9	7 x 7	Max	30
Model 8	1	12	120	9	3 x 3	Average	30
Model 9	1	12	120	9	3 x 3	Average	100
Model 10	2	12	2040	5	3 x 3	Max	30
Model 11	2	12	2040	9	3 x 3	Average	30
Model 12	2	12	2040	9	3 x 3	Average	28-47
Alexnet	2	12	2040	AlexNet	Default	Default	Default
GoogLeNet	2	12	2040	GoogLeNet	Default	Default	Default
SqueezeNet	2	12	2040	SqueezeNet	Default	Default	Default

\*Note: Model 8 and 11 are based on the same configuration with different dataset  
 Model 1 and 10 are based on the same configuration with different dataset

### 3. RESULTS AND DISCUSSION

Each model with different configurations was experimented and run ten times where results were logged for performance calculation.

#### 3.1. Multi layers effect

Experiment was conducted on dataset 1 with different number of layers in order to see the effect of a high number of layers towards plant recognition accuracy. Figure 3 illustrates the performance of different CNN models with different number of layers. Significant improvement can be seen in accuracies when the layers were increased and slightly decreased when 11 layers were applied. The proposed model (model 1) with 5 layers has an average accuracy of 41.11%, 7 layers (71.11%), 9 layers (77.22%) and 11 layers (70.83%). While adding more layers would provide more features maps, the excessive layer may cause overfitting where additional features that are not part of the object were classified as the actual object [17]. Since model 3 performed better than the rest of the models and ran at a shorter time, this model was selected to be tweaked in the next experiment.

Multi Layer Effect (Dataset 1)

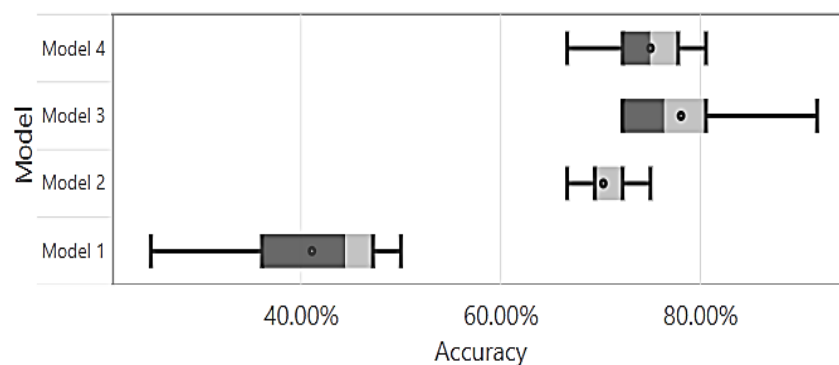


Figure 3. Multi-layer effect to accuracy

#### 3.2. Convolve layers features detector

This experiment focuses on fine-tuning convolve layer based on the model chosen previously which is model 3. Figure 4 illustrates the performance of different convolve layers where convolve layer with 3x3 kernel size has higher performance with an average accuracy of 77.22% compared to 5x5 with an average accuracy of 66.67%. Nonetheless, 7x7 kernel size could not be implemented since the input images are not large enough for convolving process to complete the model. Convolve layer acts as a learning filter where it will perform feature extraction. Since the input image in this test has been resized to 160x120 pixels, having lower input of feature detector is more suitable and provides more accurate features extraction compared to higher number of feature detector.

Convolve Layer Feature Map (Dataset 1)

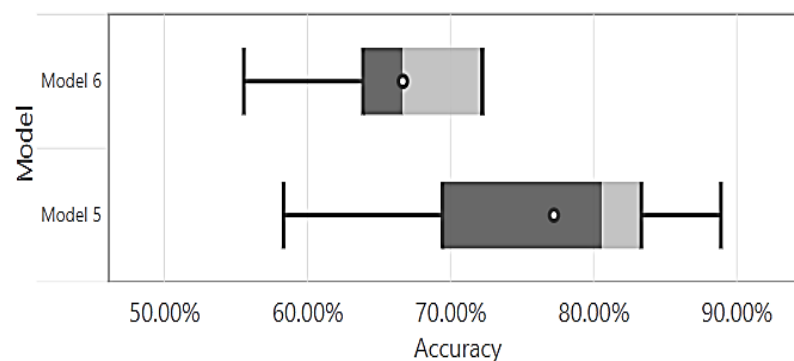


Figure 4. Convolve layer feature map effect

### 3.3. Pooling layer

This experiment provides a comparison between pooling parameter using model 3 and model 8. Figure 5 illustrates the performance of different pooling method where average pooling with an average accuracy of 80.00% outperformed max pooling with an average accuracy of 77.22%. Max pooling is used to select brighter pixels from an image while average pooling smoothes the images. Pooling was used since it reduces sensitivity to image variances by aggregating the local activations to global representation. Besides that, it reduces computational efficiency by down-sizing the feature dimension [18-19]. Model 8 was used for the next experiment since it outperforms model 3.

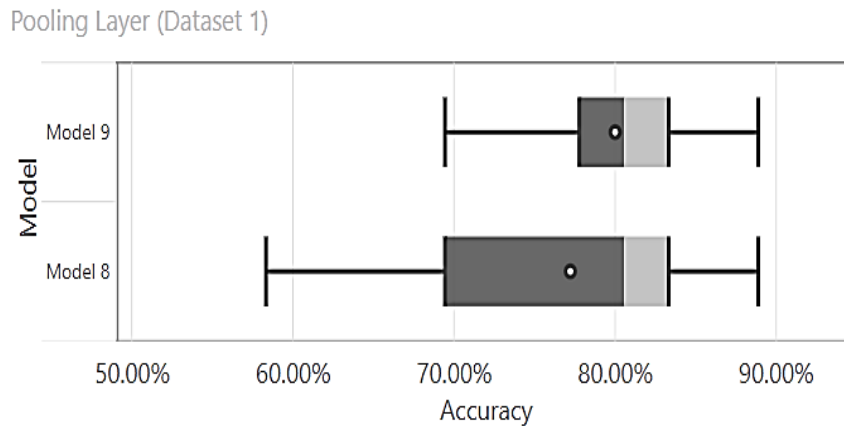


Figure 5. Pooling layer effect

### 3.4. Epoch

Based on the experiments conducted earlier, model 8 is replicated with different epoch options. The default training options was set to 30 and in general higher epoch number may contribute to higher accuracy as it increases the round of optimization of the model during training process. The number of epoch for model 9 was increased to 100 and another condition was added where the training will automatically stop once the model achieved the same accuracy three times consecutively. Figure 6 illustrates the performance of models with different number of epoch where model 8 achieves slightly higher average accuracy compared to the model run with higher number of epoch. Similarities of accuracies achieved between different epoch are caused by the low number of sample data [20]. Therefore, it is suggested that the model with low number of samples to be run at small number of epoch as it can achieve similar accuracy without consuming additional resources.

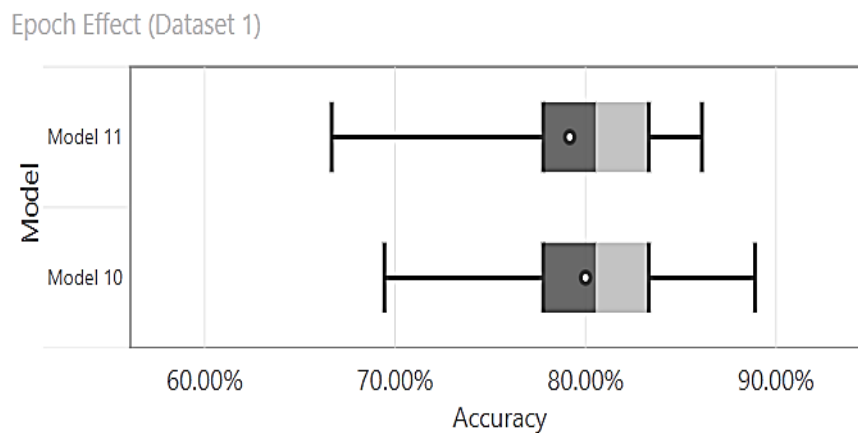


Figure 6. Epoch configuration

### 3.5. Data augmentation

This experiment focuses on the effect of data augmentation on the model performance where models were tested with dataset 2 containing 2040 samples. Dataset 2 samples were generated by augmenting methods mentioned in the data pre-processing section. Figure 7 illustrates the model's performances between model 10, model 11 and model 12. Model 11 which was built based on model 8 performed slightly better with an average accuracy of 99.56% compared to model 10 (99.41%) and model 12 (99.26%). Significant increase of accuracies across all models against dataset 2 indicates the importance of variations of sample data since it will significantly improve the performance and reduce overfitting [21-23].

Model 12 was experimented with an added condition where it will stop the running process when the accuracy achieved three similar results consecutively during every iteration. While this model has the lowest average of accuracy compared to the other model, it can be seen that some of the instances were running at lower epoch and shorter time. Additionally, model 12 also achieved the lowest average loss compared to the other models. Table 3 summarizes the results produces by model 10, 11 and 12.

Multi Layer Comparison (Dataset 2)

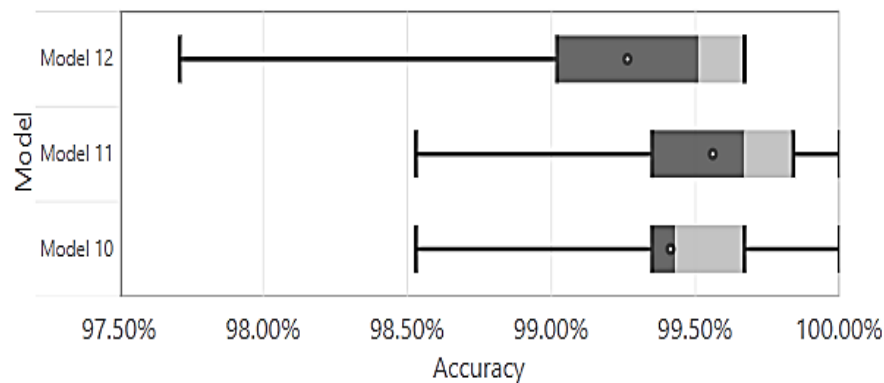


Figure 7. Multi-Layer comparison using dataset 2

Table 3. Model description with the best results

Model	Min	Max	Avg	SD	Epoch	Time (Sec)	Avg Loss
10	0.9853	1.0000	0.9941	0.0044	30	31 - 32	0.0306
11	0.9853	1.0000	0.9956	0.0042	30	31 - 32	0.0255
12	0.9771	0.9967	0.9926	0.0062	22 - 47	23 - 49	0.0222

### 3.6. Data augmentation performance comparison with pretrained models

This experiment compares the best performance model with pretrained models namely AlexNet, GoogleNet and SqueezeNet. These pretrained models are available and can be used with Matlab©. Dataset 2 images were resized to 227x227 pixels as this is the requirement needed by these pretrained models. Figure 8 illustrates the performance of pretrained models with the best model where model 11 has outperformed pretrained models. This shows that the model selected has better recognition performance compared to pretrained models that have higher number of layers due to invariance. Invariance occurred since the model was trained and validated against dataset 2 where the images were rotated and flipped multiple times to provide more training samples with different variations to prevent the model from learning irrelevant patterns [24-25].

Although our model outperforms pretrained models, this only occurred against dataset 2 which contains augmented images. These pretrained models have similar performance against dataset 1 that contains only 120 images where our model performs poorly as mentioned in the first test. This is due to the complexity of these pretrained models that makes them perform better even with lower samples. However, these pretrained models require GPU computational power while our model will still be run using CPU computational power.

Comparison with Pretrained Models (Dataset 2)

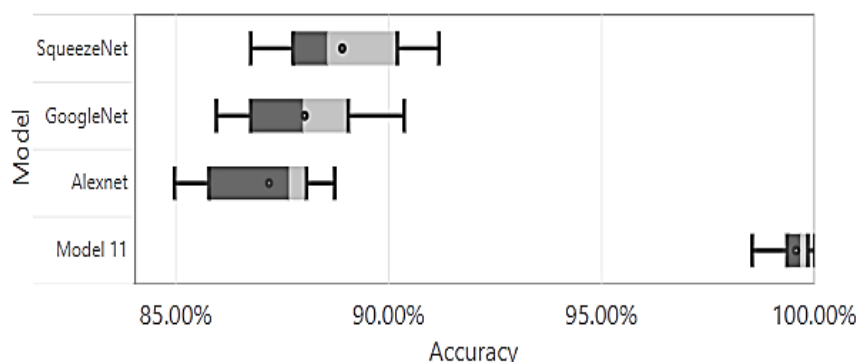


Figure 8. Comparison with pretrained models

#### 4. CONCLUSION

Based on the experiments conducted previously, it can be concluded that high accuracy can be acquired when more sample data was used and by increasing the complexity of the model. The high number of layers achieved higher accuracy as it learns more features which is important in classifying the object. Additionally, increasing the complexity of the layers helped in getting a reasonable accuracy with a very small sample value with lower processing times. Fine-tuning the model would also, in general, helps to increase the accuracy such as adding more convolve and pooling layers but it can be sample dependent because based on the experiment performed, the number of accuracy gained was small with high number of layers. The experiment performed also showed that data augmentation in general is able to increase the accuracy significantly as it enables the model to learn more features. Simple image augmentation such as rotation and flip were sufficient to create invariance and allow the model to learning relevant features. Future work includes adding more herbal plants as samples to further validate the model constructed in this study and develop GUI that is able to provide information such as benefit, usage and method of preparation for these herbal plants. Once this information is available, a mobile application can be developed, and later released for public usage as well as cataloguing purposes.

#### ACKNOWLEDGEMENTS

The authors gratefully acknowledge the research grant provided by Institute of Research Management & Innovation (IRMI), Universiti Teknologi MARA (UiTM), 600-IRMI/ MyRA 5/3/LESTARI (060/2017) for sponsoring this research.

#### REFERENCES

- [1] Leng Guan, S., "Forest resources and ecosystem conservation in Malaysia," *Recursos Geneticos Forestales (FAO)*, 1992.
- [2] Milow, P., et al., "Medicinal Plants of the Indigenous Tribes in Peninsular Malaysia: Current and Future Perspectives." *Intech Open, Active Ingredients from Aromatic and Medicinal Plants*, March 2017
- [3] Onrizal, O., et al., "Diversity of plant community at Gunung Ledang, Malaysia," *IOP Conference Series: Earth and Environmental Science*, vol. 260, no. 1, p. 012068, May 2019.
- [4] Belhumeur, P. N., et al., "Searching the world's herbaria: A system for visual identification of plant species," *European Conference on Computer Vision*, Springer, Berlin, Heidelberg, pp. 116-129, Oct 2008.
- [5] Joly, A., et al., "Interactive plant identification based on social image data," *Ecological Informatics*, vol. 23, pp. 22-34, Sep 2014.
- [6] Nesbitt, M., et al., "Linking biodiversity, food and nutrition: The importance of plant identification and nomenclature," *Journal of food composition and analysis*, vol. 23 no 6, pp. 486-498, Sep 2010.
- [7] Strezoski, G., et al., "Deep learning and support vector machine for effective plant identification," *In Proceedings of ICT Innovations 2015 Conference Web Proceedings*, pp. 221-233, 2017.
- [8] Bruno, O. M., et al., "Fractal dimension applied to plant identification" *Information Sciences*, vol 178 no 12, pp. 2722-2733, June 2008.
- [9] Kumar, N., et al., "Leafsnap: A computer vision system for automatic plant species identification," *European Conference on Computer Vision –ECCV 2012*, Springer, Berlin, Heidelberg, pp. 502-516, 2012.
- [10] Goëau, H., et al., "The imageCLEF plant identification task 2013" *Proceedings of the 2nd ACM international workshop on Multimedia analysis for ecological data*, ACM, pp. 23-28, Oct 2013.

- [11] Camargo, A., and Smith, J. S., "Image pattern classification for the identification of disease causing agents in plants," *Computers and Electronics in Agriculture*, vol. 66, no 2, pp. 121-125, 2009.
- [12] Sladojevic, S., et al., "Deep neural networks based recognition of plant diseases by leaf image classification," *Computational intelligence and neuroscience*, Juni 2016.
- [13] Wahid, A. et al., "Plant Disease Identification using Image Classification Techniques," *International Journal of Engineering Science*, vol. 8, no. 4, pp. 16795-16796, April 2018.
- [14] Ibrahim, Z., et al., "Palm Oil Fresh Fruit Bunch Ripeness Grading Recognition Using Convolutional Neural Network," *Journal of Telecommunication, Electronic and Computer Engineering JTEC*, vol. 10, no. 3-2, pp. 109-113, 2018.
- [15] C. Du and S. Gao, "Image Segmentation-Based Multi-Focus Image Fusion Through Multi-Scale Convolutional Neural Network," in *IEEE Access*, vol. 5, pp. 15750-15761, 2017.
- [16] Hu, J., et al., "A Multiscale Fusion Convolutional Neural Network for Plant Leaf Recognition," in *IEEE Signal Processing Letters*, vol. 25, no. 6, pp. 853-857, June 2018.
- [17] L. Xie and A. Yuille, "Genetic CNN," *2017 IEEE International Conference on Computer Vision ICCV*, Venice, pp. 1388-1397, 2017.
- [18] Yu, D., et al., "Mixed pooling for convolutional neural networks," *International Conference on Rough Sets and Knowledge Technology*, Springer, Cham, October, pp. 364-375, 2014.
- [19] Zheng, L., et al., "Good practice in CNN feature transfer," *arXiv preprint arXiv:1604.00133*, April 2016.
- [20] Xie, L., et al., "DisturbLabel: Regularizing CNN on the Loss Layer," *2016 IEEE Conference on Computer Vision and Pattern Recognition CVPR*, Las Vegas, NV, pp. 4753-4762, 2016.
- [21] Ding, Y., et al., "A deep learning model to predict a diagnosis of Alzheimer disease by using 18F-FDG PET of the brain" *Radiology*, vol. 2, pp. 456-464, Nov 2018.
- [22] Wong, S. C., et al., "Understanding Data Augmentation for Classification: When to Warp?," *2016 International Conference on Digital Image Computing: Techniques and Applications DICTA*, Gold Coast, QLD, pp. 1-6, 2016.
- [23] Wigington, C., et al., "Data Augmentation for Recognition of Handwritten Words and Lines Using a CNN-LSTM Network," *2017 14th IAPR International Conference on Document Analysis and Recognition ICDAR*, Kyoto, pp. 639-645, 2017.
- [24] Suzuki, H., et al., "A CNN handwritten character recognizer," *International journal of circuit theory and applications*, vol. 20, no. 5, pp. 601-612, Oct 1992.
- [25] Henriques, J. F., et al., "Warped convolutions: Efficient invariance to spatial transformations," *The 34th International Conference on Machine Learning*, vol. 70, pp. 1461-1469, 2017.