Face recognition for presence system by using residual networks-50 architecture

Yohanssen Pratama, Lit Malem Ginting, Emma Hannisa Laurencia Nainggolan, Ade Erispra Rismanda
Faculty of Informatics and Electrical Engineering, Institut Teknologi Del, Laguboti, Indonesia

Article Info

ABSTRACT

Presence system is a system for recording the individual attendance in the company, school or institution. There are several types presence system, including the manually presence system using signatures, presence system using fingerprints and presence system using face recognition technology. Presence system using face recognition technology is one of presence system that implements biometric system in the process of recording attendance. In this research we used one of the convolutional neural network (CNN) architectures that won the imagenet large scale visual recognition competition (ILSVRC) in 2015, namely the Residual Networks-50 architecture (ResNet-50) for face recognition. Our contribution in this research is to determine effectiveness ResNet architecture with different configuration of hyperparameters. This hyperparameters includes the number of hidden layers, the number of units in the hidden layer, batch size, and learning rate. Because hyperparameter are selected based on how the experiments performed and the value of each hyperparameter affects the final result accuracy, so we try 22 configurations (experiments) to get the best accuracy. We conducted experiments to get the best model with an accuracy of 99%.

Corresponding Author:

Yohanssen Pratama
Faculty of Informatics and Electrical Engineering
Institut Teknologi Del
Sisingamangaraja Street, Sitoluama, Laguboti 22381, Indonesia
Email: yohanssen.pratama@del.ac.id

1. INTRODUCTION

Biometrics is a term used to determine the DNA of an individual, hand geometry, face or physical characteristics, such as signatures, sounds and so on. Biometric systems are generally used to authenticate and identify individuals by analyzing the individual's physical characteristics, such as fingerprints, irises, veins and others [1]. Biometric systems use unique physical characteristics of individuals which different from others to be identified and analyzed to achieve certain goals [2].

One of biometric system that also superior and can be applied to the attendance system based on the comparison made is using faces [3]. As identity information, human faces have the advantage of being unique and free of imitation [4]. In the face recognition system, the technology used is face detection which is the first step in facial recognition process and face recognition [5]. One of the supporting media that can be relied upon in the attendance system using face detection and face recognition is a real time video camera. Camera in the face detection system have the advantage of application flexibility, so they do not require users to make direct contact with the attendance system [6].

In this research, based on a case study of presence system, we conducted a study of the application on biometric systems by using deep learning. Deep learning is a type of machine learning method that makes computers learn from experience and knowledge without explicit programming and extract useful patterns from raw data [7]. With the presence system using deep learning for face detection and face recognition, it is expected that the process of recording student attendance is more efficient, as well as reducing fraud that might occur.

In develop the presence system there will be 2 stages, namely face detection and face recognition. In the face detection stage, the haar cascade classifier method is used to detect elements on the face, namely the eyes, nose and mouth [8]. In the face recognition stage, the convolutional neural network (CNN) algorithm is used for the process of recognizing and matching input data with data on the model. In our research we only use CNN which the values of the kernel are determined by training, while a haar-feature is manually determined. While well-trained CNN could learn more parameters (and thus detect a larger variety of faces), haar based classifiers run faster [9]. Haar cascade detect human faces enclosed by a square and give center points of face elements (eyes, nose, and mouth) [10]. The haar cascade classifier is also called the Viola-Jones Method, which is the most widely used method for detecting objects. The application of human faces detection by using haar cascade classifier can be carried out to get a comprehensive result such as for detect human faces on thermal image [11].

Among all deep learning structure, CNN and recurrent neural networks (RNN) are the most popular structures [12]. As state above we use CNN because we want to get the best accuracy and CNN has been proven to be very effective in areas such as facial recognition and classification compared to another method [13]. Also, CNN extracts features automatically, so there is no need to select features manually [14]. There are already some researches in face recognition based on CNN, some of them implemented augmented reality to compared it with face database and give high accuracy [15], and many of them implementing softmax architecture for facial recognition which already proven give a good accuracy [16]. In here we propose to use ResNet-50 architecture for recognition, because has good performance if compared to simple CNN [17]. Residual networks (ResNet) are a convolutional network that is trained on more than 1 million images from the ImageNet database and for ResNet-50 the total number of weighted layers is 50, with 23534592 parameters that can be trained [18].

Our contribution in this research is to determine effectiveness ResNet architecture with different configuration of hyperparameters. This hyperparameters includes the number of hidden layers, the number of units in the hidden layer, batch size, and learning rate [19]. Because hyperparameter are selected based on experiments performed and the value of each hyperparameter affects the final result accuracy, so we try 22 configurations (experiments) to get the best accuracy [20]. We also want to prove that learning rate has a large influence on accuracy among hyperparameters in here [21].

2.   RESEARCH METHOD

The explanation of the data processing design scheme is as follows as shown in Figure 1:
− Data in the form of face images are used as input to be processed.
− Each person images are capture with 15 different position and expression in RGB color space and JPG format.
− Pre processing stage is the stage of image data uniformity consisting of uniforming the size of the image and image augmentation.
− In here we make each image has same specification, such as its color space and resolution.
− Classification, namely the stage of recognizing faces with several stages consisting of convolutional layers, pooling layers, flatten, fully connected layers and softmax.
− In here we use CNN with a certain number of layers. The number of layers that contribute to a model of data is called the depth of a model [22]. In every stage we do several hyperparameter configuration for every experiment that we conduct until we found the best combination and get the best accuracy. The detail about the hyperparameter configuration could be seen in the explanation subsection about the model experiment design.
− The output of processed data is information from the data that has been identified.
− In here we try to get the result wether the recognition system could recognize the person correctly or not. We do the experiment with 9 different persons, each of them will be identified by the system that we build, from that we could know how robust our system to recognize the face of each person.

2.1. Data

The images collected in this study are using the .jpg file format. The collected images are arranged in 53 directories, each directory representing one student. The format of the name in each directory is X_Y,

with the explanation X is the first name of the student and Y is the student's NIM, for example students on behalf of Emma with NIM 11S15003, then the directory folder naming format is Emma_11S15003. In each directory contains 15 different images by the same student. The format for naming image files is NamaMahasiswa_XX.jpg, with the explanation StudentName is the first name of the student and XX is the number of each picture starting from number 0. For example, for students on behalf of Emma, the image file format in the directory is Emma_0.jpg. The directory setting and naming of each file aims to make it easier for researchers to manage the images of each student. In here we use RGB colour space as an input, so in converting an RGB image into a matrix, it takes 3 matrices that can represent each of the red, green and blues values for each pixel [23].
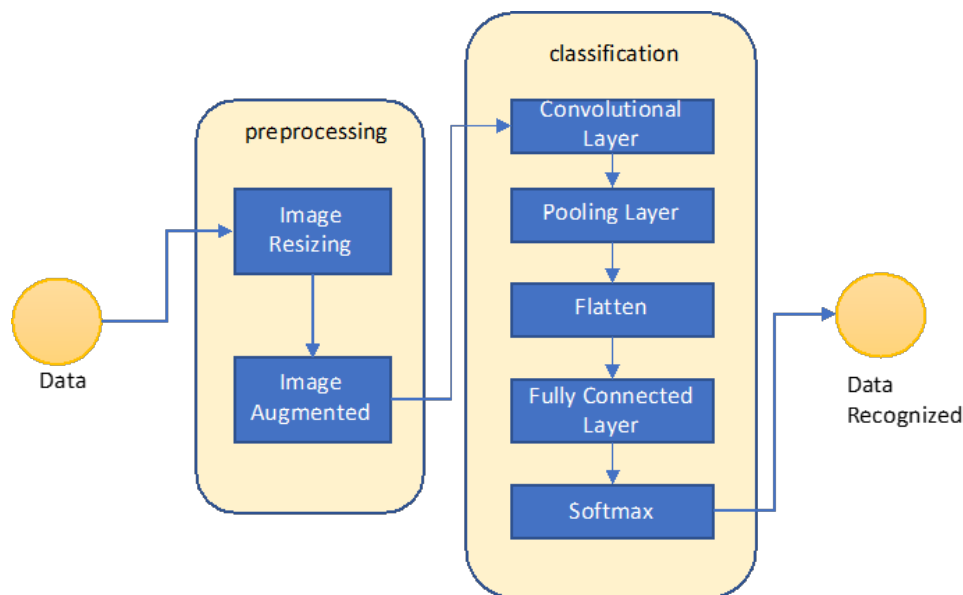
Figure 1. Data processing design scheme

2.2. Preprocessing

Preprocess data on a convolutional neural network has several stages, the first stage is Image Scaling at this stage, the input data will be equal in size. This stage is needed because the available image size does not always match the image size specified as a dataset. After that we continue with Augmentation process, at this stage, augmentation consists of 3 stages: Giving a blur effect to the image, giving a random noise effect (noise) and adding light intensity to the image. The purpose of this augmentation is to uniform image data in order to simplify the classification process.

We also do some geometrical operation such as flip, shift (translation), rotation, and segmentation. Flip consists of 3 stages: Data changing the image to horizontal, vertical and rotated horizontally and vertically. The purpose of this flip is to reproduce data to simplify the classification process. At Shift stage, we translate/shifts the location of the object from the original object in the data. Then in the rotation stage, the image will be shifted counterclockwise according to a predetermined angle. Last is the segmentation stage, segmentation is used to detect the edges of faces to get an image from an image.

2.3. Classification

The explanation of the scheme in Figure 2 is as follows:
− Input data will enter the convolution layer, which is the process of manipulating images to produce a new image to be entered at a later stage. We use zero padding in convolutional process [24] as shown in Figure 3.
− Pooling layer at this stage is to do calculations on each pixel of the image feature that has been converted in here into a matrix. The goal is to divide an image into several features to make it easier to do an image match.
− Flatten is the stage where the features produced at the pooling layer with a matrix size n x m will be provided that n> 1 and m> 1 will be converted to the order matrix 1x1.

− Fully connected layer which is the stage of producing output in the form of the probability of an image that will be used in the classification process of output data.
− Softmax is the stage of calculating probabilities on all labels in the data.
− The final result of this process is the value of the softmax calculation, which is the probability of each label in the data.
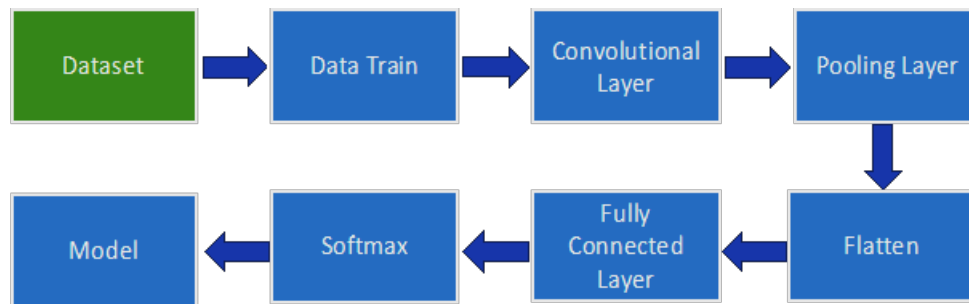

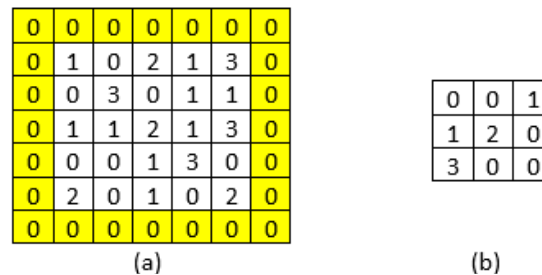Figure 2. Design of convolutional neural network model

Figure 3. Zero-padding operation in, (a) 5x5 matrices, (b) filter 3x3

2.4. Model experiment design

We divide the data that has been preprocessed into a data train and data test, with a share of 80% for the data train and 20% for the test data. 1050 data were divided into 2 parts, 840 data to be data train and 210 data to be test data. Data that has been divided into data train and data test will go through the training model stage using the convolutional neural network algorithm. In the process of training models, for the first experiment we use hyperparameter that has learning rate with value 0.1, epoch with value of 10 and step per epoch with value of 100. In data processing using convolutional neural network, the data goes through several stages, namely pooling layer, flatten, fully connected layer, softmax calculation to produce a model. The resulting model will be evaluated to find the accuracy of a model. The experimental design for this model can be seen in Figure 4.
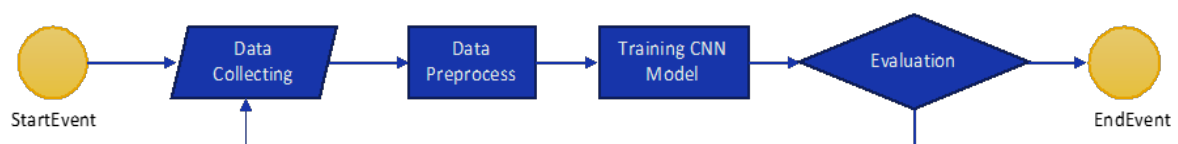

Figure 4. Model experiment design

3. RESULTS AND DISCUSSION

In this section, it is explained the results of research and at the same time is given the comprehensive discussion. The results obtained and discussion of the implementation of attendance system development using face recognition technology are as follows:

3.1. Result of data collections

Each person dataset must consist of 15 image data with different conditions, we use this based on design of smart door system for live face recognition based on image processing [25] and we do not use augmented reality database for comparison. We collected images of each individual faces by taking each individualfaces with different variations and then developing image collection techniques with preprocess data methods to homogenize the whole picture. The dataset sample could be seen in Table 1.

Table 1. Dataset example

Caption	Example for dataset
Strapping face, eyes facing the camera and without expression.	
Face shape facing left 45 °, eyes facing left and smiling expression.	
The face forms tilted to the right 45 °, and smiles.	
Face shape facing up, eyes closed and expressionless.	
The shape of the face is tilted to the right 45 °, and without expression.	
Face shape facing right 45 °, and expressionless.	
Face shape facing upwards of 45 °, and smiling expressions.	
The shape of the face is tilted to the left 45 °, and expressionless.	
Face shape facing up to 45 °, eyes facing up and without expression.	
Face shape facing right 45 ° and smiling expression.	
Face shape facing right 45 °, and expressionless.	
Face shape facing left 45 °, and smiling expressions.	
The shape of the face is tilted to the left 45 °, and the expression smiles.	
Face shape facing up with eyes staring straight at the camera and smiling expression.	
Face up with eyes closed and without expression.	

3.2. Result of preprocessing data

In this phase the initial data that already collected has different sizes and not uniform, so the preprocessing stage was needed in this research. We include the function of image normalization to ensure the uniformity in image size and augmentation. The tests that carried out in the preprocessing stage use 15 variations of image data for 1 class. Following is an example of the test results for one image data that has through the preprocessing phase that can be seen in Table 2.

At the preprocess phase, one image data produces 87 preprocess data. So, for 1 class that contains 15 variations of data, the total data generated after preprocessing is 87x15=1,305 data. From 53 classes collected, preprocess data will be obtained, that is 53 classes x 1,305 data = 69,165 preprocess data.

Table 2. Preprocessing result example

No	Preprocess	Result	Caption
1	Original data		Data collected manually
2	Added blur effect		The original data is given blur effect
3	Added random noise effect		The original data is given random noise effect
4	Added light intensity		The original data is given the effect of light intensity
5	Added random noise and light intensity		The original data is given random noise and light intensity effect
6	Added blur and light intensity		The original data is given blur and light intensity effect
7	Added random noise, blur, and light intensity		The original data is given random noise, blur, and light intensity effect
8	Change the picture horizontally		Picture that already given effect 2-7 rotated horizontally
9	Change the picture verticallys		Picture that already given effect 2-7 rotated vertically
10	Change the picture vertically and horizontally		Picture that already given effect 2-7 rotated horizontally & vertically
11	Gives a translational effect on the image		Picture that already given effect 2-10 given translation effect
12	Gives the rotation effect on the image		Picture that already given effect 2-10 given rotation effect
13	Perform segmentation (edge detection)		Segmentation on original data, no 4 & 6 data

3.2.1. Result of model testing

In this phase we conducted an experiment with data sharing which is 80:20, with lots of data 1050 with 840 for data train and 210 for test data. We also conducted an experiment by using 13050 data with 10440 for data train and 2610 for test data. The number of classes modeled is 10 classes, this is due to limitations on inadequate support resources for modeling 53 classes. The result of model testing with different configuration of hyperparameter can be seen in Table 3. From the results of the modeling experiments above, it can be concluded that the 22nd experiment has the best accuracy with 99% data train accuracy and 99% data test accuracy. This is because we has experimented with hyperparameters and obtained the right hyperparameters to build a model with value of data train accuracy and data test accuracy that reached 99%.

3.2.2. Result of prototype presence system

The results of this implementation are the prototype that built using a graphical user interface (GUI) provided by Python 3.6 and tkinter which is successfully built to recognize the face from each student by using previously trained model with 22nd experiment hyperparameter configuration as shown in Figure 5.

Table 3. Model testing result for 22 hyperparameters configuration

| Experiment Number:	Amount of Data	Data train	Data test	Learning rate	Epoch	Step per-epoch	Data Train Accuracy	Data Test Accuracy
1	1.050	840	210	0.1	10	100	9%	5%
2	1.050	840	210	0.01	20	100	11%	7.5%
3	1.050	840	210	0.001	10	100	55%	50%
4	1.050	840	210	0.0001	10	100	95%	85%
5	1.050	840	210	0.00001	10	100	84%	62.5%
6	1.050	840	210	0.000001	10	100	22%	37.5%
7	1.050	840	210	0.0001	20	100	98%	86.8%
8	1.050	840	210	0.0001	50	100	98%	87.5%
9	1.050	840	210	0.0001	70	100	98%	88%
10	1.050	840	210	0.0001	100	100	99%	94%
11	1.050	840	210	0.0001	100	150	99%	97.5%
…	…	…	…	…	…	…	…	…
21	13.050	10.440	2.610	0.0001	100	100	99%	98%
22	13.050	10.440	2.610	0.0001	100	150	99%	99%
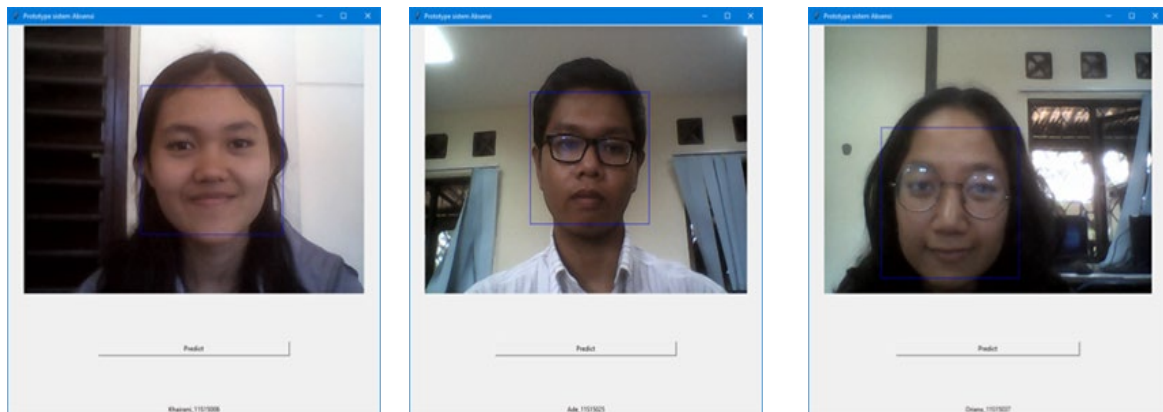
Figure 5. Example of a figure caption and classification result

3.2.3. Result of prototype presence system

After implementing data modeling, we tests the results of the models that have been built. To get the results of evaluations that can be compared and concluded the results, we conducted an object experiment on 9 students. Each student did 5 object experiments. The result of object testing can be seen in Table 4.

Table 4. Result of object testing

Experiment for object no:	Amount of successful detection	Amount of unsuccessful detection
1	5	0
2	5	0
3	5	0
4	5	0
5	5	0
6	4	1
7	5	0
8	5	0
9	4	1
Total	43	2

4.   CONCLUSION

The conclusion obtained from the research that already conducted is the presence system was develop in the form of a prototype using the convolutional neural network (CNN) algorithm by conducting trial experiments on hyperparameters such as the learning rate with a value of 0.0001, epoch with a value of 100, and step per epoch with a value of 150 so this hyperparameter configuration give us a model with accuracy of 99 %. After that we built a presence system prototype by using a graphical user interface (GUI) which is provided by python named tkinter, then we applied the model that has been obtained into the prototype so that the presence system prototype can be used to predict the facial image.

ACKNOWLEDGEMENTS

REFERENCES

[1] H. Suad and V. Asaf, "Real Time Face Recognition System (RTFRS)," *2016 4th International Symposium on Digital Forensic and Security (ISDFS),* 2016, pp. 107–111, doi: 10.1109/ISDFS.2016.7473527.

[2] M. Baykara and R. Das, "Real Time Face Recognition and Tracking System," *International Conference on Electronics, Computer and Computation (ICECCO),* 2013, pp. 159–163, doi: 10.1109/ICECCO.2013.6718253.

[3] N. Bawany, R. Ahmed, and Q. Zakir, "Common Biometric Authentication Techniques: Comparative Analysis, Usability and Possible Issues Evaluation," *Research Journal of Computer and Information Technology Sciences,* vol. 1, no. 4, pp. 5–14, 2013.

[4] L. Cuimei, Q. Zhiliang, J. Nan, and W. Jianhua, "Human face detection algorithm via Haar cascade classifier combined with three additional classifiers," *International Conference on Electronic Measurement and Instruments (ICEMI),* 2017, pp. 483–486, doi: 10.1109/ICEMI.2017.8265863.

[5] M. Belahcene, A. Chouchane, and N. Mokhtari, "2D and 3D Face Recognition Based on IPC Detection and Patch of Interest Regions," *International Conference on Connected Vehicles and Expo (ICCVE),* 2014, pp. 627–628, doi: 10.1109/ICCVE.2014.7297624.

[6] E. Winarno, W. Hadikurniawati, I. H. Al Amin, and M. Sukur, "Anti-Cheating Presence System Based on 3WPCA Dual Vision Face Recognition," *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI),* 2017, pp. 228–232, doi: 10.1109/EECSI.2017.8239115.

[7] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial Examples: Attacks and Defenses for Deep Learning," *International Transactions on Neural Networks and Learning Systems,* vol. 30, no. 9, pp. 2805–2824, 2019, doi: 10.1109/TNNLS.2018.2886017.

[8] X. Zhao and C. Wei, "A Real-Time Face Recognition System Based on the Improved LBPH Algorithm," *International Conference on Signal and Image Processing (ICSIP),* 2017, pp. 72–76, doi: 10.1109/SIPROCESS.2017.8124508.

[9] M. D. Putro, T. B. Adji, and B. Winduratna, "Adult Image Classifiers Based on Face Detection Using Viola-Jones Method," *International Conference on Wireless and Telematics (ICWT),* 2015, pp. 1–6, doi: 10.1109/ICWT.2015.7449208.

[10] E. K. Shimomoto, A. Kimura, and R. Belem, "A Faster Detection Method combining Bayesian and Haar Cascade Classifier," *2005 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON),* 2015, pp. 7–12, doi: 10.1109/Chilecon.2015.7400344.

[11] C. H. Setjo, B. Achmad, and Faridah, "Thermal Image Human Detection Using Haar Cascade Classifier," *International Annual Engineering Seminar (InAES),* 2017, pp. 1–6, doi: 10.1109/INAES.2017.8068554.

[12] D. Zhang, X. Han, and C. Deng, "Review on the Research and Practice of Deep Learning and Reinforcement Learning in Smart Grids," *CSEE Journal of Power and Energy Systems,* vol. 4, no. 3, pp. 362–370, 2018, doi: 10.17775/CSEEJPES.2018.00520.

[13] M. Wang, Z. Wang, and J. Li, "Deep Convolutional Neural Network Applies to Face Recognition in Small and Medium Databases," *International Conference on Systems and Informatics (ICSAI),* 2017, pp. 1368–1372, doi: 10.1109/ICSAI.2017.8248499.

[14] D. Qu, Z. Huang, Z. Gao, Y. Zhao, X. Zhao, and G. Song, "An Automatic System for Smile Recognition Based on CNN and Face Detection," *International Conference on Robotics and Biomimetics (ROBIO),* 2018, pp. 243–247, doi: 10.1109/ROBIO.2018.8665310.

[15] Y. Kewen, H. Shaohui, S. Yaoxian, L. Wei, and F. Neng, "Face Recognition Based on Convolutional Neural Network," National Natural Science Foundation (NNSF), pp. 4077–4081, 2017, doi: 10.23919/ChiCC.2017.8027997.

[16] M. Coskun, A. Ucar, O. Yildirim, and Y. Demir, "Face Recognition Based on Convolutional Neural Network," *International Conference on Modern Electrical and Energy Systems (MEES),* 2017, pp. 376–379, doi: 10.1109/MEES.2017.8248937.

[17] I. Gruber, M. Hlaváč, M. Železný, and A. Karpov, "Facing Face Recognition with ResNet: Round One," in A. Ronzhin, G. Rigoll, and R. Meshcheryakov (eds), *Interactive Collaborative Robotics. ICR 2017. Lecture Notes in Computer Science,* vol. 10459, 2017.

[18] E. Rezenda, G. Ruppert, T. Carvalho, F. Ramos, and P. De Geus, "Malicious Software Classification using Transfer Learning of ResNet-50 Deep Neural Network," *International Conference on Machine Learning and Applications (ICMLA),* 2017, pp. 1011–1014, doi: 10.1109/ICMLA.2017.00-19.

[19] P. Balaprakash, M. Salim, T. D. Uram, V. Vishwanath, and S. M. Wild, "DeepHyper: Asynchronous Hyperparameter Search for Deep Neural Networks," *International Conference on High Performance Computing (HiPC),* 2018, pp. 42–51, doi: 10.1109/HiPC.2018.00014.

[20] Z. Li, L. Jin, C. Yang, and Z. Zhong, "Hyperparameter search for deep convolutional neural network using effect factors," *China Summit and International Conference on Signal and Information Processing,* 2015, pp. 782–786, doi: 10.1109/ChinaSIP.2015.7230511.

[21] K. Yamana *et al.*, "Adaptive Learning Rate Adjustment with Short-Term Pre-Training in Data-Parallel Deep Learning," *International Workshop on Signal Processing Systems (SIPS),* 2018, pp. 100–105, doi: 10.1109/SiPS.2018.8598429.

[22] C. Francois, "Deep Learning with Python," United States of America: Manning Publications Co., 2018.

[23] C. G. Rafael and E. W. Richard, "Digital Image Processing," Third Edition. United States of America: Pearson Education. Inc., 2008.
[24] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A Convolutional Neural Network for Modelling Sentences," *arXiv:1404.2188,* 2014.
[25] V. E. Vyanza, C. Setianingsih, and B. Irawan, "Design of Smart Door System for Live Face Recognition Based on Image Processing using Principal Component Analysis and Template Matching Correlation Methods," *Asia Pacific Conference on Wireless and Mobile (APWiMob),* 2017, pp. 23–29, doi: 10.1109/APWiMob.2017.8283999.

BIOGRAPHIES OF AUTHORS

Yohanssen Pratama, Current Faculty Members & Researcher in Institut Teknologi Del. 4+ years experience specializing in back-end/infrastructure, analytical tools development and computer programming. Teach academic and vocational subjects to undergraduate also pursue my own research to contribute to the wider research activities of my department.

Lit Malem Ginting, Current Faculty Members & Researcher in Insitut Teknologi Del since 2014. He is currently a Senior Lecturer at the Faculty of Informatics and Electrical Engineering, Institut Teknologi Del (IT Del), Laguboti. He dedicates himself to university teaching and conducting research. Her research fields include artificial intelligence, game development, and computer programming.

Emma Hannisa Laurencia Nainggolan, Currently, work as a Quality Assurance at Salt, West Jakarta. Have experience at image processing using neural network, web & desktop developing, UX Design.

Ade Erispra Rismanda, Currently, work as a Software Engineer at Onoff Insurance, Jakarta. Have experience at image processing using neural network, web & desktop developing, microservice system.