TELKOMNIKA Telecommunication Computing Electronics and Control
Vol. 20, No. 6, December 2022, pp. 1264~1275
ISSN: 1693-6930, DOI: 10.12928/TELKOMNIKA.v20i6.21783

A comparative study of mango fruit pest and disease recognition

Kusrini[1], Suputa[2], Arief Setyanto[1], I Made Artha Agastya[1], Herlambang Priantoro[3], Sofyan Pariyasto[1]
[1]Magister of Informatics Engineering, Universitas AMIKOM Yogyakarta, Yogyakarta, Indonesia
[2]Department of Plant Protection, Faculty of Agriculture, Universitas Gadjah Mada, Yogyakarta, Indonesia
[3]PT Bank Mandiri, Jakarta, Indonesia

Article Info

Article history:

Received Sep 17, 2021
Revised Sep 12, 2022
Accepted Sep 22, 2022

Keywords:

CNN
InceptionResNet-V2
Inception-V3
Mango pest and disease
ResNet50
VGG16

ABSTRACT

Mango is a popular fruit for local consumption and export commodity. Currently, Indonesian mango export at 37.8 M accounted for 0.115% of world consumption. Pest and disease are the common enemies of mango that degrade the quality of mango yield. Specialized treatment in export destinations such as gamma-ray in Australia, or hot water treatment in Korea, demands pest-free and high-quality products. Artificial intelligence helps to improve mango pest and disease control. This paper compares the deep learning model on mango fruit pests and disease recognition. This research compares Visual Geometry Group 16 (VGG16), residual neural network 50 (ResNet50), InceptionResNet-V2, Inception-V3, and DenseNet architectures to identify pests and diseases on mango fruit. We implement transfer learning, adopt all pre-trained weight parameters from all those architectures, and replace the final layer to adjust the output. All the architectures are re-train and validated using our dataset. The tropical mango dataset is collected and labeled by a subject matter expert. The VGG16 model achieves the top validation and testing accuracy at 89% and 90%, respectively. VGG16 is the shallowest model, with 16 layers; therefore, the model was the smallest size. The testing time is superior to the rest of the experiment at 2 seconds for 130 testing images.

Corresponding Author:

Kusrini
Magister of Informatics Engineering, Universitas AMIKOM Yogyakarta
Jl. Ringroad Utara Condong Catur Depok Sleman, Yogyakarta, Indonesia
Email: kusrini@amikom.ac.id

1. INTRODUCTION

Mango is a potential commodity for local consumption and export. Indonesian coastal areas with high sun exposure all year round are suitable for mango. The regional consumption is 0.5 kg per capita per year, while exports account for 0.11% of world mango consumption. The international demand is high; however, due to the importer country requirements, Indonesian mango is challenging to enter the global market. Pest and disease-free requirements in Japan, Australia, and the Korean market hinder Indonesian mango from accessing their market. Therefore, pest and disease control to ensure the fruit product's quality plays a significant role in improving international market acceptance.

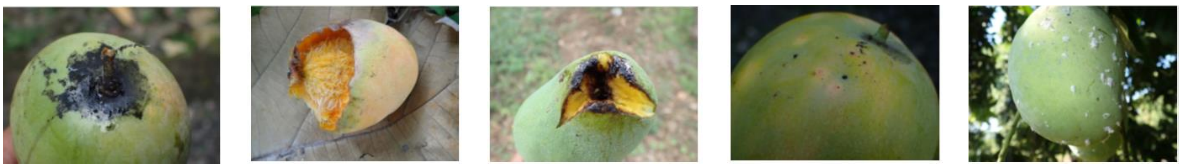
The ability of farmers to identify pests and diseases and proper handling is a significant factor. There are two mango farm models: the big professional farm and household mango trees. On big farms such as in east java and west java, experienced farmers manage large-scale areas. While in the household mango, some people grow a few trees around their house. Skilled people carry out pest and disease control on a big farm with sufficient knowledge. However, it is unavailable for general people with few mango trees around their houses. An innovative way to disseminate pests and diseases control techniques is desirable to overcome the knowledge gap.

Mobile application and image recognition are an opportunity to alleviate the pest and disease control dissemination knowledge problem. The penetration of smartphones is currently at about 56% of Indonesian citizens. Therefore, it is feasible to deliver knowledge through the smartphone. Image recognition has matured to detect many visual clues, including the pest and diseases on leaves and fruits. With the help of the mobile application, image recognition has been implemented in many recognition work for pests and diseases of plants such as [1], [2].

Deep learning has enjoyed tremendous success in classification tasks, particularly for image data. The availability of huge labeled datasets such as ImageNet [3] enables researchers to propose, test and validate many convolutional neural network (CNN)-based architectures. The transfer learning concept enables researchers to use knowledge learned by other problem sets (datasets) to be implemented in their specific problems with smaller dataset sizes [4]. In transfer learning concepts, we can use the weight parameter. To adjust the network to the new problems, fine-tuning was carried out. The adjustment can be applied to the entire network of only selected layers. Identify the best performance deep learning architectures, selecting which part of the layer needs tuning/adjusting. Currently, researchers proposed many well-known deep learning architectures, to name a few: AlexNet [5], AlexNetOWTBn [6], GoogLeNet [7], Overfeat [8], visual geometry group network (VGGnet) [9], residual neural network (ResNet) [10], InceptionResnet-V2 [11], Inception-V1 [7], Inception-V2 [12], Inception-V3 [12], and Inception-V4 [11], and DenseNet [13]. They all have different network structures, number of layers, size of filters and many other differences. Those lead to different weight parameters ranging from thousands to hundreds of millions. Consequently, they have different computational complexity, training and testing time.

This research is carried out to extend our previous work on recognizing mango pests on leaves. The recognition system for pests on mango leaves has been implemented on a mobile application [14]. Following up on suggestions on evaluation results [15], we improve the capability as an extension to the fruit. This research collects the mango fruit dataset and involves pests and diseases expert to manually classify the images into five classes. The appearance of pests and diseases of mango fruit in each class is shown in Figure 1(a) to Figure 1(e).

With the dataset size on hand we expect to be able to recognize the image collected in the real farm through the mobile application. Accurate and high speed recognition is desired to serve the mobile application. We aim to seek the most acceptable performance deep learning architecture with high accuracy and fast recognition. Therefore, we compare available architectures in transfer learning mode and compare their speed and accuracy.



(a) (b) (c) (d) (e)

Figure 1. Type of mango fruit pest and disease: (a) *Capnodium mangiferae*, (b) *Cynopterus titthaecheilus*, (c) *Deanolis albizonalis*, (d) *Pseudaulacaspis cockerelli*, and (e) *Pseudococcus longispinus*

2. RELATED WORKS

Plant pest and disease recognition based on visual data has attracted computer vision researchers in the past five decades. Researchers employ a support vector machine (SVM) to classify pomegranate fruit images into four classes consisting of normal and three infection stages [16]. Support vector machine classifies the pomegranate images using some features, which are color coherence vector (CCV), color histogram, and shape. They achieve 82% accuracy on the testing dataset. A classification of 82 disease classes on 12 plants has been reported by Barbedo *et al.* [17]. Evaluation of each plant has been carried out independently. The input image was segmented using guided active contour (GAC). Histogram similarity to the reference image is ranked as the basis of disease recognition. In [18], the image is segmented into the uninfected and infected regions. Researchers rely on the hue and co-occurrence matrix of infected leaf images to extract the features. Their work achieves 95.71% accuracy using SVM to predict the type of leaf disease. In [19], a comparison between a sparse representation-based classification (SRC), SVM, and artificial neural network (ANN) is carried out to classify cucumber leaf disease. The SRC outperforms SVM and ANN at 85.7% accuracy. A combination of dictionary learning and spare representation has been reported 92.9% accuracy on Caltex Leaves dataset [20]. Tan *et al.* [21] employ SVM to recognize a particular cacao fruit disease called cacao black pod rot (BPR) using k-means clustering and SVM. Their experiments reported 84% accuracy in recognizing BPR.

In the last decade, deep learning gained popularity and showed a breakthrough performance in image classification. Computer vision researchers made use of deep learning algorithms in recognizing plant pests and diseases such as in [22], [23] to recognize various leaf diseases [24] in a controlled condition. The promising result has been achieved at 96.3% of precision and 99.35% of accuracy. CNN was utilized by Krizhevsky *et al.* [5]. The transfer learning concept allows researchers to adopt a model trained by a huge dataset like image-net and adapt to particular cases with a smaller number of data. Deng *et al.* [25] implemented transfer learning and carried out fine-tuning to achieve high-performance classifier. Lu *et al.* [26] classified rice leaf disease using CNN and reported testing accuracy at 95.48%. They also identified that the stochastic pooling layer gave the best results after evaluating three different pooling layers. Wheat is an important food source. Therefore, in 2017 researchers collected seven classes of wheat disease in the wild called wheat disease database 2017 (WDD2017). In [27], they reported CNN-based networks with no fully connected layer (FCL) layers have been superior compared to the original CNN in classifying the WDD2017. Too *et al.* [28] reported a comparison of some well-known CNN architectures in classifying the PlantVillage dataset [24]. They evaluated VGGnet [9], Inception V4 [11], DenseNet [13], and ResNet [10]. According to their experiments, the DenseNet outperforms the rest of the architectures at 99.75% of accuracy. Researchers introduced a mobile-based wheat leaf disease recognition at [2]. They used ResNet architecture with 50 layers to carry out a classification task, and it showed promising classification performance at 96% of accuracy. Ferentinos [29] has put their effort in classifying leaf disease problems using five CNN architectures which are AlexNet [5], AlexNetOWTBn [6], GoogLeNet [7], Overfeat [8], VGGnet [9]. They reported that VGGnet outperforms the rest of the architectures, and it reaches 99.48% accuracy. A study of key factors impacting deep learning performance has been reported in [30]. They found that image background, image capture conditions, symptom representation, covariate shift, symptom segmentation, symptom variations, simultaneous disease, and symptom similarity are impacting factors to the deep learning performance. In [31], independent processing to each color channel input was introduced. The result was combined as an input of FCL. They evaluated the three channel CNN, GoogleNet, and LeNet-5 [32] to classify cucumber leaf disease and found that the three channels CNN achieved the best accuracy at 91.15%. In [33], the apple trunk disease recognition was carried out using VGGnet. They compared a VGGNet with Focal loss and softmax loss function. The VGGnet using focal loss function better performance with 2% margin at 94.5% accuracy compared VGGnet with softmax loss function. In [34], VGGnet was used to recognize mildew diseases and reach 95% of accuracy. Barbedo [35] reported that a classification task of 14 leaf diseases attain 94% of accuracy on implementation of GoogleNet architecture.

Despite the popularity of mango, there are a limited number of studies on mango pest and diseases recognition. The author reported 48.95% of accuracy on a recognition task of four diseases and a normal leaf using SVM [36]. They extract several features from the gray-level co-occurrence matrix (GLCM) matrix such as contrast, correlation, energy, homogeneity, mean, standard-deviation, entropy, root mean square (RMS), variance, smoothness, kurtosis, and skewness. Singh *et al.* [37] used CNN to recognize anthracnose disease and reported 97.13% accuracy. The obvious visual cue was responsible for the high achievement of this task. In our previous work, we classified mango pests [1] on affected leaf images. We collected the dataset [38] from mango farms in Indonesia and organized them into sixteen classes. We implemented augmentation techniques such as noise addition, blur, contrast, and affine transformation (i.e., rotation and translation in Cartesian coordinate) in order to improve the performance of VGGnet classifier. According to our experiments, augmentation successfully improved the accuracy by a 4% margin after using augmented images in the training phase.

3. RESEARCH METHOD

This is interdisciplinary research that involves mango pest expert and computer scientists. The dataset was collected around Indonesia. The image collection is labelled by mango pest expert. Once the dataset is labeled, data size standardization is carried out.

3.1. Dataset

The dataset consists of 653 labeled mango fruit images with five pests and a disease identified. In the real case, usually only one pest on particular mango fruit as reflected in the dataset each fruit images have a single pest label. The dataset is divided into training, testing, and validation set at 60%, 20%, and 20% respectively as presented in Table 1.

Table 1. Number of images in dataset

Pest/disease name	Train	Valid	Test	Total
capnodium_mangiferae	38	13	13	64
cynopterus_titthaecheilus	63	21	21	105
deanolis_albizonalis	86	28	28	142
pseudaulacaspis_cockerelli	159	52	52	263
pseudococcus_longispinus	47	16	16	79
Total	393	130	130	653

3.2. Deep learning image classifier

We implement a convolutional neural network using five well-known architectures. Their names are VGG16, RestNet50, InceptionResNet-V2, Inception-V3, and DenseNet. They will be discussed in following sub sections.

3.2.1. VGG16 model

We apply a CNN architecture named VGG16 which was used to win Imagenet competition in 2014. Figure 2 presents the detail architectures. This research adopts transfer learning methods as weight initialization. The VGG16 network has already been trained upon ImageNet dataset. So the initial weights of our network are duplicates of the ImageNet pre-trained model. A replacement of the final layers is carried out in the original VGG16 architecture. All the layer is freezed to retain the trained weight from the ImageNet, while the training set is performed to train the last replaced layer only. By limiting the weight for the last layer of the network, we can speed up the training time without sacrificing the classifier accuracy. Consequently, we only train the last layer, which is the fully connected layer (FCL) with a softmax function.

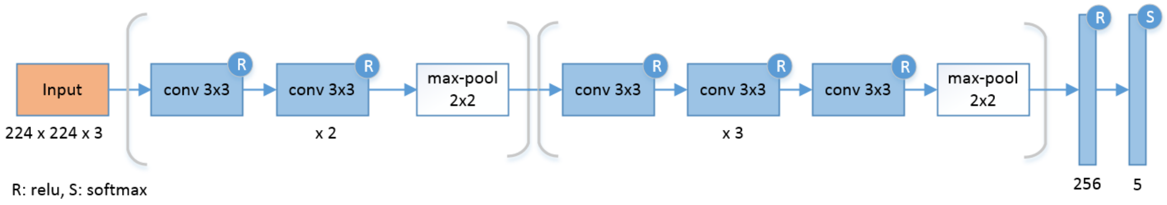


Figure 2. VGG16 architectures

The VGG16 is a convolutional neural network with 16 layers, and it is quite heavy computing complexity in the training process. However, the VGG-16's trained model execution is fast on the personal computer because CNN is a parameterized learning. By multiplying the saved weights again with a new sample, the model can predict its class. The training process is executed in a dedicated deep learning server. Therefore, computational load is not a significant matter. This research develops a high-accuracy model to serve client applications that recognize the pest from fruit images.

3.2.2. ResNet50 model

He *et al.* [10] introduced the ResNet model in their publication, which served as the basis for The ImageNet large-scale visual recognition challenge (ILSVRC) 2015 and Microsoft Common Objects in Context (COCO) 2015 classification challenges. Their model was ranked first in ImageNet classification with an error rate of 3.57%. Multiple non-linear layers' failure to learn identity mappings and the degradation problem spurred the development of the deep ResNet.

ResNet is a network-in-network (NIN) architecture that is built on a foundation of numerous stacked residual units. These leftover units are the network's building blocks. A collection of residual units serves as the foundation for the ResNet architecture [10]. Convolution, pooling, and layering are used to create the residual units. The architecture is comparable to that of the VGG network [9], which consists of 33 filters, although ResNet is approximately eight times deeper. This is because global average pooling is used instead of fully connected layers. ResNet was further updated to improve accuracy by changing the residual module to use identity mappings. As in [10], a ResNet model of 50, 101, and 152 layers were built and loaded with pre-trained weights from ImageNet. Finally, a bespoke softmax layer was constructed for the purpose of identifying plant diseases. We use ResNet50 in this research. The architecture is shown in Figure 3.

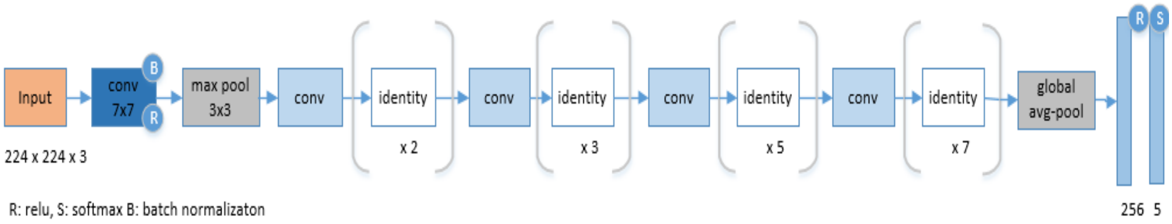


Figure 3. ResNet50 architecture

3.2.3. InceptionResNet-V2 model

InceptionResnet-V2 [11] is a CNN architecture that is a development of InceptionResnet-V1. InceptionResnet-V2 is used for the transfer learning classification process and is built based on the inception architecture by combining residual connections. InceptionResnet-V2 was developed by replacing the filter part of the Inception process [39], [40].

In Figure 4, there are several stages of classification to detect mango images. The input information is an image that will be converted into a frame to apply image processing techniques. Then the mango image is detected to produce important features that are in accordance with the characteristics and special characteristics of the mango fruit [41]. Inception Resnet-V2 in detecting mangoes is used as a high-level feature extractor that provides image content that can help identify pests on mango fruit [42].

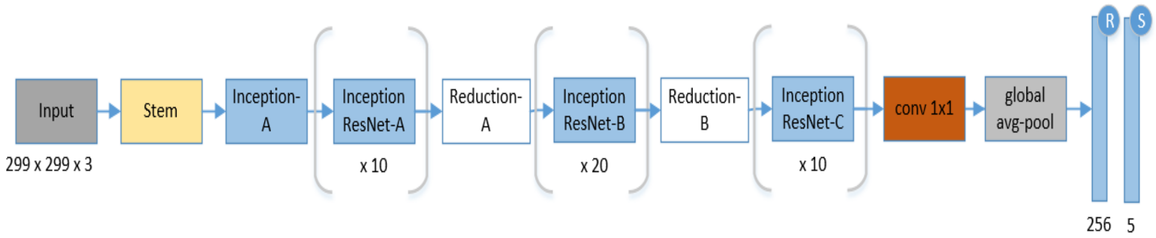


Figure 4. InceptionResNet-V2 architecture

3.2.4. Inception-V3 model

Inception has four versions, namely Inception-V1 [7], Inception-V2 [12], Inception-V3 [12], and Inception-V4 [11]. The inception model uses several filters on the usual layers. The results of several filters are combined using a concatenated channel before entering the next iteration [43]. There are 48 Layers in Inception-V3, which is deeper than its predecessor deep convolutional neural network architecture named Inception-V1 or GoogLeNet [7].

Inception V3 network structure uses the convolution kernel splitting method to split large volume integrals into small convolutions. For example, the convolution 3×3 is divided into convolutions 3×1 and 1×3. Through the separation method, the number of parameters can be reduced; hence, network training speed can be accelerated while spatial features can be extracted more effectively [44].

This study uses one of the deep learning neural network models, the Inception-V3 model used in TensorFlow to extract and classify mango fruit image features [45], [46]. The Inception-V3 model was used in TensorFlow to develop an image classifier to classify images of pests on mangoes based on three features: texture, shape, and color. The architecture is shown in Figure 5.

3.2.5. DenseNet model

In their paper, Huang *et al.* [13] introduced a densely connected convolutional network architecture. To ensure maximum information flow between layers in the network, all layers are connected directly with each other in a feed-forward manner. For each layer, the feature-maps of all preceding layers are used as inputs, and its own feature maps are used as inputs into all subsequent layers. DenseNets alleviate the problem of the vanishing-gradient problem and has substantially reduced number of parameters [13]. For this task of plant disease identification, DenseNets model with121 layers as described in [13] was created. Additionally, the model was loaded with pre-trained weights from ImageNet. Finally, another fully-connected model with our own customized softmax on the top layer was created.

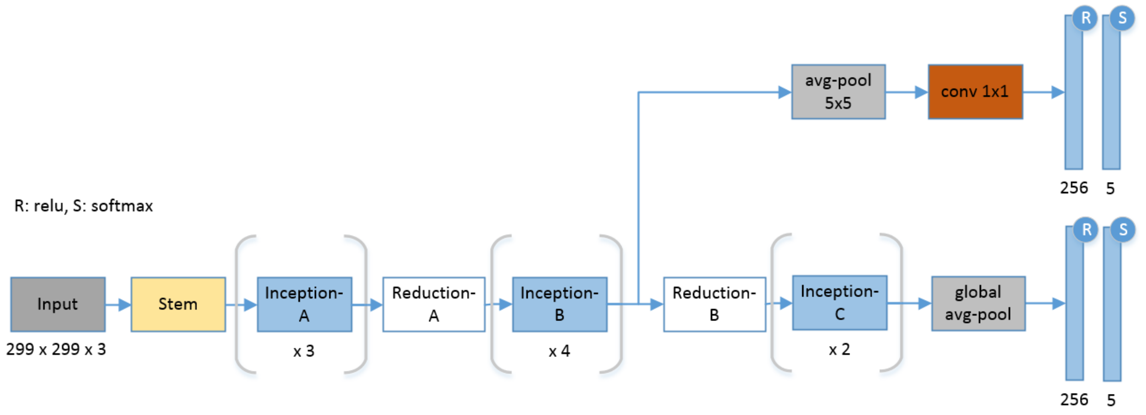


Figure 5. Inception-V3 architecture

3.3. Fine tuning

This research adopts the transfer learning concept, where all the architecture above has been pre-trained using imageNet dataset for 1000 target classes. Transfer learning aims to use the knowledge gained during training in one type of problem is used to train in another related task or domain [4]. In order to adjust the network to fit our classification problem, the head of the network is replaced so that the number of target class are five classes.

Fine-tuning is a concept of transfer learning. In our research, fine-tuning is carried out to the entire network in order to re-train all the weight parameters. Fine-tuned learning is started from the initial condition where the weight parameters are already trained on other problems. With new training set, it is need an adjustment for all the layers or particular layers. The researcher can select which layer needs to be re-trained and freeze other particular layers. Even though the training is needed for adjusting the new problem set, the initial knowledge can significantly cut the learning effort compared to training from scratch [23]. More importantly, in manual cases it is more accurate compared to models trained from scratch.

In this research, the CNN models were fine-tuned to identify and classify five categories of fruit disease with pre-trained models on ImageNet dataset. ImageNet dataset is a huge collection of 1.2 Million labeled images in 1000 categories. The CNN architectures with the new head are re-train with a small number of mango fruit images.

4. RESULTS AND DISCUSSION

The research aims to identify the deep learning architecture with acceptable performance and high accuracy. Testing time and testing accuracy are two main considerations in the problem sets as the algorithms are designed to serve mobile client applications with multiple requests concurrently. Training time is important, but it was not the main consideration, since the training will only take place in the modelling task. It is worth mentioning that the server in these experiments is the prototype of the server that we use to serve the running pest visual recognition mobile application.

4.1. Experiment setup and parameters

The experiment in this research is conducted using computer server with specification:

– Processor: i9 9900K.
– Memory: 64 Gb.
– GPU: NVIDIA TITAN V, Memory 12 Gb, Tensor Cores 640, CUDA Cores 5120.

This research did not optimize the parameter. The only parameters set is the learning rate at 0.00005. The rest of the parameters is set at default value. Optimization algorithm is using “Adam”.

4.2. Evaluation metrics

The accuracy and loss for training and validation set were recorded. Training time and validation time for the entire training and validation set were also become this research focus. Finally, the testing accuracy and average testing time among the algorithm was recorded to consider the most feasible model to be implemented in the implementation. The execution time might not repeatable to get the exact similar number in other research due to different experiment settings, however, the comparison between deep learning models might show a similar trend.

4.2.1. VGG16
In VGG16, the training loss decreased until 50 epochs. However, the validation loss starts to fluctuate in epoch 5. It shows us that model start to over fit with the data training. The minimum accuracy and validation loss are 0.0458 and 0.3108, respectively. The validation accuracy starts to be stagnant in epoch 5. The maximum training and validation accuracy are 0.9821 and 0.90769, respectively. The accuracy and loss graphic is shown in Figure 6.

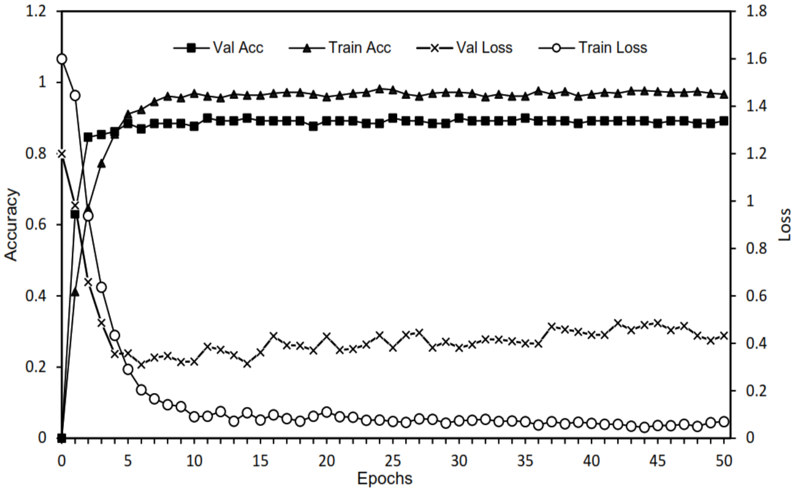


Figure 6. VGG16 training and validation accuracy and loss

4.2.2. ResNet50
In Resnet50, the training loss decrease until 50 epochs. However, the validation loss starts to increase in epoch 8. It shows us that model start to over fit with the data training. The minimum accuracy and validation loss are 0.0422 and 0.3885, respectively. The validation accuracy starts to be stagnant in epoch 12. The maximum training and validation accuracy are 0.9821 and 0.8923, respectively. The accuracy and loss graphic is shown in Figure 7.

4.2.3. InceptionResNet-V2
In InceptionResNet-V2, the training loss decrease until 50 epochs. However, the validation loss starts to increase in epoch 20. It shows us that model start to over fit with the data training. The minimum accuracy and validation loss are 0.0537 and 0.4207, respectively. The validation accuracy starts to be stagnant in epoch 9. The maximum training and validation accuracy are 0.9796 and 0.8846, respectively. The accuracy and loss graphic is shown in Figure 8.

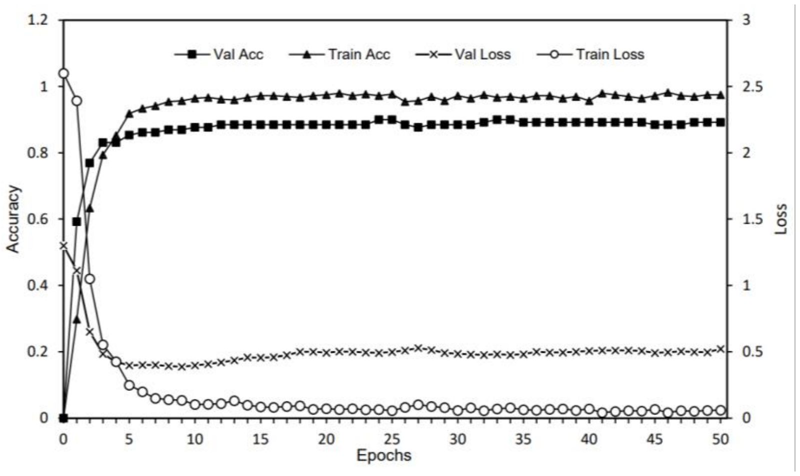


Figure 7. RestNet50 training and validation accuracy and loss

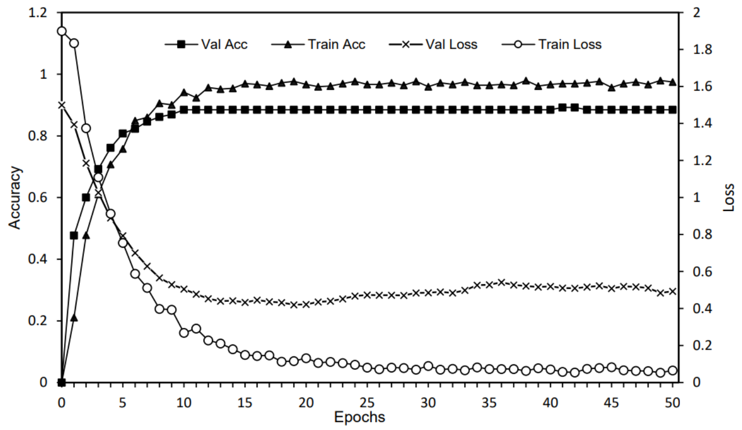


Figure 8. InceptionResNet-V2 training and validation accuracy and loss

4.2.4. Inception-V3

In Inception-V3, the training loss decrease until 50 epochs. However, the validation loss starts to be flat in epoch 12. It shows us that model start to over fit with the data training. The minimum accuracy and validation loss are 0.0531and 0.5618, respectively. The validation accuracy starts to be stagnant in epoch 10. The maximum training and validation accuracy are 0.9821 and 0.8692, respectively. The accuracy and loss graphic is shown in Figure 9.

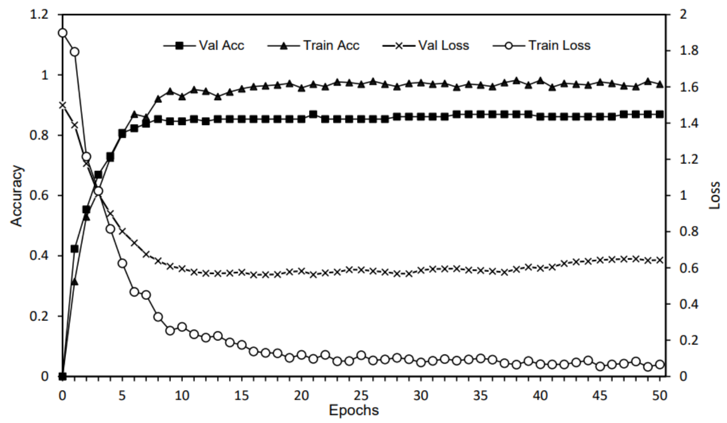


Figure 9. Inception-V3 training and validation accuracy and loss

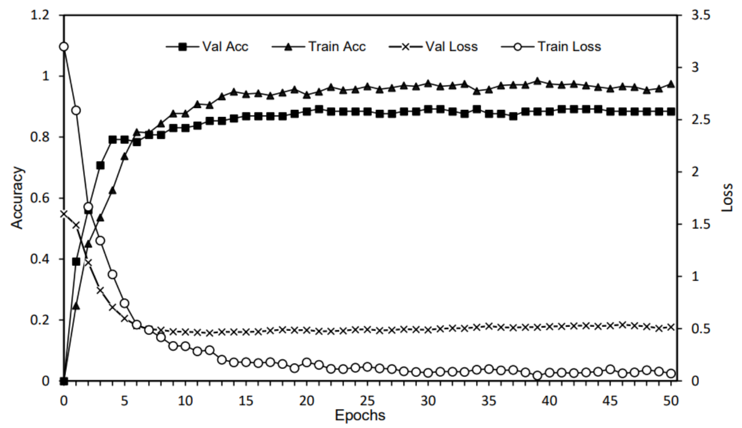


Figure 10. DensNet training and validation accuracy and loss

4.2.5. DenseNet121

In DenseNet121, the training loss decrease until 50 epochs. However, the validation loss starts to increase in epoch 12. It shows us that model start to over fit with the data training. The minimum accuracy and validation loss are 0.0529 and 0.4579, respectively. The validation accuracy starts to be stagnant in epoch 20. The maximum training and validation accuracy are 0.9847and 0.8846, respectively. The accuracy and loss graphic is shown in Figure 10.

4.3. Discussion

According to Table 2, the least layer is VGG-16 model as it is only 16 layers. In the parameter size, the DenseNet121 produced 9.3 million parameters that is the smallest number parameter among five of them. Consequently, the model size of DenseNet121 is the smallest too that is 113 Mb. In the training accuracy, the VGG-16, ResNet50, and Densnet121 can achieve 0.9821 accuracies. Although, the smallest training loss is achieved by ResNet50 model. The best validation accuracy is achieved by VGG-16 and Resnet50. However, the VGG-16 has a slightly lower validation loss than ResNet50. The lower loss means the model can better predict the test data. It is proved by the testing accuracy of VGG-16 can overcome all competitor as it achieved 0.9076. In addition, the training and testing time of VGG-16 is the smallest comparing the other models. The training and testing times are 141.72 s and 2.15 s, respectively. VGG 16 architectures shows its superiority compare to the rest in term of testing accuracy and time. VGG16 is the shallowest architecture with 16 layers. Therefore, the model size is the smallest compared to deeper architectures. Based on the results, we can confidence that the VGG-16 model is suitable for fruit disease detection.

Table 2. Accuracy and loss of training and execution time on 50 epochs

Model	Layers	Parameter size	Model size (mb)	Training accuracy	Training loss	Validation accuracy	Validation loss	Testing accuracy	Training time (s)	Testing time (s)
VGG-16	16	21,138,757	253	0.9821	0.0458	0.8923	0.3108	0.9076	141.72	2.15
ResNet50	50	28,307,845	340	0.9821	0.0422	0.8923	0.3885	0.8923	173.09	15.67
InceptionResNetV2	164	55,911,141	673	0.9796	0.0537	0.8846	0.4207	0.8846	554.12	49.88
InceptionV3	48	23,901,447	287	0.9821	0.0531	0.8692	0.5618	0.8923	178.75	19.46
DenseNet121	121	9,398,341	113	0.9847	0.0529	0.8846	0.4579	0.8923	274.29	30.9

In Kusrini *et al.* [1] we implemented VGG16 classifier for recognizing the pest on leaf images. Since image data collection of infected mango leaf is not easy to collect huge number of data, an augmentation of the original sample was carried out in order to improve the classification performance. We achieved 71% on testing data for that experiment, it is due to the cluttered background, visual similarities among many different classes as mentioned by Barbedo [30]. In this paper, we expand the classification task to the mango fruit dataset and as can be seen in Table 2 the best achievement reported by VGG16 even without data augmentation implemented. The fact that the fruit dataset is as small as the leaf dataset but the background much tidy and the similarity between class much obvious. It is lead to well performance among all the evaluated architectures.

We also found an interesting fact that the deeper architecture cannot improve better accuracy. The fact that the problem is simple with only five target classes, low interclass similarity and tidy background enable to simpler network to capture the pattern of the training very well. Longer network lead to overfitting network and it is indicated by the high training accuracy while lower validation and training accuracy.

With the small dataset, we can achieve 90.76% of accuracy in test set and it would be quite useful if we brought thus result to the current implementation on mobile application pest detection. It is because the implementation is not purely automatics but we put the human in the loop. In the future, along with the implementation of this recognition task and human feedback, we do expect the rich dataset captured from the field. With more dataset and human in the loop as the user and the crowd labeling expert we expect more data and we can retrain the classifier and gain better recognition rate.

The model built in this research will be applied in a mobile application. To reduce the possibility of errors due to different input data, before the process of classifying pests on fruit, the application will identify whether the image entered is a fruit image or something else. The identification model will use the results of previous research [47] by adding fruit data as one of the classes.

5. CONCLUSION

VGG 16 can effectively recognize the mango fruit pest with 90% of accuracy and about 0.0165 second recognition time. The speed and the accuracy is acceptable for mobile application pest recognition system. The rest of the algorithms shows lower accuracy and time-consuming recognition therefore for current available dataset we conclude that the implementation of VGG16 is acceptable.

ACKNOWLEDGEMENTS

The research activity leading to the publication has been funded by the Ministry of Education, Culture, Research and Technology of The Republik Indonesia with contract number 311/E4.1/AK.04.PT/2021, 3281.11/LL5/PG/2021, 188/KONTRAK-LPPM/AMIKOM/VII/2021. Author would like to express gratitude to The Ministry of Farming for its data support. This research has been made possible by endless administrative support from the research directorate of the Universitas Amikom Yogyakarta and Faculty of Farming Universitas Gadjah Mada Indonesia. Also, the authors would like to thank NVIDIA Corporation for the donation of the GPU used in this study.

REFERENCES

[1] K. Kusrini *et al.*, "Data Augmentation for Automated Pest Classification in Mango Farms," *Computers and Electronics in Agriculture*, vol. 179, 2020, doi: 10.1016/j.compag.2020.105842.

[2] A. Picon, A. A. -Gila, M. Seitz, A. O. -Barredo, J. Echazarra, and A. Johannes, "Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild," *Computers and Electronics in Agriculture*, vol. 161, pp. 280–290, 2019, doi: 10.1016/j.compag.2018.04.002.

[3] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.

[4] S. Panigrahi, A. Nanda, and T. Swarnkar, "A Survey on Transfer Learning," *Intelligent and Cloud Computing*, vol. 194, pp. 781–789, 2020, doi: 10.1007/978-981-15-5971-6_83.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, 2017, doi: 10.1145/3065386.

[6] A. Krizhevsky, "One weird trick for parallelizing convolutional neural networks," *ArXiv*, pp. 1-7, 2014. [Online]. Available: https://arxiv.org/pdf/1404.5997.pdf

[7] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594.

[8] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks," *ArXiv*, 2014. [Online]. Available: https://arxiv.org/pdf/1312.6229.pdf

[9] K. Simonyan, and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *ArXiv*, pp. 1–14, 2015. [Online]. Available: https://arxiv.org/pdf/1409.1556.pdf

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[11] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," in *Proc. of the AAAI Conference on Artificial Intelligence*, 2017, vol. 31, no. 1, doi: 10.1609/aaai.v31i1.11231.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818-2826, doi: 10.1109/CVPR.2016.308.

[13] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261-2269, doi: 10.1109/CVPR.2017.243.

[14] K. Kusrini et al., "Automatic Mango Leaf and Trunk Detection as Supporting Tool of Mango Pest Identifier (MPI)," *2021 3rd International Conference on Cybernetics and Intelligent System (ICORIS)*, 2021, pp. 1-7, doi: 10.1109/ICORIS52787.2021.9649558.

[15] Kusrini *et al.*, "User Evaluation of Mobile based Smart Mango Pest Identification," in *2020 8th International Conference on Cyber and IT Service Management (CITSM)*, 2020, pp. 1-7, doi: 10.1109/CITSM50537.2020.9268883.

[16] M. Bhange and H. A. Hingoliwala, "Smart Farming: Pomegranate Disease Detection Using Image Processing," *Procedia Computer Science*, vol. 58, pp. 280–288, 2015, doi: 10.1016/j.procs.2015.08.022.

[17] J. G. A. Barbedo, L. V. Koenigkan, and T. T. Santos, "Identifying multiple plant diseases using digital image processing," *Biosystems Engineering*, vol. 147, pp. 104–116, 2016, doi: 10.1016/j.biosystemseng.2016.03.012.

[18] V. Singh and A. K. Misra, "Detection of plant leaf diseases using image segmentation and soft computing techniques," *Information Processing in Agriculture*, vol. 4, no. 1, pp. 41–49, 2017, doi: 10.1016/j.inpa.2016.10.005.

[19] S. Zhang, X. Wu, Z. You, and L. Zhang, "Leaf image based cucumber disease recognition using sparse representation classification," *Computers and Electronics in Agriculture*, vol. 134, pp. 135–141, 2017, doi: 10.1016/j.compag.2017.01.014.

[20] S. Zeng, B. Zhang, and Y. Du, "Joint distances by sparse representation and locality-constrained dictionary learning for robust leaf recognition," *Computers and Electronics in Agriculture*, vol. 142, pp. 563–571, 2017, doi: 10.1016/j.compag.2017.11.013.

[21] D. S. Tan *et al.*, "AuToDiDAC: Automated Tool for Disease Detection and Assessment for Cacao Black Pod Rot," *Crop Protection*, vol. 103, pp. 98–102, 2018, doi: 10.1016/j.cropro.2017.09.017.

[22] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification," *Computational Intelligence and Neuroscience*, 2016, doi: doi: 10.1155/2016/3289801.

[23] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, 2016, doi: 10.3389/fpls.2016.01419.

[24] D. P. Hughes and M. Salathe, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *ArXiv*, 2015. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1511/1511.08060.pdf

[25] J. Deng, W. Dong, R. Socher, L. -J. Li, K. Li, and Li F. -Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.

[26] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," *Neurocomputing*, vol. 267, pp. 378–384, 2017, doi: 10.1016/j.neucom.2017.06.023.

[27] J. Lu, J. Hu, G. Zhao, F. Mei, and C. Zhang, "An in-field automatic wheat disease diagnosis system," *Computers and Electronics in Agriculture*¸vol. 142, pp. 369–379, 2017, doi: 10.1016/j.compag.2017.09.012.

[28] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Computers and Electronics in Agriculture*, vol. 161, pp. 272–279, 2019, doi: 10.1016/j.compag.2018.03.032.

[29] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018, doi: 10.1016/j.compag.2018.01.009.

[30] J. G. A. Barbedo, "Factors influencing the use of deep learning for plant disease recognition," *Biosystems Engineering*, vol. 172, pp. 84–91, 2018, doi: 10.1016/j.biosystemseng.2018.05.013.

[31] S. Zhang, W. Huang, and C. Zhang, "Three-channel convolutional neural networks for vegetable leaf disease recognition," *Cognitive Systems Research*, vol. 53, pp. 31–41, 2019, doi: 10.1016/j.cogsys.2018.04.006.
[32] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based Learning Applied to Document Recognition," in *Proc. of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998, doi: 10.1109/5.726791.
[33] J. Hang, D. Zhang, P. Chen, J. Zhang, and B. Wang, Identification of Apple Tree Trunk Diseases Based on Improved Convolutional Neural Network with Fused Loss Functions," in *ICIC 2019: Intelligent Computing Theories and Application*, 2019, pp. 274-283, doi: 10.1007/978-3-030-26763-6_26.
[34] S. Coulibaly, B. K. -Foguem, D. Kamissoko, and D. Traore, "Deep neural networks with transfer learning in millet crop images," *Computers in Industry*, vol. 108, pp. 115–120, 2019, doi: 10.1016/j.compind.2019.02.003.
[35] J. G. A. Barbedo, "Plant disease identification from individual lesions and spots using deep learning," *Biosystems Engineering*, vol. 180, pp. 96–107, 2019, doi: 10.1016/j.biosystemseng.2019.02.002.
[36] K. Srunitha and D. Bharathi, "Mango Leaf Unhealthy Region Detection and Classification," in *Computational Vision and Bio Inspired Computing*, 2018, pp. 422–436, doi: 10.1007/978-3-319-71767-8_35.
[37] U. P. Singh, S. S. Chouhan, S. Jain, and S. Jain, "Multilayer Convolution Neural Network for the Classification of Mango Leaves Infected by Anthracnose Disease," *IEEE Access*, vol. 7, pp. 43721–43729, 2019, doi: 10.1109/ACCESS.2019.2907383.
[38] K. Kusrini *et al.*, "Dataset for pest classification in Mango farms from Indonesia," *Mendeley Data*, 2020. [Online]. Available: https://data.mendeley.com/datasets/94jf97jzc8/1 (accessed Jan. 21, 2021).
[39] C. A. Ferreira *et al.*, "Classification of Breast Cancer Histology Images Through Transfer Learning Using a Pre-trained Inception Resnet V2," in *ICIAR 2018: Image Analysis and Recognition*), 2018, pp. 763–770, doi: 10.1007/978-3-319-93000-8_86.
[40] N. Dominic, Daniel, T. W. Cenggoro, A. Budiarto, and B. Pardamean, "Transfer learning using inception-resnet-v2 model to the augmented neuroimages data for autism spectrum disorder classification," *Communications in Mathematical Biology and Neuroscience*, pp. 1–21, 2021, doi: 10.28919/cmbn/5565.
[41] A. Thomas, Harikrishnan P. M., Palanisamy P., and V. P. Gopi, "Moving Vehicle Candidate Recognition and Classification Using Inception-ResNet-v2," in *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2020, pp. 467-472, doi: 10.1109/COMPSAC48688.2020.0-207.
[42] F. Baldassarre, D. G. Morín, and L. R. -Guirao, "Deep Koalarization: Image Colorization using CNNs and Inception-ResNet-v2," *ArXiv*, pp. 1–12, 2017. [Online]. Available: https://arxiv.org/pdf/1712.03400.pdf
[43] H. V. Hoang and M. T. Tran, "DeepSense-Inception: Gait Identification from Inertial Sensors with Inception-like Architecture and Recurrent Network," in *2017 13th International Conference on Computational Intelligence and Security (CIS)*, 2017, pp. 594-598, doi: 10.1109/CIS.2017.00138.
[44] N. Dong, L. Zhao, C. H. Wu, and J. F. Chang, "Inception v3 based cervical cell classification combined with artificially extracted features," *Applied Soft Computing*, vol. 93, 2020, doi: 10.1016/j.asoc.2020.106311.
[45] X. Xia, C. Xu, and B. Nan, "Inception-v3 for flower classification," in *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, 2017, pp. 783-787, doi: 10.1109/ICIVC.2017.7984661.
[46] O. Albatayneh, L. Forslöf, and K. Ksaibati, "Image Retraining Using TensorFlow Implementation of the Pretrained Inception-v3 Model for Evaluating Gravel Road Dust," *Journal of Infrastructure Systems*¸ vol. 26, no. 2, 2020, doi: 10.1061/(ASCE)IS.1943-555X.0000545.
[47] K. Kusrini et al., "Automatic Mango Leaf and Trunk Detection as Supporting Tool of Mango Pest Identifier (MPI)," *2021 3rd International Conference on Cybernetics and Intelligent System (ICORIS)*, 2021, pp. 1-7, doi: 10.1109/ICORIS52787.2021.9649558.

BIOGRAPHIES OF AUTHORS

Kusrini is a professor from Universitas AMIKOM Yogyakarta Indonesia. She finished her doctoral program from Universitas Gadjah Mada Yogyakarta Indonesia in 2010. She is interested in exploring many things about machine learning and other artificial intelligence field. She also loves in doing research about decision support system and database. She is member of the IEEE and IEEE Systems, Man, and Cybernetics Society. She can be contacted at email: kusrini@amikom.ac.id. Website: http://kusrini.com/.

Suputa currently works at the Department of Plant Protection, Faculty of Agriculture, Universitas Gadjah Mada. He conducts research in the area of insect taxonomy, systematic and integrated pest management (IPM), generally in Agricultural Entomology. His expertise includes the identification of fruit flies, ants, aquatic insects, other insect pests using morphological and molecular characteristics (DNA) and also consultants on integrated pest management practices. His current research on Area Wide IPM of fruit fly. He can be contacted at email: puta@ugm.ac.id. Website: https://hpt.faperta.ugm.ac.id/suputa/.

Arief Setyanto was born in 1975 in Banyumas, Central Java, Indonesia. Currently he works as a lecturer in the Universitas Amikom Yogyakarta Indonesia. He received his PhD in The School of Computer Science and Electronics Engineering (CSEE), University of Essex, Colchester The United Kingdom in 2016 under Indonesian Government scholarship. He received bachelor and Master degree from Gadjah Mada University, Yogyakarta, Indonesia in 1998, 2003 respectively. His research interest includes, video/image segmentation and understanding, object tracking, visual content metadata, information retrieval and cryptography. He can be contacted at email: arief_s@amikom.ac.id.

I Made Artha Agastya earned a Master's degree in information technology, focusing on pattern recognition and data clustering, from Universitas Gadjah Mada, Indonesia in 2017. He is presently enrolled at Taylor's University, Malaysia, where he is pursuing a PhD in signal processing and recognition. Also, he is a lecturer in the informatics department, Universitas AMIKOM Yogyakarta, Indonesia. He was involved in two grant projects funded by the Ministry of Education, Culture, Research and Technology, Indonesia. Among his research interests are machine learning, affective computing, EEG signal processing, deep learning, and computer vision. He can be contacted at email: artha.agastya@amikom.ac.id.

Herlambang Priantoro is a Technical Analyst at PT Bank Mandiri (Persero) Tbk. Experienced Development Team with a demonstrated history of working in the information technology and services industry. He can be contacted at email: lambanx@gmail.com.

Sofyan Pariyasto received the M. Kom degree in Amikom University Yogyakarta in 2019. He was born in Sragen in 1989. Currently, he is active as a Product Owner in a growing private company. He has interests in AI, programming and data processing. He can be contacted at email: sofyanpariyasto@gmail.com.