

## A smart door prototype with a face recognition capability

Ivan Surya Hutomo<sup>1</sup>, Handy Wicaksono<sup>2</sup>

<sup>1</sup>Department of Electrical Engineering and Computer Science, College of Electrical and Computer Engineering, National Chiao Tung University, Hsinchu, Taiwan

<sup>2</sup>Department of Electrical Engineering, Faculty of Industrial Technology, Petra Christian University, Surabaya, Indonesia

### Article Info

#### Article history:

Received Jun 29, 2020

Revised May 17, 2021

Accepted Jul 23, 2021

#### Keywords:

Digital assistant

Face detection

Face recognition

Raspberry

Smart door lock

### ABSTRACT

This research aimed to integrate a face recognition capability in a smart door prototype. By using a camera-based face recognition, the house owner does not need to make physical contact to open the door. Avoid physical contact is important due to the coronavirus disease 2019 (COVID19) pandemic. Raspberry Pi 3B was used as the main controller, while a servo motor was utilized as a locking door actuator. The program was developed using Node-RED, Blynk, and message queue telemetry transport (MQTT) platforms which are very powerful for developing internet of things (IoT) devices. All of the programs were coded using Python. Haar cascade and local binary pattern histogram methods were implemented on the face recognition stage. Google Assistant integration was done by using Dialogflow and Firebase as Google Cloud services. Integration of face recognition and the smart door was successful. The smart door was unlocked if faces were recognized (average threshold=60%). If a face was not recognized, an email notification containing a face image is sent to the house owner. The Google Assistant could handle user requests successfully with a success rate of 92.8% from 147 trials.

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



### Corresponding Author:

Ivan Surya Hutomo

Department of Electrical Engineering and Computer Science, College of Electrical and Computer Engineering, National Chiao Tung University

No. 1001, Daxue Rd, East District, Hsinchu City, Taiwan 30010

Email: hutomoivan.eic08g@nctu.edu.tw

## 1. INTRODUCTION

Implementation of internet of things (IoT) for smart home nowadays is focused on accessibility and security systems [1], [2]. In the past, people used a mechanic door with a key and padlock, where the door must be opened with a physical key [3]. This system is not so secure because anyone who has the key can enter the door. A house owner also must bring a physical key with him so he could open the door.

Recently, electronic e.g., abstract near field communication (NFC) and biometric authentication techniques are applied to open a door. Biometric methods are the use of biological characteristics of living things that are unique and not identical to one another for security purposes [4]. Biometric methods that are widely applied in the smart door are fingerprint. The integration of face recognition and voice recognition with the smart door is considered to be more efficient and the most natural one than previous methods [5]. By using a camera-based face recognition, the house owner does not need to make physical contact to open the door because it will open automatically when an authoritative face is recognized. Due to the coronavirus disease 2019 (COVID 19) pandemic, it is really important to use a biometrics solution that does not need physical contact.

Previous research, Lim *et al.* [6] developed a face recognition method for the door, but it is not applied to any physical door key, instead, they use a magnetic door lock, so it is still not applicable to all kinds of doors. He also still uses a laptop as the controller which makes his system less portable and did not develop any notification system when there is an unknown person that tries to open the door.

To overcome those disadvantages from previous results, we equip a smart door prototype with face recognition that can be applicable to nearly all kinds of doors. We also add a natural user interface so users can “talk” to a door. A notification system is also provided, Google Assistant service is used to provide clearer information about the door condition. By integrating smart door lock with face recognition and Google Assistant, users are expected to have a more efficient, safe, and interactive way to unlocking the door.

## 2. RESEARCH METHOD

The Raspberry Pi 3B is used as the main controller of the smart door as shown in Figure 1. This includes handling image processing. Raspberry Pi camera V2 is used to capture the image which then is fed to Haar cascade as the face detection algorithm [7]. If a face is detected, the local binary pattern histogram algorithm is then utilized to recognize whose face is in that image. We implement the algorithm using open CV libraries in Python [8].

After the image processing step is done, data is then fed to Node-Red which will send it to a servo motor. This motor controls the door lock. Node-RED will also send data to Firebase and the Blynk application on smartphones so the house owner could get notifications. Dialogflow then retrieves data from Firebase and forwards it to Google Assistant. A user can conduct training phrases that are used to trigger Google Assistant [9]. Raspberry Pi 3 is also used to control the speaker and microphone with Google AIY Voice Kit.

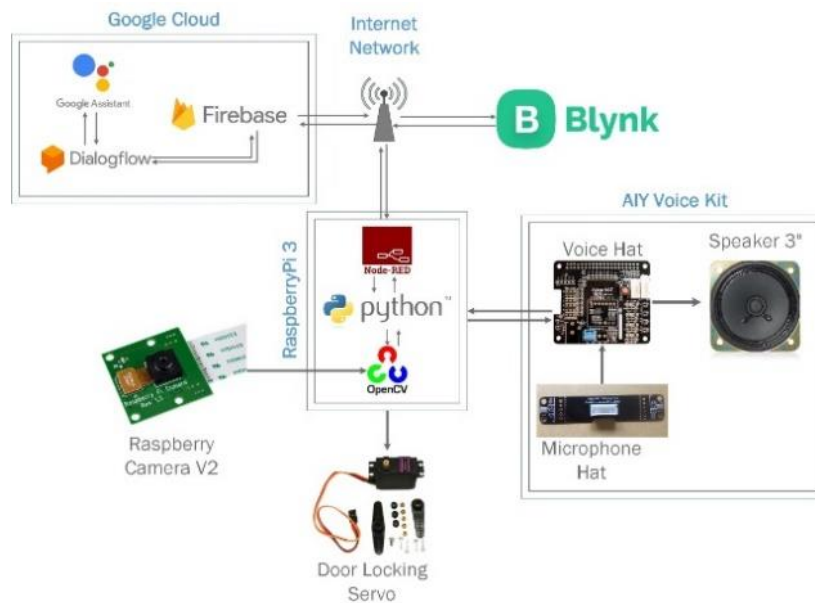


Figure 1. Overall system design

### 2.1. Haar cascade and local binary pattern histogram as face detection and face recognition algorithm

Haar cascade algorithm is used for the face detection process. In general, the Haar like feature is used to detect objects in digital images. The term Haar shows a mathematical function (Haar wavelet) in the form of a box, the principle is the same as in the Fourier function [10]. At first, image processing is only by looking at the red, green, blue (RGB) value of each pixel, but this method found already ineffective [11], [12]. Viola and Jones then developed it so that Haar like features were formed [13].

Face recognition is the next step after face detection. A face can be detected through photos and videos. By utilizing the training results from Haar cascade, the results of this process are combined with the image matching process with the local binary pattern histogram (LBPH) algorithm [14]. With this method, photos that have been learned will be matched with the detection results from streaming cameras. Where in

the latter, some images in the database are then matched with utilizing histogram values that have been extracted from the image by utilizing the LBPH.

To match the owner's face, an equation is used to get the approach of the histogram value which is then used as a predictive value to identify the owner of the face [15]. Therefore, the algorithm output is from the image with the closest histogram. The algorithm must also return the calculated distance, which can be used as a measure of confidence value [16]. Threshold values and confidence can then be used automatically to estimate whether the algorithm has recognized the image correctly. If the confidence value is lower than the threshold, the algorithm has succeeded in recognizing the image.

## 2.2. Hardware design

Previous research used laptops as the main processor of face images which were less portable and less flexible when operated permanently [6]. In that study, it was recommended to replace the laptop platform with a smaller microcontroller that can be applied to various systems, namely Raspberry Pi. Raspberry Pi is a small, less expensive, powerful, and robust microcontroller [17]. We use a Raspberry Pi 3B because it is capable to do face recognition, compatible with Google AIY Voice Kit, and has an easy graphical user interface (GUI) with a Raspbian operating system that can be expanded with various platforms such as OpenCV, Node-RED, message queue telemetry transport (MQTT), and so on [18], [19]. We show our hardware diagram in Figure 2. A power bank 5 V 2 A number 1, is used to supply Raspberry Pi and the servo motor number 9.

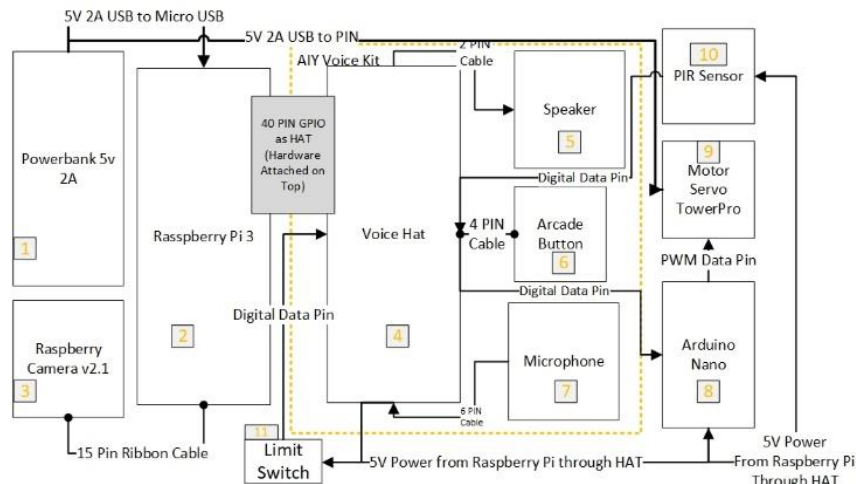


Figure 2. Hardware design diagram

Raspberry Pi camera V2 number 3 is used for face recognition [14]. Voice hardware attached on top (HAT) number 4 will be installed on top of Raspberry Pi 3B which is designed as HAT. Furthermore, the microphone Number 7, speakers Number 5, and the arcade button number 6 will be connected to the Raspberry Pi 3B via HAT. While the HAT installed on the Raspberry Pi 3, general purpose input output (GPIO) on the Raspberry Pi 3B cannot be accessed directly, so accessing GPIO can be done through Voice HAT [20]. GPIO Voice HAT is connected to Arduino number 8 and passive infrared sensor (PIR sensor) number 10.

We use Arduino to overcome the jitter on the servo motor number 9. Jitter is an unstable servo movement caused by a bad pulse-width modulation (PWM) signal [21]. Raspberry Pi 3B will only perform digital triggering on Arduino and subsequently Arduino which will emit PWM signals to change servo angles [22]. PIR sensor is used to detect a person in front of the door, so the face recognition can be started. This method can save memory resource usage and prevent temperature raising in Raspberry Pi 3B. The limit switch number 11 is installed on the door frame to find out whether the door is closed or not. When the door is closed, the servo will lock the door. The smart door prototype can be seen in Figure 3.

## 2.3. Software design

The mechanism of our smart door is shown in Figure 4. When the smart door is initiated initially, if the user does not press a button to add face data, the system will detect a person's existence in front of the

door. If there is a person, the camera will capture and try to recognize his face. If it is recognized, the door will be unlocked, then the door status and the face recognition data are uploaded to Firebase so that the Google Assistant can read them through Firebase. The system then sends an e-mail to the house owner. The authenticated person can open the door. A limit switch is used to detect the door status. If the door is closed, the servo will rotate to lock the door. However, if the face is not recognized, the door cannot be unlocked. Attempts to access the door are recorded in Firebase and a notification email is sent to the house owner to inform these attempts.

If the homeowner wants to add new face data, he must press the capture button and then enter the number that represents the owner's identity (ID), then the camera will take 30 face samples. Because training doesn't need a camera like a capture and recognition process, the process is separate from the main process. The software diagram block of the smart door can be seen in Figure 5.

The whole system is connected to Node RED. Node-RED provides message information received from the face recognition program through MQTT broker to certain paths in firebase [23]. Firebase then will communicate with Dialogflow which is integrated with the Google Assistant service on the user's smartphone [24]. Node-RED communicates with the Blynk using a virtual pin [25]. If Blynk orders to lock the door, it will give a message on an MQTT topic so that Node-RED can control the servo. A face recognition program exchanges data with Node-RED via MQTT brokers. After the face capture is done, 30 images of one face are created and fed to the training process using LBPH to make a pattern histogram for each face. If a face is recognized, then the name at certain Firebase's index is retrieved. The name is converted to a sound file by Google TTS library [26].



Figure 3. Smart door assembled miniature

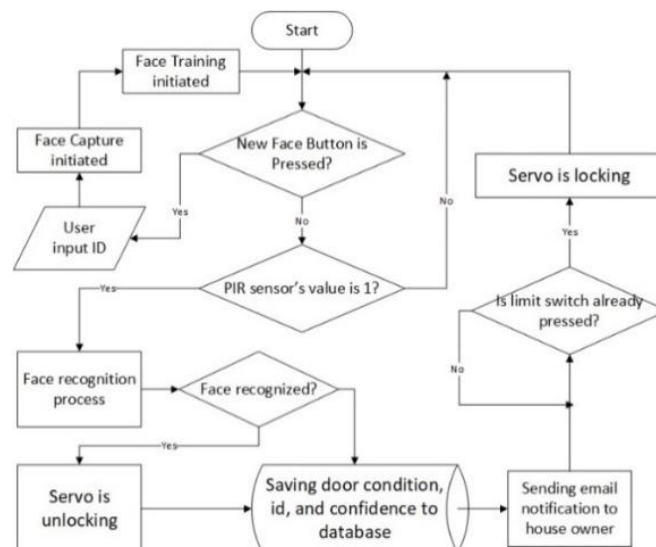


Figure 4. General software flowchart of the smart door

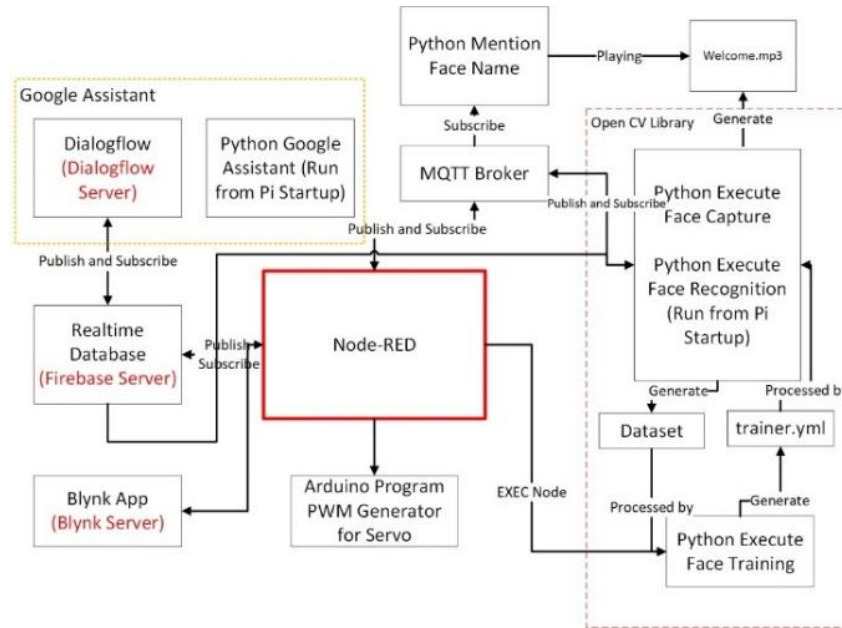


Figure 5. Software relationship block diagram of smart door

### 3. RESULTS AND DISCUSSION

In this chapter, we conduct experiments to find the best camera distance and algorithm parameters (min neighbor and scale factor). Another experiment is done to find the average confidence threshold in the face recognition process.

#### 3.1. Varying camera distance

Based on the experiment, the distance of the camera affects the resolution of the produced dataset. This dataset resolution affects training time, accuracy, and face recognition time as Table 1. The Table 1 shows that the closer the face distance to the camera, the higher the resolution of the dataset. If the data resolution is getting higher, the training time required is longer, but the accuracy of facial recognition increases to 55.03%, and the face recognition time is faster to 2.49 s (compared to the dataset with smaller resolution). The high-resolution dataset enables LBPH to map histograms with more detail, so it can improve accuracy. From the above tests, the best distance is 30 cm.

Table 1. Comparison of various camera distances

Face distance to camera	Dataset resolution	Training time	Accuracy	Face recognition time
45 cm	191x191 px	3.65 s	42.20%	3.031 s
30 cm	320x320 px	9.45 s	55.30%	2.49 s

#### 3.2. Varying the min neighbor and scale factor parameters

Experiments are done to find the most optimal scale factor and min neighbor parameters which produced the best accuracy and the fastest face recognition time. Average accuracy and time were taken from 10x face recognition experiments. In the first test, the min neighbor value is fixed with a value of 3 and the scale factor value is changed. After the most optimal scale factor value is found, the scale factor value is fixed and the min neighbor value is changed. Based on the most optimal parameter simulation for scale factor is 1.5 as Table 2 and for the min neighbor is 2 as Table 3.

At scale factor 1.1, the average time has the worst number of 6.12 s. as Table 2. This is because the reduction process to match the dataset is not large enough so that the reduction occurs repeatedly and requires a longer time to match the dataset. On increasing each value, the time produced is faster and at 1.5 has the best value. In the scale factor, the face recognition scale factor process has increased time because the resulting reduction value is too large so it is unable to match the dataset.

Table 2. Comparison of various scale factor values

scale factor	Min neighbor	Accuracy average (%)	Time average (s)
1	3		Error
1.1	3	46.93	6.1227
1.2	3	41.76	3.973
1.3	3	50.97	3.111
1.4	3	44.08	2.423
1.5	3	58.39	2.059
1.6	3	31.537	2.204
1.7	3	55.52	2.38
1.8	3	26.29	5.458
1.9	3		Error

Table 3. Comparison of various min neighbor values

Scale factor	Min neighbor	Accuracy average (%)	Time accuracy (s)
1.5	1	42.032	2.0484
1.5	2	62.04	2.093
1.5	3	59.06	2.081
1.5	4	51.71	2.118
1.5	5	47.827	2.0874
1.5	6	57.12	2.105
1.5	7	58.11	2.024
1.5	8	50.71	2.079
1.5	9	46.18	2.171
1.5	10	58.47	6.037

Based on simulation results as shown in Table 3, the best min neighbor value is 2, where the average face recognition accuracy can reach 62.04% with an average time of 2.1 s. Min neighbor does not affect the accuracy and speed of face recognition. This happens because the face is close to the camera (30 cm) and the face has filled the entire camera frame so that the possibility of a false positive is very small and does not affect accuracy. However, with the smaller min neighbor value, the time to recognize faces is faster even though it does not change significantly Table 3, this is because when min neighbor is smaller, the system will be more sensitive in recognizing faces so that the time to recognize faces is faster.

### 3.3. Finding average confidence as a threshold for unlocking smart doors

This test aims to determine the ability of the system to recognize faces. In this test, the dataset only consists of 3 registered respondents, while another respondent will not be registered in the dataset and the system should not be given authority to open the door lock (not recognized), see Table 4 for the detail.

From the above experiments, it was found that all faces are recognized. In the first test, Jischak was not captured in the dataset, but his face was recognized as Michael (44.18%), while Michael's original face has 73.63% confidence. In the second one, the faces of Gavriel and Jischak both were identified as Michael with false confidence -55.34% and 50.61%-versus Michael's original face (70.71%). In the third test, only Ivan's face was captured in the dataset, however, the faces of Michael, Gavriel, and Jischak were identified as Ivan with false confidence -50.78%, 53.41%, and 51.57% while Ivan's face had true confidence of 62.59%.

Through the three tests, each true confidence and false confidence were calculated on average and produced average false confidence of 50.98% and true confidence of 67.19%. From these results, to prevent an unknown face from opening a locked door, we determine the confidence threshold to open a door must be above 50.98% (we determine that the average confidence to open the door is 60%). Figure 6 shows that the original face that was identified as Ivan with an average confidence of up to 70%, and the unknown face recognized as Ivan with a much lower average confidence (33%).

From all the experiments we got the threshold of face recognition at 60%. This threshold actually could be improved up to more than 90% if we use neural network methods such as face net. Face net is known as the state of the art neural network for face recognition [27]. Since we integrate this on Raspberry Pi 3B with so many processes such as MQTT and Node-RED, it is not possible to use neural networks due to performance limitation issues, hence we just use LBPH (machine learning method) to recognize the face. It is possible to increase the performance of face recognition if we use neural stick processor as a neural processing helper or use a more powerful Raspberry Pi series such as Raspberry Pi 4 that already has 64 Bit processor architecture.

**Table 4. The unknown dataset to find unlocking threshold**

Dataset registered face	Names	Accuracy average (%)	Time accuracy (s)
Ivan, Michael, Gavrielle	Ivan	65.34	3.876
	Michael	73.63	3.972
	Gavriel	67.48	4.047
	Jischak	44.18	5.252
Ivan, Michael	Ivan	63.39	3.474
	Michael	70.711	3.477
	Gavriel	55.34	3.496
	Jischak	50.61	3.388
Ivan	Ivan	62.59	2.642
	Michael	50.78	3.171
	Gavriel	53.41	3.019
	Jischak	51.57	2.983
False confidence average		50.98	
True confidence average		67.19	



Figure 6. Differences in average confidence in known datasets (left) and unknown datasets (right)

**3.4. Google assistant integration experiments**

Several experiments are also carried out to test the success rate of Google Assistant in responding to orders from users. The success rate can be seen in the session flow from Dialogflow as shown in Figure 7. Based on the session flow Figure 7, from 147 experiments conducted, the success rate reached 93.2%. Google Assistant's failure to recognize user speech is only 6.8% (10 trials out of 147). This whole test shows that Dialogflow's integration with Firebase, Node-RED, and the author's Google Assistant account has been successful.

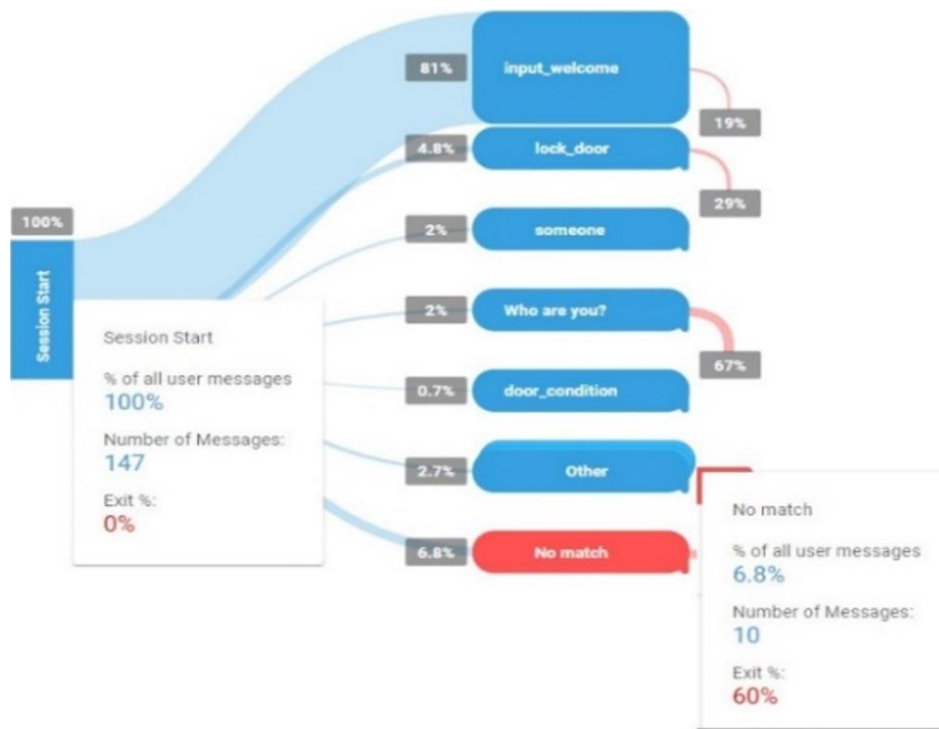


Figure 7. Session flow that shown the success rate of Google Assistant

#### 4. CONCLUSION

We design and create a smart door prototype integrated with face recognition and Google Assistant. After conducting some experiments, the following conclusions are obtained that: Face recognition using Haar cascade face detection and LBPH face recognition was successfully integrated on the smart door with these parameters: the camera distance=30 cm, min neighbor=2, and scale factor=1.5. The number of effective faces stored in the database is 3 faces. If the average confidence of the face is more than 60% the smart door will unlock successfully.

The integration of Google Assistant with the smart door works well. Google Assistant can be used to retrieve information, control smart door devices by utilizing Dialogflow and the real-time database from Firebase. Based on the simulation, it can be concluded that Dialogflow and Firebase are reliable enough to be utilized and integrated with smart doors with a success rate of response reaching 93.2%.

#### REFERENCES




- [1] W. Ali, G. Dustgeer, M. Awais, and M. A. Shah, "IoT based smart home: security challenges, security requirements and solutions," in *2017 23rd International Conference on Automation and Computing (ICAC)*, Sep. 2017, pp. 1–6, doi: 10.23919/ICAC.2017.8082057.
- [2] J. Bugeja, A. Jacobsson, and P. Davidsson, "On privacy and security challenges in smart connected homes," in *Proceedings - 2016 European Intelligence and Security Informatics Conference, EISIC 2016*, Aug. 2017, pp. 172–175, doi: 10.1109/EISIC.2016.044.
- [3] J. I. Jeong, "A study on the IoT based smart door lock system," in *Lecture Notes in Electrical Engineering*, vol. 376, Springer Verlag, 2016, pp. 1307–1318.
- [4] M. Boatwright and X. Luo, "What do we know about biometrics authentication?," in *Proceedings of the 4th annual conference on Information security curriculum development - InfoSecCD '07*, 2007, p. 1, doi: 10.1145/1409908.1409942.
- [5] H. K. Ekenel, J. Stallkamp, H. Gao, M. Fischer, and R. Stiefelhagen, "Face recognition for smart interactions," in *Multimedia and Expo, 2007 IEEE International Conference on*, Jul. 2007, pp. 1007–1010, doi: 10.1109/ICME.2007.4284823.
- [6] R. Lim, F. Rotinsuluand, and P. Santoso, "Room access control system using facial image recognition," *Applied Mechanics and Materials*, vol. 815, pp. 398–402, Nov. 2015, doi: 10.4028/www.scientific.net/amm.815.398.
- [7] R. Szabo and A. Gontean, "Industrial robotic automation with Raspberry PI using image processing," in *2016 International Conference on Applied Electronics (AE)*, Sep. 2016, pp. 265–268, doi: 10.1109/AE.2016.7577287.
- [8] J. Howse, "Training detectors and recognizers in Python and OpenCV," in *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Sep. 2014, pp. 1–2, doi: 10.1109/ismar.2014.6948516.
- [9] N. Rosruen and T. Samanchuen, "Chatbot utilization for medical consultant system," in *2018 3rd Technology Innovation Management and Engineering Science International Conference (TIMES-ICON)*, Dec. 2018, pp. 1–5, doi: 10.1109/TIMES-ICON.2018.8621678.
- [10] P. I. Wilson and J. Fernandez, "Facial feature detection using haar classifiers," *Journal of Computing Sciences in Colleges*, vol. 21, no. 4, pp. 127–133, 2006.
- [11] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, vol. 1, pp. 1–511–I–518, doi: 10.1109/cvpr.2001.990517.
- [12] Y.-Q. Wang, "An analysis of the Viola-Jones face detection algorithm," *Image Processing On Line*, vol. 4, pp. 128–148, Jun. 2014, doi: 10.5201/ipol.2014.104.
- [13] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004, doi: 10.1023/B:VISI.0000013087.49260.fb.
- [14] I. M. Sayem and M. S. Chowdhury, "Integrating face recognition security system with the internet of things," in *Proceedings - International Conference on Machine Learning and Data Engineering, iCMLDE 2018*, Dec. 2019, pp. 19–21, doi: 10.1109/iCMLDE.2018.00013.
- [15] A. Ahmed, J. Guo, F. Ali, F. Deeba, and A. Ahmed, "LBPH based improved face recognition at low resolution," in *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, May 2018, pp. 144–147, doi: 10.1109/ICAIBD.2018.8396183.
- [16] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006, doi: 10.1109/TPAMI.2006.244.
- [17] M. Maksimović, V. Vujović, N. Davidović, V. Milošević, and B. Perišić, "Raspberry Pi as Internet of Things hardware: Performances and Constraints," in *Proceedings of 1st International Conference on Electrical, Electronic and Computing Engineering IcETRAN 2014*, 2014, vol. 3, no. JUNE, p. 8.
- [18] M. Lekić and G. Gardašević, "IoT sensor integration to Node-RED platform," in *2018 17th International Symposium on INFOTEH-JAHORINA, INFOTEH 2018 - Proceedings*, Mar. 2018, vol. 2018-January, pp. 1–5, doi: 10.1109/INFOTEH.2018.8345544.
- [19] Y. Upadhyay, A. Borole, and D. Dileepan, "MQTT based secured home automation system," in *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*, Mar. 2016, pp. 1–4, doi: 10.1109/CDAN.2016.7570945.
- [20] S. Mischie, L. Matiu-lovan, and G. Gasparese, "Implementation of Google Assistant on Raspberry Pi," in *2018 International Symposium on Electronics and Telecommunications (ISETC)*, Nov. 2018, pp. 1–4, doi: 10.1109/ISETC.2018.8583899.
- [21] R. L. Hovious, "Jitter in instrument servos," *Transactions of the American Institute of Electrical Engineers, Part II: Applications and Industry*, vol. 73, no. 6, pp. 393–398, Jul. 1955, doi: 10.1109/TAI.1955.6367088.
- [22] A. S. Sadun, J. Jalani, and J. A. Sukor, "A comparative study on the position control method of dc servo motor with position feedback by using arduino," *ARNP Journal of Engineering and Applied Sciences*, vol. 11, no. 18, pp. 10954–10958, 2016.
- [23] W. J. Li, C. Yen, Y. S. Lin, S. C. Tung, and S. M. Huang, "Just IoT Internet of Things based on the Firebase real-time database," in *Proceedings - 2018 IEEE International Conference on Smart Manufacturing, Industrial and Logistics Engineering, SMILE 2018*, Feb. 2018, vol. 2018-Janua, pp. 43–47, doi: 10.1109/SMILE.2018.8353979.






- [24] J. Dalton, V. Ajayi, and R. Main, "Vote goat," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, Jun. 2018, pp. 1285–1288, doi: 10.1145/3209978.3210168.
- [25] S. Jiyal and R. Kumar Saini, "A survey on air pollution monitoring using internet of things," *International Journal of Control and Automation*, vol. 13, no. 2, pp. 137–146, 2020.
- [26] H. U. Zaman, S. Mahmood, S. Hossain, and I. I. Shovon, "Python based portable virtual text reader," in *2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA)*, Oct. 2018, pp. 1–6, doi: 10.1109/ICACCAF.2018.8776778.
- [27] S. Balaban, "Deep learning and face recognition: the state of the art," in *Biometric and Surveillance Technology for Human and Activity Identification XII*, May 2015, vol. 9457, p. 94570B, doi: 10.1117/12.2181526.

## BIOGRAPHIES OF AUTHORS



**Ivan Surya Hutomo**    received his bachelor's degree in Electrical Engineering Department at Petra Christian University, Indonesia in 2019. He then continues his study of a master's degree in Electrical Engineering and Computer Science at National Chiao Tung University, Taiwan. His field of interest includes Artificial Intelligence, Natural Language Processing, robotics, and automation. He can be contacted at email: hutomoivan.ee08@nycu.edu.tw.



**Handy Wicaksono**    is a senior lecturer in Electrical Engineering Department, Petra Christian University, Indonesia. He received his bachelor's and master's degree from Electrical Engineering Department (Institut Teknologi Sepuluh Nopember), and a Ph.D. degree from School of Computer Science and Engineering, UNSW Australia. His research interests include artificial intelligence, intelligent robot, and industrial automation. He can be contacted at email: handy@petra.ac.id.