

Reconfiguration layers of convolutional neural network for fundus patches classification

Wahyudi Setiawan¹, Moh. Imam Utoyo², Riries Rulaningtyas³

¹Informatics Department, University of Trunojoyo Madura, Indonesia

²Mathematics Department, Universitas Airlangga, Indonesia

³Physics Department, Universitas Airlangga, Indonesia

Article Info

Article history:

Received Aug 25, 2019

Revised Dec 10, 2019

Accepted Jun 17, 2020

Keywords:

Classification

Convolutional neural network

Fundus image

Gradient descent

Reconfiguration layers

ABSTRACT

Convolutional neural network (CNN) is a method of supervised deep learning. The architectures including AlexNet, VGG16, VGG19, ResNet 50, ResNet101, GoogleNet, Inception-V3, Inception ResNet-V2, and Squeezenet that have 25 to 825 layers. This study aims to simplify layers of CNN architectures and increased accuracy for fundus patches classification. Fundus patches classify two categories: normal and neovascularization. Data used for classification is MESSIDOR and Retina Image Bank that have 2,080 patches. Results show the best accuracy of 93.17% for original data and 99,33% for augmentation data using CNN 31 layers. It consists input layer, 7 convolutional layers, 7 batch normalization, 7 rectified linear unit, 6 max-pooling, fully connected layer, softmax, and output layer.

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



Corresponding Author:

Riries Rulaningtyas,

Department of Physics,

Universitas Airlangga,

Kampus C, Jl. Dr. Ir. H. Soekarno, Mulyorejo, Surabaya, Jawa Timur, 60115, Indonesia.

Email: riries-r@fst.unair.ac.id

1. INTRODUCTION

CNN is a development of conventional artificial neural network (ANN). If ANN has three to four layers, CNN has tens to hundreds of layers [1]. CNN has better feature learning capabilities compared with other deep learning methods, good capabilities for augmentation data such as image scalation, rotation, and translation [2]. Research on the classification of normal and neovascularization fundus has been done using machine learning. The previous studies use manual feature extraction that requires the right selection of features. It produces features that are suitable for the classification process. They were using different types and number of features. Jelinek et al. using 5 morphological features [3], Goatman et al. using 15 morphological and color features [4], Gupta et al. using 8 texture and morphological features [5], Akram et al. using 10 morphological features [6], Bhargavi et al using 20 features texture [7], and Welikala et al. using 21 morphological and color features [8]. The diverse types and number of features in manual feature extraction are obstacles in determining the right features for the classification process.

CNN performs feature extraction automatically in a network. There are previous studies that classified fundus images including Pratt et al. using CNN 38 layer, accuracy of classification result is up to 75% [9]. Lee et al. using CNN 39 layer, accuracy of classification result is up to 93.45% [10]. Takahashi et al. using GoogleNet 144 layer, accuracy of classification result is up to 81% [11], Lam et al. using AlexNet 25 layer, VGG16 41 layer, GoogleNet 144 layer, ResNet50 177 layer and Inception-V3 316 layer, accuracy of classification results is up to 96% [12]. Previous studies on classification of fundus images using CNN

architecture have 25 to 316 layers. The deeper layer increasing the number of parameters. This affects a longer computation time. Therefore, we need to find a way to determine the number of layers that are simpler but have a better percentage of accuracy than previous studies. Furthermore, design a CNN architecture that is suitable for fundus image classification. Determination of the layers that need to be in the model. Architecture is used: AlexNet [13], VGG16, VGG19 [14], ResNet50, ResNet101 [15], GoogleLet [16], Inception-V3 [17, 18], Inception-ResNetV2 [17], and Squeezenet [18]. Comparisons are made to get the CNN architecture that best fits the data. Then do the layer modification on the selected CNN architecture. A novelty in this study is first, a new CNN architecture configuration was created. Second, better computation time, and third, data was patches of normal and neovascularization fundus.

2. RESEARCH METHOD

The modified CNN architecture was VGG. It's based on research that has been done using the same data. The results showed that VGG produces better accuracy compared to other CNN architectures. The percentage of sensitivity, specificity, and accuracy are 90.4%, 88.2% and 89.3% for CNN without gradient descent optimization, also produced 90.7%, 94% and 92.31% for CNN with gradient descent optimization [19].

The VGG architecture that was tested: VGG16 and VGG19. This architecture has 41 and 47 layers including the input and output layers. Convolutional layer consists of 13 and 16 layers. ReLu consists of 15 and 17 layers. Pooling used is max-pooling which consists of 5 layers and also has 2 layers dropout 0.5. VGG16 has a number of parameters >138 million, while VGG19 has a number of parameters >143 million. The large number of layers and parameters burdensome computing speed. Therefore, reconfiguration layer had focus on these 2 problems. Comparison between VGG16 and VGG19 shows in Table 1.

Table 1. The comparison of VGG architecture

CNN	Layer	Convolution layer	Parameter
VGG16	41	16	>138 million
VGG19	47	19	>143 million

Fundus image classification requires at least 7 layers of CNN. These layers are input layer, convolutional layer, rectified linear unit (ReLU), pooling layer, fully connected layer (FCL), softmax, and output layer. Another type of layer that can be added to CNN layer configuration is batch normalization (BN). BN normalizes the data on same scale. This aims to speed-up computation [20]. The testing was conducted using 7 to 35 layers of CNN. This step also using augmentation data with image rotation and translation.

CNN processes images through a network layer and produces output for a particular class. Each layer does learning. The output of each layer is used as input for the next layer. At the beginning of the network, layer produces simple features such as color, brightness, and edges. Furthermore, network will produce more complex features [21]. CNN layers have a part of convolutional layer include filter, stride, and padding. A stride is several shifts in the convolution layer. Padding adds zero values to all the outer edges of matrix. Another layer of CNN is batch normalization, ReLU, pooling layer, FCL, softmax, and output layer. Moreover, CNN parameter in each layer consists of weights and biases [22].

CNN parameter calculated with the backpropagation method. Backpropagation workflow consists of a forward-pass and a backward-pass. When forward-pass, CNN produces an error or loss function. Loss function is difference between prediction and target. Loss functions are minimized using gradient descent optimization algorithm [23]. Procedure for correcting loss functions using backward pass. This occurs iteratively until one of the two conditions is reached: maximum epoch has been reached or data validation was overfitting [24].

Overfitting occurs when accuracy of training data increases and accuracy of validation data decreases. In this condition, training process will be stopped earlier than an epoch that has been set. Figure 1 shows CNN backpropagation training. Backpropagation aims to update weight and bias based on errors or losses obtained during feedforward process. The steps of backpropagation are [25]:

- Calculate loss function during feedforward process. Loss function is a difference between target and prediction
- Calculate loss function gradient of all existing parameters by finding partial derivative of a function
- Update all weighting parameters and bias using gradient descent optimization algorithm by adding old weight with new gradient value. Gradient Descent used for parameter optimization. There are 3 Gradient Descent method: Momentum, RMSProp, and Adam

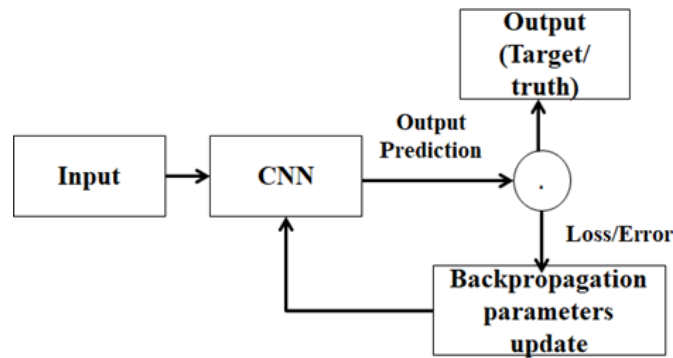


Figure 1. CNN backpropagation training [25]

2.1. Fundus classification

Fundus classification is carried out by testing various scenarios. The CNN algorithm shown in Figure 2.

```

Input : fundus patches data, fundus label (neovascularization, normal)
Process :
1 Divide fundus patches, set labels randomly 70% for training data and 30% for validation data
2 Reconfiguration layer of CNN with training options (7,11,15,19,23,27,31 and 35 layers):
3 repeat :
4   Forward Propagation:
5   fk ← filter kernel
6   cd ← Convolution2D(fundus patches, fk, stride, padding)
7   bn ← batch-normalization (cd)
8   rl ← ReLU (bn)
9   mp ← MaxPooling(rl)
10  fc ← Fully_connected(mp)
11  class label ← softmax(fc)
12  Backward Propagation:
13  conduct backward propagation with GDM, RmsProp or Adam
14 until max-epoch reached or overfitting
15 use the trained network to predicts the labels

Output: sensitivity, specificity, accuracy
  
```

Figure 2. CNN algorithm for fundus patches classification

The complete steps for fundus classification while looking for the appropriate reconfiguration layer are as follows:

- Input the original data 2,080 images. The data is divided into 2 parts 70% for training and 30% for testing. Additionally, add rotational and translational augmentation data. Furthermore, create a network architecture with the least layers to be able to carry out classification tasks. The minimum number of layers consists of 7 that consist of input layer, convolutional layer, normalization layer, ReLU, FCL, softmax, and output layer
- Initializing the variables values include learning rate, max-epoch, validation frequency, and minibatch size
- Fundus image classification using CNN algorithm as shown in Figure 2
- Calculate performance measures using sensitivity, specificity, and accuracy
- Check the accuracy value. If the accuracy value is > 92.31% which is the best accuracy of the VGG, then go to step 8. If not to step 6
- Check the number of layers. If the number of layers > 41, then it's complete. If not to step 7. The number of layers 41 is the number of VGG16 layers
- Add a block of convolutional layer consisting of convolutional layer, normalization layer, ReLU, and max-pooling layer. Next to step 3
- Set the number of layers and the performance measures
- Validation using 10-fold cross validation [26]. Finished

To find out relationship pattern between accuracy and variable, 15 test scenarios are set out in Table 2. Scenarios consist of small, medium and big variable values. The variable consist of learning rate, max-epoch, validation frequency, and minibatch size.

Table 2. Scenario of varying variable values for reconfiguration layers

Scenario	Learning rate	Max-epoch	Validation frequency	Minibatch size
1	0,0001	6	3	64
2	0,0001	30	3	64
3	0,0001	30	200	64
4	0,0001	200	500	128
5	0,0001	200	500	256
6	0,001	6	3	64
7	0,001	30	3	64
8	0,001	30	200	64
9	0,001	30	200	128
10	0,001	200	500	256
11	0,01	6	3	64
12	0,01	30	3	64
13	0,01	200	200	64
14	0,01	200	500	128
15	0,01	200	500	256

3. RESULTS AND DISCUSSION

Additionally, the augmentation data used for varied of validation data. It consists of image rotation and translation. The performance measures result of the reconfiguration layers with varying variable values are shown in Table 3.

Table 3. Result using original and augmentation data

Scenario	Layer	Optimize method	Original data		Augmentation data	
			Acc. (%)	Time (Sec.)	Acc. (%)	Time (Sec.)
1	35	RmsProp	85,7	18	85,9	27
2	23	GDM	85,3	76	80,45	47
3	23	RmsProp	88,9	367	96,15	387
4	31	GDM	92,6	680	97,12	685
5	35	GDM	93,1	705	96,79	720
6	31	GDM	83,5	18	82,53	9
7	35	Adam	84,8	22	79,81	10
8	19	RmsProp	87,3	749	85,42	750
9	23	Adam	89,9	363	87,5	362
10	31	GDM	91,2	672	94,87	674
11	35	Adam	82,5	27	83,49	20
12	35	GDM	81,9	15	84,78	29
13	27	Adam	89,9	1181	94,39	1172
14	27	GDM	88,6	1169	96,79	1178
15	27	GDM	89,4	1156	93,59	1166

Four variables are combined to find the best accuracy. There is no regular pattern of results test. The results with 15 scenarios show best parameters using learning rate value 0,0001, max-epoch 200, validation frequency 500 and minibatch size 128. The performance measure of accuracy was 97.12%. The results show best accuracy can initialize small learning rate value, big max-epoch and medium validation frequency and minibatch size. The gradient descent with momentum optimization was the best optimization with 10 out of 15 scenarios produce best accurac. But this does not justify that GDM algorithm is better than the RMSProp or Adam algorithm because it has not been applied to another case studies.

Furthermore, data for cross-validation is divided into ten parts. Each section consists of 208 patches used as test data, while other data is used as training. The results are shown in Table 4. The results show that the average percentage of accuracy using data augmentation had better results. Comparison of sensitivity, spesificity, an accuracy of VGG and new configuration layer show in Table 5. It shows the highest comparison of accuracy achieved by measuring sensitivity and specificity.

CNN modified architecture with 31 layers, learning rate 0,0001, max-epoch 200, validation frequency 500 and minibatch size 256 can increase percentage of sensitivity, specificity and accuracy of fundus patches classification: normal and neovascularization. Figure 3 shows the modification architecture of VGG that uses 31 layers. Architecture consist of an input layer, 7 convolutional layers, 7 ReLU, 7 Batch

Normalization, 6 max-pooling, Fully Connected layer, Softmax dan output layer. Table 6 shows a comparison of VGG16, VGG 19 dan modified CNN 31 layer. Comparison feature of VGG and proposed method can find in Table 7.

Table 4. Accuracy and computation time with 10 fold cross-validation

No of test	Original data		Augmentation data	
	Accuracy (%)	Time	Accuracy (%)	Time
1	91,35	14 min. 11 sec.	99,52	14 min.17 sec.
2	95,19	14 min. 38 sec.	100	14 min. 32 sec.
3	92,79	14 min.54 sec.	98,56	14 min. 12 sec.
4	95,15	14 min. 18 sec.	99,52	14 min. 23 sec.
5	93,27	14 min. 20 sec.	100	14 min. 15 sec.
6	93,75	14 min. 11 sec.	99,52	14 min. 27 sec.
7	93,27	14 min. 47 sec.	98,56	14 min. 20 sec.
8	94,23	14 min. 22 sec.	99,04	14 min. 23 sec.
9	91,35	14 min. 10 sec.	100	14 min. 06 sec.
10	91,35	14 min. 07 sec.	98,56	14 min. 05 sec.
Avg	93,17	14 min. 24 sec.	99,33	14 min. 18 sec.

Table 5. Comparison of performance measure

Step	Architecture	Sensitivity	Specificity	Accuracy
1	VGG19	87.8	90.7	89,3
2	VGG16	94.2	90.4	92,3
3	CNN 31 layer	93.67	92.7	93,17
	CNN 31 layer (augmentation data)	98.9	99.78	99,33

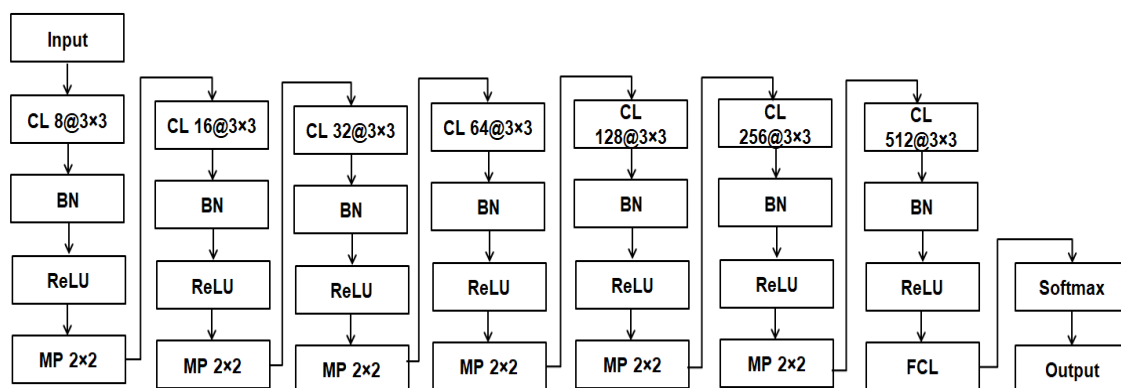


Figure 3. The architecture of VGG 31 layer with CL=convolutional layer, BN=Batch normalization, MP=max pooling

Table 6. Comparison of VGG architecture

Architecture	VGG16	VGG19	VGG 31 layer
Layer	41	47	31
Convolutional layer	13	16	7
Batch Normalization	0	0	7
ReLU	15	17	7
max-pooling	5	5	6
dropout	2	2	0
Learning parameter	138.357.494	143.667.190	1.584.826

Results of fundus patches classification using a modified CNN 31 layers architecture compared to AlexNet, GoogleLet, VGG16, VGG19 and ResNet50. The max-epoch, minibatch-size, learning rate and validation-frequency values on CNN architecture are set according to modified CNN 31 layers: 200, 128, 0,0001 and 500. Comparison of the results with previous studies is shown in Table 6.

Results show that average accuracy using AlexNet is 96.2% in original data and 99.4% in augmentation data. The average accuracy of GoogleNet is 93.3% using original data and 99.3% using augmentation data. The experiment results of the modified CNN 31 layer architecture have an average accuracy of 93.17% using the original data and 99.3% using augmentation data. The test results using the VGG16, VGG19 and ResNet50 architectures cannot be done because of an error during training. This happens because of the combination of minibatch-size values, CNN architecture and GPU specifications unable to carry out the training process.

Table 7. Comparison of accuracy and computation time

Architecture	Original data		Augmentation data	
	Acc.(%)	Time	Acc.(%)	Time
AlexNet	96.2	18 min 30 sec	99.4	18 min 40 sec
GoogleNet	93,3	62 min 38 sec	99,3	62 min 54 sec
CNN 31 layer	93,17	14 min 24 sec	99,3	14 min 18 sec

The results also show that the modified CNN 31 layer architecture requires an average computation time of 14 minutes 24 seconds using the original data and 14 minutes 18 seconds using augmentation data. AlexNet requires an average computation time of 18 minutes 30 seconds using original data and 18 minutes 40 seconds using augmentation data. GoogleNet Architecture requires an average computing time of 62 minutes 38 seconds using the original data and 62 minutes 54 seconds using augmentation data.

4. CONCLUSION

The best accuracy for fundus patches classification are VGG16 and VGG19 architecture. Both have the same architecture, the difference is a number of each layer consisting of 41 and 47 layers. Based on this, the experiment was made based on a modified VGG architecture. The modification layer consist batch normalization that used in each convolutional layer, data augmentation using image rotation and translation. It also used gradient descent optimization. The experiment uses a modified CNN 31 layers with original data has an average accuracy of 93.17%, an average sensitivity of 93.67% and an average specificity of 92.7%. The experiment uses a modified CNN 31 layers architecture with augmentation data has an average accuracy of 99.33%, an average sensitivity of 98.9% and an average specificity of 99.78%. The average computing time is 14 minutes 24 seconds for original data and 14 minutes 18 seconds for augmentation data.

REFERENCES

- [1] C. Steger, M. Ulrich, and C. Wiedemann, *Machine Vision Algorithms and Applications*, 1st Edition, Weinheim: Wiley-VCH, 2007.
- [2] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision : A Brief Review," *Hindawi Comput. Intell. Neurosci.*, vol. 2018, special issue, pp. 1–13, 2018.
- [3] H. F. Jelinek, M. J. Cree, J. J. G. Leandro, J. V. B. Soares, R. M. Cesar, and a Luckie, "Automated segmentation of retinal blood vessels and identification of proliferative diabetic retinopathy," *Journal of the Optical Society of America A*, vol. 24, no. 5, pp. 1448–1456, 2007.
- [4] K. A. Goatman, A. D. Fleming, S. Philip, G. J. Williams, J. A. Olson and P. F. Sharp, "Detection of New Vessels on the Optic Disc Using Retinal Photographs," in *IEEE Transactions on Medical Imaging*, vol. 30, no. 4, pp. 972-979, 2011.
- [5] G. Gupta, S. Kulasekaran, K. Ram, N. Joshi, M. Sivaprakasam, and R. Gandhi, "Local characterization of neovascularization and identification of proliferative diabetic retinopathy in retinal fundus images," *Computerized Medical Imaging and Graphics*, vol. 55, pp. 124–132, Jan. 2017.
- [6] M. U. Akram, S. Khalid, A. Tariq, and M. Y. Javed, "Detection of neovascularization in retinal images using multivariate m-Mediods based classifier," *Computerized Medical Imaging and Graphic*, vol. 37, no. 5–6, pp. 346-357, July–September 2013.
- [7] R. B. V and R. K. Senapati, "Bright Lesion Detection in Color Fundus Images Based on Texture Features," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 5, no. 1, pp. 92–100, March 2016.
- [8] R. A. Welikala *et al.*, "Computerized Medical Imaging and Graphics Genetic algorithm based feature selection combined with dual classification for the automated detection of proliferative diabetic retinopathy," *Computerized Medical Imaging and Graphic*, vol. 43, pp. 64–77, 2015.
- [9] H. Pratt, F. Coenen, D. M. Broadbent, S. P. Harding, and Y. Zheng, "Convolutional Neural Networks for Diabetic Retinopathy," *Procedia Computer Science*, vol. 90, pp. 200-205, 2016.
- [10] C. S. Lee, D. M. Baughman, and A. Y. Lee, "Deep learning is effective for the classification of OCT images of normal versus Age-related Macular Degeneration," *Ophthalmol Retina*, vol. 1, no. 4, pp. 322–327, 2016.
- [11] H. Takahashi, H. Tampo, Y. Arai, Y. Inoue, and H. Kawashima, "Applying artificial intelligence to disease staging: Deep learning for improved staging of diabetic retinopathy," *PLoS One*, vol. 12, no. 6, pp. 1–11, 2017.
- [12] C. Lam, C. Yu, L. Huang, and D. Rubin, "Retinal Lesion Detection With Deep Learning Using Image Patches," *Investigative ophthalmology & visual science*, vol. 59, no. 1, pp. 590–596, 2018.
- [13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in neural information processing systems*, pp. 1097-1105, 2012.
- [14] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *ICLR*, pp. 1-14, 2015.
- [15] W. H. Organization *et al.*, "Global Report on Diabetes," *Curr. Med. Res. Opin.*, vol. 56, no. 1, pp. 1051–1062, 2014.
- [16] 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)*, Las Vegas, NV, pp. 2818-2826, 2016.

- [17] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pp. 4278-4284, 2017.
- [18] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," in *ICLR*, pp. 1-13, 2017.
- [19] W. Setiawan, "Comparison of Convolutional Neural Network for Fundus Classification," *Simantec*, vol. 7, no. 2, 2019.
- [20] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv preprint arXiv*, 2015.
- [21] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," In *European conference on computer vision*, Springer, Cham, vol. 8689, pp. 818-833, 2014.
- [22] W. Setiawan, M. . Utoyo, and R. Rulaningtyas, "Classification of neovascularization using convolutional neural network model," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 17, no. 1, pp. 463-473, February 2019.
- [23] S. Ruder, "An overview of gradient descent optimization," *arXiv preprint arXiv*, 2016.
- [24] J. E. Zafra, R. J. Moreno, and R. D. Hernández, "Comparison between Backpropagation and CNN for the Recognition of Traffic Signs," *International Journal of Applied Engineering Research*, vol. 12, no. 17, pp. 6814-6820, 2017.
- [25] S. Sena, "Pengenalan Deep Learning Part 3 : BackPropagation Algorithm," *Medium Corporation*, 2017. [Online]. Available at: <https://medium.com/@samuelsena/pengenalan-deep-learning-part-3-backpropagation-algorithm-720be9a5fbb8>.
- [26] T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning," *Math. Intell.*, vol. 27, no. 2, pp. 83-85, 2001.

BIOGRAPHIES OF AUTHORS



Wahyudi Setiawan received his Doctoral Degree in Mathematics and Natural Science from the Universitas Airlangga in 2019. He is currently a Lecturer in the Department of Informatics at University of Trunojoyo Madura, Indonesia. His research interests include deep learning, image processing and signal processing. He is a member of Association of Information System Indonesia (AISINDO) and International Association of Engineers (IAENG)



Moh. Imam Utoyo received his Doctoral Degree in Mathematics and Natural Science from the Universitas Airlangga in 2012. He is currently a lecturer in the Departement Mathemeatics at the same universities. His research interest include analysis mathemathics and fractional. He is a member of Indonesian Mathematical Society (IndoMS).



Riries Rulaningtyas received her Doctoral Degree in Electrical Engineering from the Institut Teknologi Bandung in 2015. She is currently a lecturer in the Department of Physics at the Universitas Airlangga. Her research interest include image processing, signal processing, machine learning and artificial intelligence. She is a member of Physical Society of Indonesia.