International Journal of Electronics and Communications System
Volume 2, Issue 2, 75-83.
ISSN: 2798-2610
http://ejournal.radenintan.ac.id/index.php/IJECS/index

Comparison Study of Convolutional Neural Network Architecture in Aglaonema Classification

Yessi Mulyani
University of Lampung, INDONESIA

Dzihan Septiangraini
University of Lampung, INDONESIA

Meizano Ardhi Muhammad*
University of Lampung, INDONESIA

Gigih Forda Nama
University of Lampung, INDONESIA

Article Info

Article history:

Received: September 5, 2022
Revised: November 25, 2022
Accepted: December 30, 2022

Keywords:

Aglaonema;
AlexNet;
CNN;
Inception V3;
LeNet;
ResNet50;
VGG16.

Abstract

Convolutional Neural Network (CNN) is very good at classifying images. To measure the best CNN architecture, a study must be done against real-case scenarios. Aglaonema, one of the plants with high similarity, is chosen as a test case to compare CNN architecture. In this study, a classification process was carried out on five classes of Aglaonema imagery by comparing five architectures from the Convolutional Neural Network (CNN) method: LeNet, AlexNet, VGG16, Inception V3, and ResNet50. The total dataset used is 500 image data, with the distribution of training data by 80% and test data by 20%. The segmentation process is performed using the Grabcut algorithm by separating the foreground and background. To build a model for CNN architecture using Google Colab and Google Drive storage. The results of the tests carried out on five classes of Aglaonema images obtained the best accuracy, precision, and recall results on the Inception V3 architecture with values of 92.8%, 93%, and 92.8%. The CNN architecture has the highest level of accuracy in classifying aglaonema plant types based on images. This study seeks to close research gaps, contribute to the field of research, and serve as a platform for primary prevention research.

To cite this article: Y. Mulyani, D. Septiangraini, M. A. Muhammad, and G. F. Nama, “Comparison Study of Convolutional Neural Network Architecture in Aglaonema Classification,” *Int. J. Electron. Commun. Syst.*, vol. 2, no. 2, pp. 75-83, 2022

INTRODUCTION

Information technology is increasing with many innovations in various fields. One of the fields that have developed rapidly is deep learning. Deep learning is one of the fields that can perform the classification process on pattern recognition[1], [2]. An additional method is required to facilitate the classification process, namely the Convolutional Neural Network (CNN) method[3]. The CNN method is commonly used in pattern recognition, such as image recognition[4].

In the CNN method, some architectural models can assist the pattern recognition process [5], [6]. CNN architectures that are well-known are LeNet, AlexNet, VGGNet, GoogleNet, and ResNet. Each CNN architecture has a fair degree of accuracy to perform the classification process [7]. Based on those architectures, the value and error rate need to be measured so it can produce good accuracy in the classification process.

Based on those architectures, the value and error rate need to be measured to produce good accuracy in the classification process. *Aglaonema* plants have several varieties spread worldwide with different characteristics of each type based on the related literature[8]. *Aglaonema* is a language that originates from Greece, if interpreted *Aglaonema* is a shining thread[9]. *Aglaonema* is a language originating from Greece. Suppose interpreted *Aglaonema* is a shining thread[9]. There are more than 100

- **Corresponding author:**
Meizano Ardhi Muhammad, University of Lampung, INDONESIA. ✉ meizano@eng.unila.ac.id
© 2022 The Author(s). **Open Access.** This article is under the CC BY SA license (https://creativecommons.org/licenses/by-sa/4.0/)

varieties of *Aglaonema* plants in Indonesia, and they are very popular with the public because of the beauty of their leaves.

The selection of these five types is due to their similar physical appearance. It is difficult for people to recognize because many laypeople find it challenging to identify the type of Aglaonema. Those difficulties are caused by *Aglaonema's* similarity in leaf structure, such as shape, color, and texture[10]. So, unscrupulous sellers sell cheap Aglaonema types with expensive Aglaonema names. *Aglaonema* plants have various varieties. Many ordinary people must be more accurate in distinguishing Aglaonema plants from Dieffenbachia plants. Moreover, the price of this Aglaonema is very varied. Of course, there will be losses if the seller or buyer incorrectly identifies the type of Aglaonema.

To make identifying and classifying Aglaonema plants easier, images are taken, which will then be processed or processed into data that will be classified using a Convolutional Neural Network (CNN). CNN is widely applied to data in images because it stores unique information from image data to produce good classification results [6]. In classifying aglaonema plant species based on leaf characteristics, the CNN method is expected to correctly carry out the classification process to reduce errors in recognizing aglaonema types.

Convolutional Neural Network (CNN) is the development of Multilayer Perceptron (MLP), which is designed to process two-dimensional data. CNN is included in the type of Deep Neural Network because of its high network depth and widely applied to image data. In the case of image classification, MLP is unsuitable for use because it does not store spatial information from image data and considers each pixel as an independent feature, resulting in poor results [11]. Average pooling returns the average of all values from the portion of the image covered by the kernel. Average pooling only reduces dimensions as a noise-reduction mechanism [12].

Numerous research trends exist about the architecture in aglaonema classification. The CNN method can identify aglaonema leaf images well. In addition, this study also proves that models with background train data have higher accuracy compared to models that do not have a background[13]. The K-Nearest Neighbor (KNN) method The K-Nearest Neighbor (KNN) method can be used to classify aglaonema plants based on leaf patterns with an identification accuracy rate of 75.55%[14]. The block-based architecture of CNN encourages learning in a modular fashion, thereby making architecture more straightforward and more understandable. A block concept is a structural unit that will persist and further enhance CNN performance[15].

However, only a few studies published in credible publications exploring the classification process were carried out on five classes of Aglaonema imagery by comparing five architectures from the Convolutional Neural Network (CNN). There is yet to be a free-from-classification process correctly to reduce errors in recognizing aglaonema types. Many ordinary people are mistaken in distinguishing Aglaonema plants from Dieffenbachia plants or better known as Blanceng plants, because of the many morphological similarities. So that in classifying aglaonema plant species based on leaf characteristics, the CNN method is expected to carry out the classification process correctly to reduce errors in recognizing aglaonema types.

In this study, we will offer thorough information on comparing the CNN architecture with the highest level of accuracy in classifying the types of *Aglaonema* plants based on images. This study seeks to cover research gaps, contribute to the research field, and serve as a platform for primary prevention research. This research also used 5 CNN architectures and five varieties. To create and collect the data sets, we use the architecture and test the performance of the models.

METHOD

In this research, the process of classifying images to identify the type of Aglaonema plant using the Convolution Neural Network (CNN) method involves several steps. As shown in Figure 1.

Dataset Preparation

The data needed for this research is a sample image of the Aglaonema plant that will be acquired. Image acquisition of the Aglaonema plant is carried out first to be used as a digital image. There are five image classes: *Aglaonema variety rhapsody in green*, *Aglaonema variety donnacarmen and Aglaonema costatum var. Albovariegatum*, *Aglaonema varietas siam aurora, and Aglaonema butterfly L*. Each class has 100 pictures; therefore, the total numbers of Aglaonema plant images are 500. All images are placed in local storage.

Features Selection

The steps to carry out the feature selection process are shown in Figure 2. The feature selection process uses Jupyter Notebook tools with the Python programming language. The first process is to import the libraries needed for the feature selection. Then, import the dataset onto the local storage from the laptop by entering the dataset file name. The next step is to select the main subject by marking it as a rectangle. The algorithm will then delete the object that is not defined as the main object. After the algorithm deletes the background, it can be seen if the main thing is ready to be used or if other entities are not meant, and the object can be removed manually by marking the object so that it can remove it. The process of removing or returning these items may be executed several times. If the image segmentation results are appropriate, they may be saved and used in training and testing.

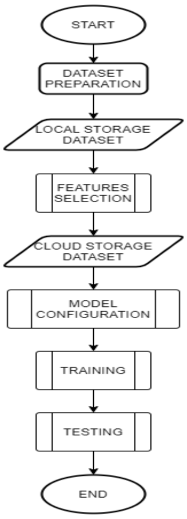
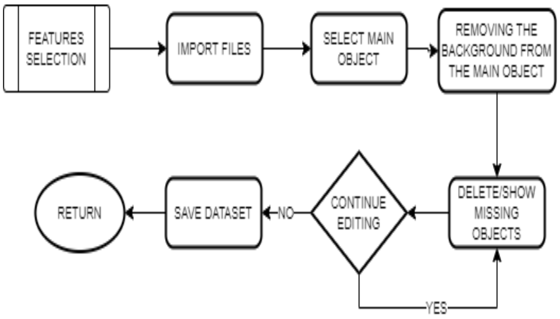


Figure 1. Research Stages



Figures 2. Features Selection Process

Model Configuration

Process The process configuration is shown in Figure 3.

The process used to build the training model is:

a. Import the necessary libraries to create a training model,
b. Connect Google Colab with Google Drive as dataset storage. Once Google Drive is connected, the data training can be labeled,
c. Divide the training data into 2 sections: 90% training data and 10% validation data, and
d. The steps for setting up models for the training process, each model having a different training setup. The LeNet architectural model has five layers for the training process, split into three convolution layers and two fully connected layers. The AlexNet architectural model has a training model setup with eight main layers, five convolution layers, and three fully connected layers. The VGG16 model has a model training set up with 16 main layers, divided into 13 convolution layers and three fully connected layers. The Inception V3 architecture model has a training setup comprising 105 main layers, divided into 102 convolution layers and three fully connected layers. Meanwhile, the ResNet50 model has a training setup with 50 main layers.

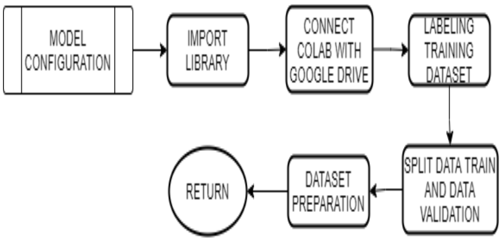

MODEL CONFIGURATION
IMPORT LIBRARY
CONNECT COLAB WITH GOOGLE DRIVE
LABELING TRAINING DATASET
SPLIT DATA TRAIN AND DATA VALIDATION
DATASET PREPARATION
RETURN

Figure 3. Model Configuration

Training

The process training model is shown in figure 4. The training process is carried out by setting up an architectural model that has been made, and then the training process is executed for 50 epochs. Based on the training results, the loss, accuracy, val_loss, and val_accuracy values can be seen, which can then be seen as the time required for the model during the training process. Once the training process is complete, the training model can be saved to be used for the testing process. In 5 architectural models, the training process will be performed in 5 iterations, taking the average value of each architectural model.

Testing

The process for the testing model is shown in Figure 5. The testing process for the specification Graphics Processing Unit (GPU) uses NVIDIA Tesla T4. After the training process is completed, the model weights are stored, and then the model weights that were stored are recalled to complete the prediction process on the test data set. After importing the model weights, calls are made to the test dataset in cloud storage on Google Drive. Then make a prediction model to test the results of the training model that has been done. After making a prediction model, the next step is to predict 100 test datasets without being labeled.

The results of predictions on 100 test datasets will be stored and used to measure the performance of the CNN architectural model using the confusion n matrix to get the values of True Positive, True Negative, False Positive, and False Negative. This value will produce a percentage for each architectural model's accuracy, precision, and recall.

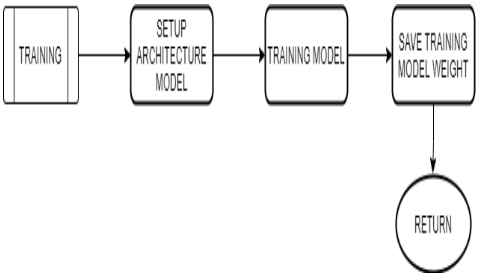


Figure 4. Training Process

RESULTS AND DISCUSSION

The results of the CNN model architectural comparison process are presented here.

Features Selection

In the segmentation process, separate the main object of the leaf from the background by using the grab-cut algorithm[16], [17]. After selecting the main thing, the task of the grab cut algorithm is to eliminate objects that are not selected by marking them as background. When objects are not needed, the object removal process may be performed[18], [19]. If a part of the main object is missing, you can return the thing by placing a marker on the missing piece. This process can be performed multiple times.

Model Configuration

The process of model configuration used google collab and google drive storage. Each architectural model uses a different setup based on the number of layers used. LeNet is one of the architectures for approximating the first CNN method introduced in 1998. This architecture was initially designed to perform digit classification for 28x28 grayscale images. The most popular version of LeNet, LeNet-5, contains two convolution layers followed by two fully-connected layers[20].

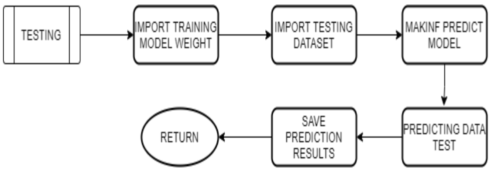


Figure 5. Testing Process

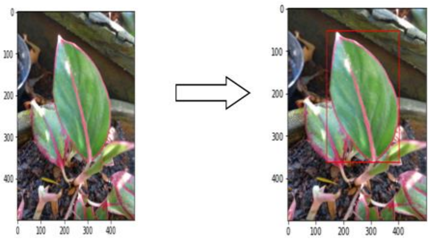


Figure 6. Select the Main Object

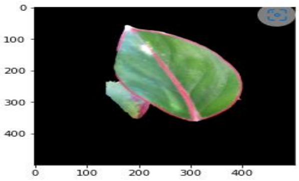


Figure 7. The Result after Removing the Background

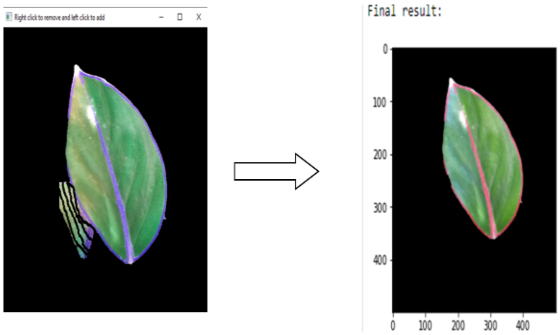
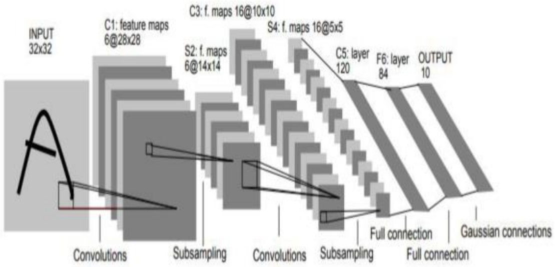


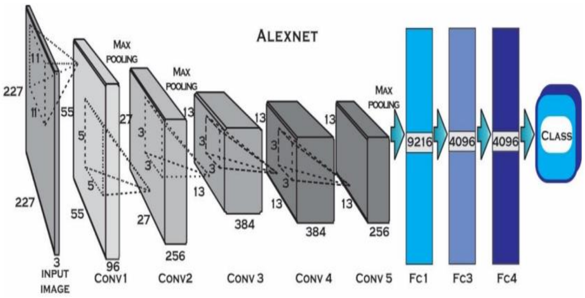
Figure 8. Remove or Return the Object



Figures 9. Architecture LeNet [7]

AlexNet was the first CNN method to win the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012 and made the CNN method better known. This architecture has an input image of 227x227 and consists of five convolution layers followed by three fully-connected layers. Each convolution layer has 96 to 384 filters and filter sizes from 3x3 to 11x11, with 3 to 256 channels each [20].

The VGG16 model achieves nearly 92.7% top-5 test accuracy on ImageNet. ImageNet consists of more than 14 million images from almost 1000 classes. In addition, one of the most popular models was submitted to ILSVRC-2014. It replaces the sizeable kernel-sized filter with multiple 3x3 kernel-sized filters one after the other, thus making a significant improvement over AlexNet. The VGG16 has an image input of 224x224.



Figures 10. Architecture AlexNet [21]

InceptionV3 is a multi-layered architecture of up to 90 convolution layers and three fully-connected layers. The number of layers is because this architecture has a module that combines several convolution layers into one part in the concatenate layer, where this layer will combine the input values in the previous layer at a predetermined dimension. This module aims to avoid the loss of gradients in each layer during training to increase the accuracy value[20].

The ResNet50 architecture has 50 layers, including 49 convolution layers and one fully-connected layer. Each convolution layer has a variety of filters, namely 64, 128, 256, 512, 1024, and 2048. The kernel size in this architecture also varies, namely 1x1, 3x3, and 7x7 [20]. In this architecture, a shortcut module is also helpful in preventing the occurrence of weight values that will interfere with the next layer. When a weight value does not match the previous layer, the next layer will be skipped because it is considered to interfere with the weight search being carried out[2], [22].

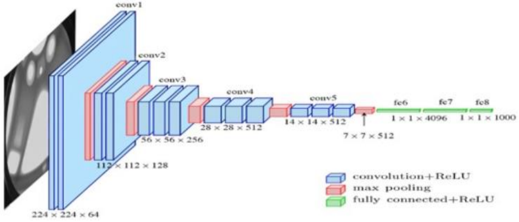


Figure 11. Architecture VGG16 [23]

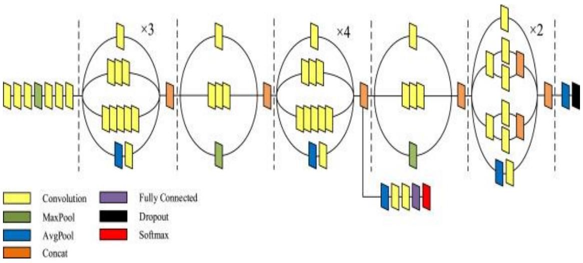
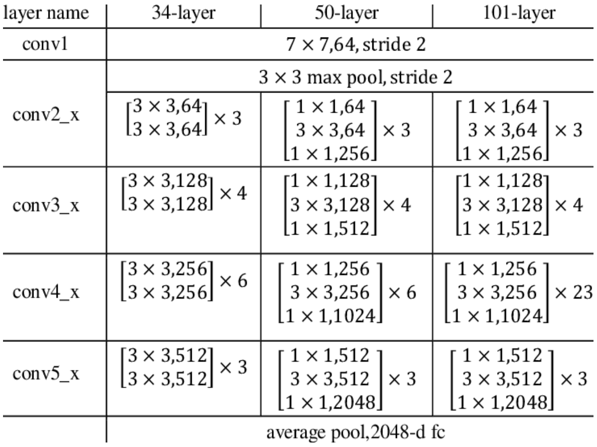


Figure 12. Architecture Inception V3 [24]

Training

The results of the LeNet training from five iterations have been performed, and the average results are shown in Table 1 and Table 2. The results of the VGG16 training from the five iterations that have been performed, the average results, are shown in Table 3. The effects of the Inception V3 training from the five iterations that have been performed. the intermediate results are shown in Table 4.



layer name	34-layer	50-layer	101-layer
conv1	7 × 7,64, stride 2		
	3 × 3 max pool, stride 2		
conv2_x	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$
conv3_x	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times4$
conv4_x	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times23$
conv5_x	$\begin{bmatrix}3\times3,512\\3\times3,512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$
	average pool, 2048-d fc		

Figure 13. Architecture ResNet50 [25]

Table 1. Architecture LeNet Training Result

i	loss	acc	V_loss	V_acc	time (s)
1	0.9054	0.6556	0.8688	0.6625	200
2	1.0689	0.5306	1.2340	0.5250	200
3	0.8747	0.6472	1.1322	0.5000	200
4	0.8848	0.6444	0.8529	0.6000	200
5	0.9867	0.5889	1.0831	0.5250	200
Avg	**0.9441**	**0.6133**	**1.0342**	**0.5625**	**200**

Table 2. Architecture AlexNet Training Result

i	loss	acc	V_loss	V_acc	time (s)
1	0.7672	0.7389	8.0780	0.1500	447
2	0.8748	0.6722	4.2888	0.2250	417
3	0.8387	0.7528	4.8215	0.5000	416
4	0.8100	0.7528	3.6535	0.4750	423
5	0.7901	0.7194	3.5136	0.3250	418
Avg	**0.8161**	**0.7272**	**4.8710**	**0.335**	**424**

Table 3. Architecture VGG16 Training Result

i	loss	acc	V_loss	V_acc	time (s)
1	0.0217	0.9917	0.2180	0.9500	485
2	0.0351	0.9944	0.5240	0.9500	417
3	0.0555	0.9861	0.3369	0.9250	401
4	0.1753	0.9361	0.2073	0.9250	413
5	0.0767	0.9639	0.2250	0.9250	401
Avg	**0.0728**	**0.9744**	**0.3022**	**0.935**	**423**

Table 4. Architecture Inception V3 Training Result

i	loss	acc	V_loss	V_acc	time (s)
1	0.0935	0.9917	0.2353	0.9500	676
2	0.4265	0.9667	0.0083	1.0000	560
3	0.4450	0.9694	2.3414	1.0000	559
4	0.2463	0.9667	1.0066	1.0000	560
5	0.1720	0.9806	0.0026	1.0000	560
Avg	**0.2766**	**0.9750**	**0.7188**	**0.9900**	**583**

The results of the ResNet50 training from the five iterations that have been performed, the average results, are shown in Table 5.

Testing

The test process is conducted using 100 segmented Aglaonema image data. This process is performed for five iterations and then determined based on the average value. Performance measurement of each architecture is calculated using a confusion matrix to get the accuracy, precision, and recall values. The results of the LeNet testing from the five iterations that have been performed, the average results, are shown in Table 6. The average weight of accuracy, precision, and recall of the five iterations performed is 56.8%, 59.86%, and 56.8%.

Table 5. Architecture ResNet50 Training Result

i	loss	acc	V_loss	V_acc	Time (s)
1	1.0658	0.6778	1.3826	0.5750	676
2	2.1550	0.6333	1.8607	0.6250	408
3	1.2730	0.7222	1.5127	0.5500	407
4	1.0869	0.7361	2.1234	0.5500	407
5	0.9755	0.7806	2.9214	0.4750	410
Avg	**1.3112**	**0.7100**	**1.9601**	**0.5550**	**461**

Table 6. Architecture LeNet Testing Result

Iteration	Accuracy (%)	Precision (%)	Recall (%)
1	57	61.4	57
2	50	60	50
3	61	59.2	61
4	59	61.8	59
5	57	56.9	57
Average	**56.8**	**59.86**	**56.8**

The results of the AlexNet testing from the five iterations that have been performed, the average results, are shown in Table 7. The average value of accuracy, precision, and recall of the five iterations performed is 27.4%, 41.6%, and 27.4%. The results of the VGG16 testing from the five iterations that have been performed, the average results, are shown in Table 8. The average value of accuracy, precision, and recall of the five iterations performed is 84%, 87.12%, and 84%.

Table 7. Architecture AlexNet Testing result

Iteration	Accuracy (%)	Precision (%)	Recall (%)
1	22	10.8	22
2	17	5	17
3	28	38.2	28
4	33	34.2	33
5	37	56.9	37
Average	**27.4**	**41.6**	**27.4**

Table 8. Architecture VGG16 Testing Result

Iteration	Accuracy (%)	Precision (%)	Recall (%)
1	76	83.2	76
2	82	87.4	82
3	95	95.6	95
4	81	83.2	81
5	86	86.2	86
Average	**84**	**87.12**	**84**

The results of the Inception V3 testing from the 5 iterations that have been performed, the average results are shown in Table 9. The average value of accuracy, precision, and recall of the 5 iterations performed is 92.8%, 93%, and 92.8%.

Table 9. Architecture Inception V3 Testing Result

Iteration	Accuracy (%)	Precision (%)	Recall (%)
1	92	92.2	92
2	91	91.2	91
3	96	96	96
4	91	91	91
5	94	94.6	94
Average	**92.8**	**93**	**92.8**

The results of the ResNet50 testing from the 5 iterations that have been performed, the average results are shown in Table 10. The average value of accuracy, precision, and recall of the 5 iterations performed is 57.8%, 68.82%, and 57.8%.

Table 10. The Architecture ResNet50 Testing Result

Iteration	Accuracy (%)	Precision (%)	Recall (%)
1	54	65.6	54
2	49	69.6	49
3	66	68	66
4	60	72.4	60
5	60	68.5	60
Average	**57.8**	**68.82**	**57.8**

The comparison of the 5 CNN architectures that have been tested is shown in table 11.

Table 11. The Comparison of CNN Architecture

Architecture CNN	Accuracy (%)	Precision (%)	Recall (%)	Time (s)	Classification (s)	Weight model
LeNet	56.8	59.86	56.8	200	0.5	824.94KB
AlexNet	27.4	25.96	27.4	424	1.06	283.83MB
VGG16	84	87.12	84	423	1.0575	512.3MB
InceptionV3	92.8	93	92.8	583	1.4575	342.96MB
ResNet50	57.8	68.82	57.8	461	1.1525	96.04MB

Table 11 depicts the results of calculating the performance of 5 CNN architecture models for LeNet, AlexNet, VGG16, Inception V3, and ResNet50. Architecture models with an accuracy value above 70% are Inception V3 and VGG16. In contrast, architectures with accuracy values below 70% are LeNet, AlexNet, and ResNet50. The architecture model with the fastest training time is LeNet for 200 seconds, while the model with the longest training time is Inception V3 for 583 seconds. The architecture model with the lowest model weight is LeNet of 824,94kB, and the architecture model with the most significant weight is VGG16 which is 512,3.

CONCLUSION

Based on the results of calculations using the fusion matrix of the five tested Convolutional Neural Network (CNN) architectures, the CNN architectures that have succeeded in carrying out the Aglaonema classification process are Inception V3 and VGG16 with accuracy values above 70%. To make an aglaonema classification application, the CNN architectural model that can be used is Inception V3, with an accuracy of 92.8% with a classification time needed under 2 seconds and a model weight of only 342.96MB. The CNN architecture has the highest level of accuracy in classifying aglaonema plant types based on images. This study seeks to close research gaps, contribute to the field of research, and serve as a platform for primary prevention research.

REFERENCES

[1] L. Alzubaidi *et al.*, *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*, vol. 8, no. 1. Springer International Publishing, 2021. doi: 10.1186/s40537-021-00444-8.

[2] A. Anton, N. F. Nissa, A. Janiati, N. Cahya, and P. Astuti, "Application of Deep Learning Using Convolutional Neural Network (CNN) Method For Women's Skin Classification," *Sci. J. Informatics*, vol. 8, no. 1, pp. 144–153, 2021, doi: 10.15294/sji.v8i1.26888.

[3] M. I. Mardiyah and T. Purwaningsih, "Developing deep learning architecture for image classification using convolutional neural network (CNN) algorithm in forest and field images," *Sci. Inf. Technol. Lett.*, vol. 1, no. 2, pp. 83–91, 2020, doi: 10.31763/sitech.v1i2.160.

[4] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition," *Smart Innov. Syst. Technol.*, vol. 195, no. 9, pp. 61–629, 2021, doi: 10.1007/978-981-15-7078-0_3.

[5] A. Biswas and M. S. Islam, "An Efficient CNN Model for Automated Digital Handwritten Digit Classification," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 7, no. 1, p. 42, 2021, doi: 10.20473/jisebi.7.1.42-55.

[6] O. I. Abiodun *et al.*, "Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition," *IEEE Access*, vol. 7, no. February 2017, pp. 158820–158846, 2019, doi: 10.1109/ACCESS.2019.2945545.

[7] D. Seo, J. Ahn, and H. Nam, "Comparison of CNN Architectures using RP Algorithm for Burst Signal Detection," *Int. Conf. ICT Converg.*, vol. 2020-Octob, pp. 809–812, 2020, doi: 10.1109/ICTC49870.2020.9289320.

[8] "No Title," 2019.

[9] "Buku Budidaya Tanaman Hias Daun Athurium dan Aglonema.pdf."

[10] C. Of *et al.*, "Karakterisasi tanaman aglaonema di dataran tinggi rejang lebong (," vol. 17, no. 2, pp. 141–151, 2019.

[11] W. S. Eka Putra, "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101," *J. Tek. ITS*, vol. 5, no. 1, 2016, doi: 10.12962/j23373539.v5i1.15696.

[12] I. A. Sabilla, "Arsitektur Convolutional Neural Network (Cnn) Untuk Klasifikasi Jenis Dan Kesegaran Buah Pada Neraca Buah," *Tesis*, no. 201510370311144, pp. 1–119, 2020.

[13] S. Muhammad and A. T. Wibowo, "Klasifikasi Tanaman Aglaonema Berdasarkan Citra Daun Menggunakan Metode Convolutional Neural Network (Cnn," *e-Proceeding Eng.*, vol. 8, no. 5, pp. 10621–10636, 2021.

[14] H.Taufiq, K. Patmi, and S. Ardi, "Classification of Aglaonema Plants Berdasarkan Corak Daun," *Semin. Nas.*

Inov. Teknol., pp. 223–228, 2019, [Online]. Available: https://proceeding.unpkediri.ac.id/index.php/inotek/article/view/541

[15] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, *A survey of the recent architectures of deep convolutional neural networks*, vol. 53, no. 8. Springer Netherlands, 2020. doi: 10.1007/s10462-020-09825-6.

[16] N. Anantrasirichai, S. Hannuna, and N. Canagarajah, "Automatic Leaf Extraction from Outdoor Images," no. September, 2017, [Online]. Available: http://arxiv.org/abs/1709.06437

[17] H. Wu, Y. Liu, X. Xu, and Y. Gao, "Object Detection Based on the GrabCut Method for Automatic Mask Generation," *Micromachines*, vol. 13, no. 12, 2022, doi: 10.3390/mi13122095.

[18] L. Zhang and M. Chang, "An image inpainting method for object removal based on difference degree constraint," *Multimed. Tools Appl.*, vol. 80, no. 3, pp. 4607–4626, 2021, doi: 10.1007/s11042-020-09835-0.

[19] S. Kulkarni, "Removal of Unwanted Objects From Images Using Statistics," *ICTACT J. Image Video Process.*, vol. 9, no. 2, pp. 1887–1893, 2018, doi: 10.21917/ijivp.2018.0268.

[20] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient Processing of Deep Neural Networks," *Synth. Lect. Comput. Archit.*, vol. 15, no. 2, pp. 1–341, 2020, doi: 10.2200/s01004ed1v01y202004cac050.

[21] D. Irfansyah, M. Mustikasari, and A. Suroso, "Arsitektur Convolutional Neural Network (CNN) Alexnet Untuk Klasifikasi Hama Pada Citra Daun Tanaman Kopi," *J. Inform. J. Pengemb. IT*, vol. 6, no. 2, pp. 87–92, 2021.

[22] S. Mariyam, A. Osman, and C. Ramadhe, "Weight Changes for Learning Mechanisms in Two-Term Back-Propagation Network," *Artif. Neural Networks - Archit. Appl.*, 2013, doi: 10.5772/51776.

[23] A. H. Sianturi, "Implementasi Algoritma Convolutional Neural Networks di Microsoft Azure untuk Mendeteksi Jenis Kebutaan Mata yang dialami Penderita Penyakit Diabetes," *Anal. Kesadahan Total dan Alkalinitas pada Air Bersih Sumur Bor dengan Metod. Titrim. di PT Sucofindo Drh. Provinsi Sumatera Utara*, pp. 44–48, 2018.

[24] A. Salma, A. Bustamam, and D. Sarwinda, "Diabetic Retinopathy Detection Using GoogleNet Architecture of Convolutional Neural Network Through Fundus Images," vol. 2021, pp. 1–6, 2021, doi: 10.11594/nstp.2021.0701.

[25] Z. Niswati, R. Hardatin, M. N. Muslimah, and S. N. Hasanah, "Perbandingan Arsitektur ResNet50 dan ResNet101 dalam Klasifikasi Kanker Serviks pada Citra Pap Smear," *Fakt. Exacta*, vol. 14, no. 3, p. 160, 2021, doi: 10.30998/faktorexacta.v14i3.10010.