# Design of Smart Door Lock System Using Face Recognition and Mask Detection Based on Viola-Jones Algorithm with Android Integration

**Tania Nurindini[1], Mitodius Nicho Swacaesar[2], Rathalia Widita Astika[3], Hendrik Purwanto[4], Rieke Adriati Wijayanti[5], Mochammad Taufik[6]**

[1,2,3,4,5] Digital Telecommunication Network Study Program,
Department of Electrical Engineering, State Polytechnic of Malang, Malang City, 65141, Indonesia
[6] Telecommunication Engineering Study Program,
Department of Electrical Engineering, State Polytechnic of Malang, Malang City, 65141, Indonesia

[1] nurindini@gmail.com, [2] mito.nicho@gmail.com, [3] thalia.widita@gmail.com, [4] hendrikpurwanto29@gmail.com,
[5] riekeaw@polinema.ac.id, [6] moch.taufik@polinema.ac.id

***Abstract*— Indonesia has been affected by COVID-19 since 2020. In order to stop the virus spreading, the government puts Large Scale Social Restrictions. There is no denying that during this pandemic, criminality is increasing, and this is due to nothing other than the need to survive. The stealing and invading of homes is one of the crimes that have increased. Additionally, the government has strengthened health procedures to require that individuals wear masks while doing activities outside. A system that can identify faces and identify the use of masks was developed in order to solve this issue. Face recognition is used to increase security, and mask detection systems work to enforce the wearing of masks when engaging in outside activities. Both of these technologies can be combined into a single smart door lock system for the home. The system will apply facial recognition identification when someone wants to enter the house and mask detection when they want to do activities outside the house. As a monitoring tool, data will be delivered in the form of an image via the user's Android. This is proven by the accuracy rates of facial identification of 95% and mask detection of 97.5% in this research.**

**Keywords— *Face Recognition, Internet of Things, Mask Detection, Smart Door Lock, Viola-Jones***

## I. INTRODUCTION

The COVID-19 virus is greatly widespread, accompanied by an increasing number of positive cases and deaths [1]. The transmission of the COVID-19 virus in Indonesia has entered the smallest unit in society, namely the family. A family cluster is a spread of the coronavirus that comes from family members or people who live in one house [2]. One way to prevent the transmission of the COVID-19 virus is to wear a mask [3]- [4].

The COVID-19 pandemic is also increasing crime rates including stealing, which is quite worrying for the neighborhood. In order to finish their action, thieves will search for openings and attempt various door-breaking methods. Technology is required to overcome the vulnerability of locks, such as door padlocks which are used to secure homes [5].

Face recognition is one of the technologies being developed right now. Face recognition is a technology that is able to recognize faces and can be applied to existing technology. Humans regularly carry out a variety of tasks the traditional way, such as locking doors using traditional locks. The disadvantages of traditional security include the need for a physical key that must be carried at all times. The door cannot be opened if the keyholder forgets to bring it or if the key is broken. By applying face recognition, a person can enter without a key or access card by using face recognition because their face has already been registered in the database [6].

In research conducted by M. Arsal et al, a security system for bank employee's access doors was created using face recognition, they are using machine learning named deep learning and CNN (Convolutional Neural Network) algorithm [7]. Other research that conducted by Baay et al, a designed system incorporates deep learning models, face detector, tracking, and counting programs become an automated system assisted by Graphic Users Interface (GUI) with an alarm device. This system detect the use of masks [8].

The mask detection system by Kamale Usha et al, uses the Viola-Jones algorithm to detect whether a person is using a mask or not and the correct position of the mask or not. This system helps in screening so that it can provide security at work or in public places [9]. Another example, a system to monitor the use of masks with the intention of enforcing COVID-19 health protocols by Fathul A. L., et al. If a user is not wearing a mask, the system has an auditory feature that acts as a warning. For the purpose of detecting oral and facial objects, the Haar Cascade approach is employed. An alarm and text will sound when the mouth and face are detected. In contrast, a text warning will appear as a notice that a mask was worn when the face is identified but the mouth is not [10]. Another research by

Samuel Ady et al, the detection of the use of masks that can be used for mitigation, evaluation, and prevention of COVID-19. This mask detection system was created by utilizing machine learning algorithms, namely MobileNetV2 [11].

A security system by Indra Darma Wijaya et al, by utilizing a Raspberry Pi as a processor and usb webcam as a face detection tool which will then be processed by Raspberry Pi using OpenCV to determine whether a human face or not, then the face will enter the facial recognition process with the triangle face method which utilizes the calculation of the distance between facial features such as eyes, nose and mouth. After the face is recognized, Raspberry pi will perform a command on the servo to open the door of the space [12]. A smart door system with facial recognition identification by Sandesh R et al. The system will capture an image, then detect the presence of a face, and recognize the face to access the entrance. This system uses OpenCV for the facial recognition identification process and the Eigen Face algorithm. This system works using the Raspberry Pi as its main processor [13].

From the previous research by Ilyas Novansyah, a system that can recognize user was developed for security with telegram integration [14]. In this study, a security system prototype was made, using a Raspberry Pi as the control center for the physical security system, and will be integrated with Android system. Also from the previous research that conducted by Janhavi Baikerikar, an affordable home security system is capable of detecting intruders, when there is an intruder coming in, they will be given an alarm warning [15].

For improvement, in this study, will have both face recognition and facemask detection. The system works when someone tries to open the door without authorization, the Raspberry Pi will take pictures of them and send them via Firebase to the user's Android smartphone. The door will be open, but the user must put on a mask before leaving. The Viola-Jones method will be used to process the webcam image, because this algorithm has the advantages of a high detection rate for tracking faces in images with a low error rate.

## II. METHOD

### A. System Diagram

Fig. 1 depicts a system overview that describes the hardware process of the system.

The facial detection system must be activated by the keyholder or user before entering by pressing the sign-in button. The face is then directed at the door's webcam after that. The face image will be captured by the webcam and analyzed by the Raspberry Pi using the Viola-Jones method. If the face has been successfully detected and identified, then the Raspberry Pi will send a command to open the solenoid padlock through the relay and turn on the buzzer. When a face can be spotted but not recognized or identified, the camera will save a webcam image of the face and send it to Firebase. Next, a notification and photo obtained from Firebase will be sent to the smartphone that holds the key.

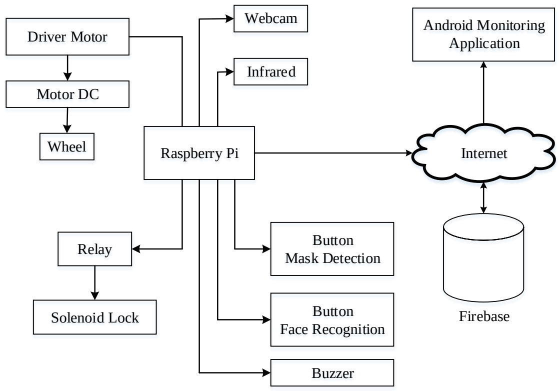



Figure 1. System Diagram

The mask detection system is designed for key owners who are going out of the house. First, they had to press the button to activate the face detection system. After that, the face is directed to the webcam attached to the door. The face image will be captured by the webcam and analyzed by the Raspberry Pi using the Viola-Jones method. The Raspberry Pi will send a command to open the solenoid padlock through the relay and activate the buzzer if the mask has been successfully detected. And, if the mask cannot be identified, it is required to press the button again to re-detect it.

### B. Data Flow Diagram

DFD, or the design of an information flow system between hardware systems, databases, and software, is depicted in Fig. 2.

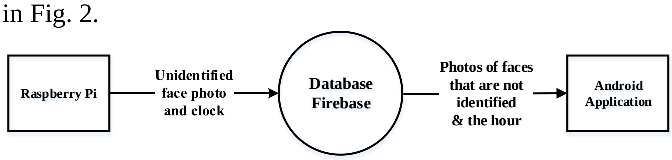



Figure 2. Data Flow Diagram

The hardware system will send information in the form of an unrecognized facial image along with the time. Then, the Raspberry Pi microcontroller will send the info to be stored in Firebase. Furthermore, the data stored in the firebase will be sent directly to the monitoring application.

### C. Flowchart Viola-Jones Algorithm

Fig. 3 depicts the flowchart of Viola-Jones Algorithm process.

In the Viola-Jones algorithm, grayscale images will be scanned per-sub-window to look for positive features with AdaBoost and Cascade Classifier. When a face is detected, a square depiction will then be added to the image of the face. From each square depiction, it inevitably gets a very large total number of Haar Features, exceeding the number of pixels. In order for the classification process to be faster, the training process must eliminate the features that are mostly the same and concentrate on the necessary set of features. AdaBoost's algorithm that has been trained with the Haar Feature is used to separate between square depictions containing positive objects (desired ones) and negative objects (unwanted ones).

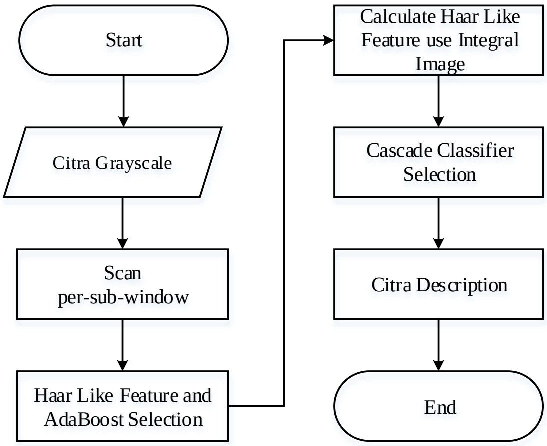



Figure 3. Flowchart of Viola-Jones Algorithm

### *D. System Flowchart Diagram*

Fig. 4 shows the flow of face recognition system performance depicted in the form of a flowchart.

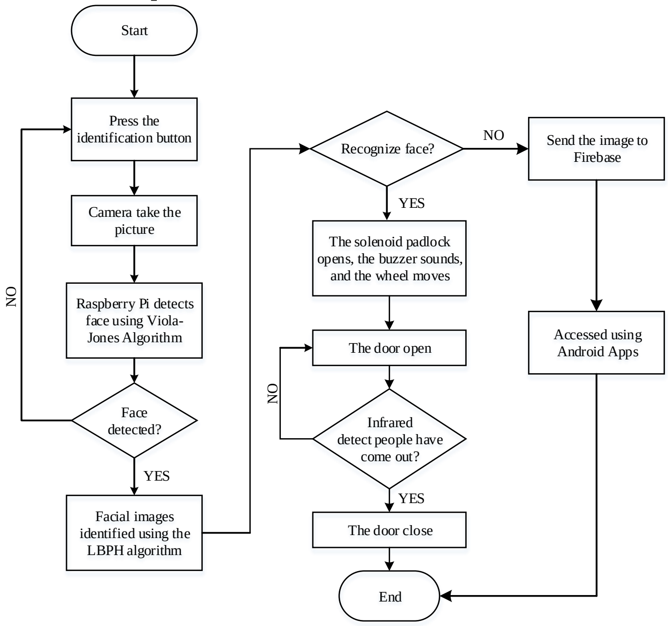



Figure 4. Flowchart of Face Recognition System

The first step when the keyholder wants to enter is to press the face recognized button to turn on the camera. The face will be recognized after the camera has taken a picture of it. If the face is not detected or recognized, the person must press the button once more to repeat the process. If it is recognized, the Raspberry Pi will open the solenoid locks, ring the buzzer, and move the wheels so the door opened. The photograph will be submitted to firebase along with the time it was taken if the face image cannot be recognized. If the face recognition system comes failure three times, there are emergency button system in Android application. This system only enabled when getting three times failure.

Along with face recognition system, there is a facemask detection system. This system detect mask on user's face who want to go outside. This system prevents people from forgetting the use of mask. Fig. 5 shows the flow of mask detection system performance depicted in the form of a flowchart.

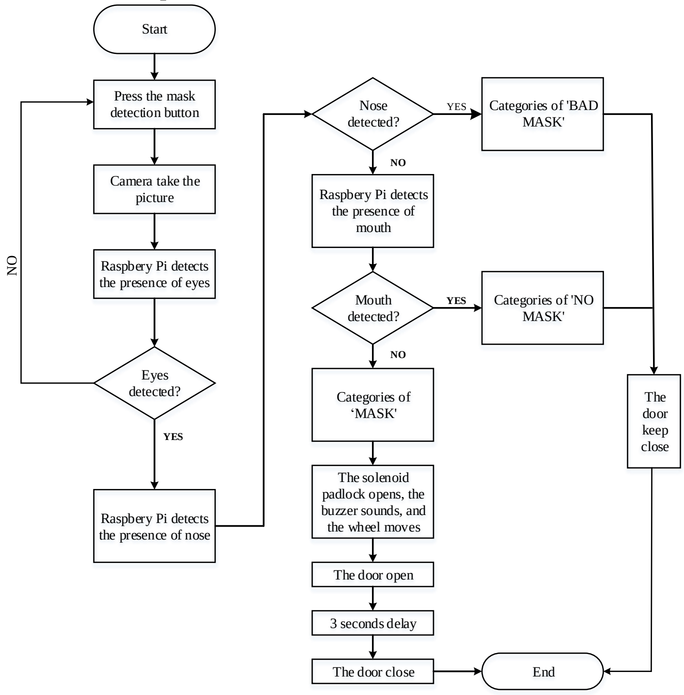



Figure 5. Flowchart of Mask Detection System

When someone is willing to leave, they only need to press the mask detection button to turn on the system, and the camera will take a picture of their face. There are three types of output, they are "mask", "no mask", and "bad mask". Output "mask" occurs when eyes is detected, but nose and mouth are not detected. Output "no mask" occurs when eyes, mouth, and nose are detected. When the eyes and nose are detected but the mouth is not, the output "bad mask" occurs. The door will open only if the output is "mask," and after a 3-second delay, the door will be closed automatically.

## III. RESULT AND DISCUSSION

### *A. Prototype Implementation Result*

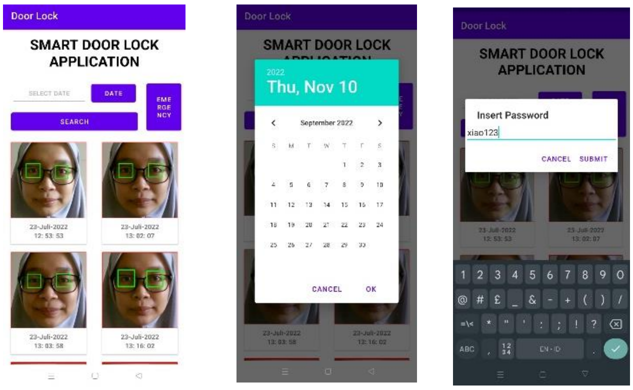
The prototype of the door house has dimensions of 85 cm in height, 60 cm in width, and 60 cm in length. The first system box is placed on the bottom left side. This box contains a Raspberry Pi, a buzzer, and a relay. The second system box is placed above the door. This box contains the motor driver. The door is then given a wheel on the lower left side so that it can be moved without being pushed. This wheel is driven by a DC motor. In order to automatically close the door, once someone enters, an infrared sensor is positioned to detect them. A solenoid door lock is mounted on the left side of the backdoor frame, As shown in Fig. 6.

(a) (b) (c)

Figure 6. Prototype Implementation (a) Front View (b) Back View (c) Side View

## B. *Software Implementation Result*

The information below provides documentation of the software implementation in the form of an Android application in this house door prototype. This software was created using Android Studio software, and this app is only made for homeowners, as shown in Fig. 7.




(a) (b) (c)

Figure 7. Smart Door Lock Application (a) Main Page (b) Date Feature (c) Emergency Feature

Fig. 7 (a) shows the main page of this application contains the 'SEARCH', 'DATE', and 'EMERGENCY' buttons. Images are also displayed by date and time. Fig. 7 (b) shows the application also provides a filter feature by date, this feature can be accessed when pressing the 'DATE' button. Fig. 7 (c) shows Password dialog alert feature, this feature will appear when the 'EMERGENCY' button can be pressed.

## C. *Image Processing of Face Recognition*

Fig. 8 shows preprocessing starts from the process of compiling a dataset in the form of images that are divided into several folders and named after the usernames. In preprocessing, the Viola-Jones algorithm is used to detect the presence of faces in the image. After the presence of the face is detected, the crop process will be carried out on the face only. Then, the results of the cropped image will go through a resizing process with an image size of 64 x 64 pixels. Next, the image will be converted to a grayscale image, as shown in Fig. 9.

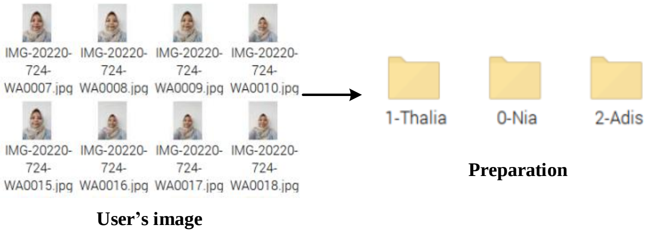



Figure 8. Dataset Preparation

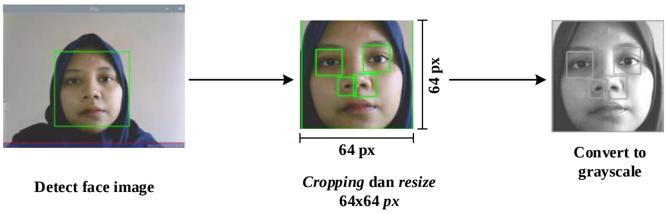



Figure 9. Preprocessing

Furthermore, the training process is intended to train the system to be able to recognize faces in the training data folder, the LBPH (Local Binary Pattern Histogram) algorithm is used in this process. The amount of data used in this training process amounts to 150 images per person. The following is the result of training data stored in .yml format, as shown in Fig. 10.

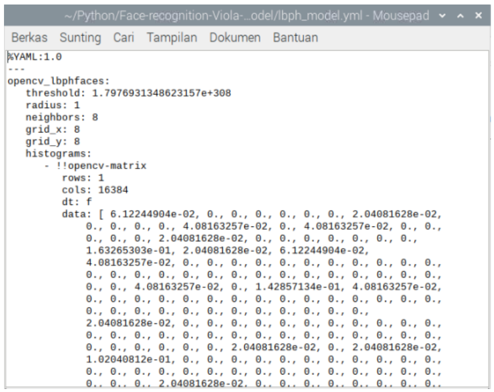

~/Python/Face-recognition-Viola-...odel/lbph_model.yml - Mousepad
Berkas Sunting Cari Tampilan Dokumen Bantuan
```
%YAML:1.0
---
opencv_lbphfaces:
   threshold: 1.7976931348623157e+308
   radius: 1
   neighbors: 8
   grid_x: 8
   grid_y: 8
   histograms:
      - !!opencv-matrix
        rows: 1
        cols: 16384
        dt: f
        data: [ 6.12244904e-02, 0., 0., 0., 0., 0., 0., 2.04081628e-02,
            0., 0., 0., 0., 4.08163257e-02, 0., 4.08163257e-02, 0., 0.,
            0., 0., 0., 2.04081628e-02, 0., 0., 0., 0., 0., 0., 0.,
            1.63265303e-01, 2.04081628e-02, 6.12244904e-02,
            4.08163257e-02, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
            0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
            0., 0., 4.08163257e-02, 0., 1.42857134e-01, 4.08163257e-02,
            0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
            0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
            2.04081628e-02, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
            0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
            0., 0., 0., 0., 0., 0., 2.04081628e-02, 0., 2.04081628e-02,
            1.02040812e-01, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
            0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
            0., 0., 0., 2.04081628e-02, 0., 0., 0., 0., 0., 0., 0., 0.,
```


Figure 10. Training Data Result

The examining process or testing process is the final stage of the facial image recognition process, as shown in Fig. 10. The testing process aims to find out whether the system is able to distinguish the faces of users and strangers. The Viola-Jones and LBPH algorithms are used in this process. Fig. 11 shows the output of this process is a face image and a caption containing the name of a user or unknown person.

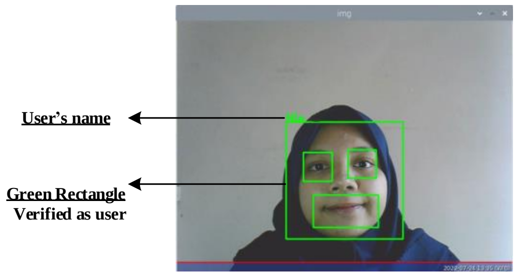



Figure 11. Face Recognition Examining Output

### *D. Image Processing of Facemask Detection*

At the preprocessing and training stages, model data is collected in the form of a Haar Cascade file. The model data is obtained from an open-source site, so it can be used freely. The required model data is in the form of eye, nose, and mouth model data. This model data is in the form of a file in xml format, as shown in Fig. 12.

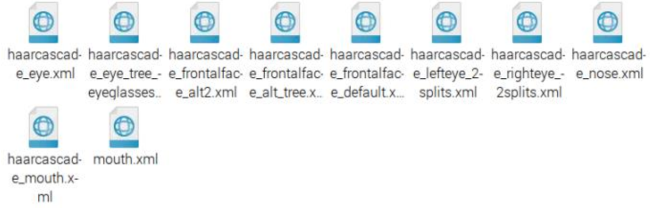



Figure 12. Haar Cascade Models

Fig. 13 shows the examining process aims to find out whether the system is able to detect the use of appropriate and inappropriate masks, and not to use masks. The output of this process is a face image and a caption containing the conditions of wearing the mask.

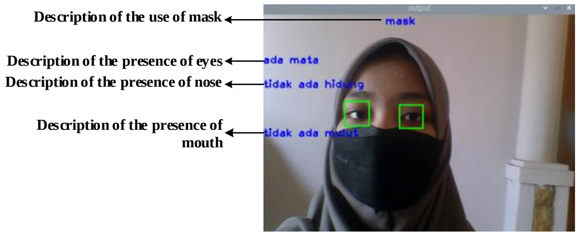



Figure 13. Facemask Detection Examining Output

### *E. System Testing for Face Recognition*

In the facial recognition test, facial recognition accuracy tests will be conducted at various distances, including 30 cm, 50 cm, and 80 cm in two separate time periods, namely 12.00 - 15.00 and 16.00 - 18.00. This accuracy calculation will use a confusion matrix as shown in Table I.

TABLE I
FACE RECOGNITION CONFUSION MATRIX

| Category | Description |
|---|---|
| True Positive (TP) | Faces are listed in the trainer data and identified. |
| True Negative (TN) | Faces are not listed in the training data and not identified. |
| False Positive (FP) | The face is not listed in the training data and identified; Faces are identified as other users' faces. |
| False Negative (FN) | Faces are listed in the training data, not identified. |

The Table II shows the results of facial recognition identification.

TABLE II
FACE RECOGNITION RESULT

| **Image** |  |  |  |
|---|---|---|---|
| **Name** | Nia | Thalia | Adis |

Following testing and the collection of 180 data items from three participants in the training data, the accuracy calculation was carried out, and the following results as shown in Table III.

TABLE III
FACE RECOGNITION ACCURACY RESULT

| No. | Distance | Time | Accuracy |
|---|---|---|---|
| 1. | 30 cm | 12.00 – 15.00 | 100% |
| 2. | 50 cm | | 100% |
| 3. | 80 cm | | 96,6% |
| 4. | 30 cm | 16.00 – 18.00 | 96,6% |
| 5. | 50 cm | | 76,6% |
| 6. | 80 cm | | 73,3% |

A room with windows and ventilation was used for testing from 12.00-15.00 in order to acquire good lighting. Each subject performed the experiment ten times at three different distances of 30 cm, 50 cm, and 80 cm. Following the experiment, it is also known that the accuracy percentage value at each distance reached a perfect value of 100%. However, at an 80 cm distance, the accuracy value dropped by 96.6%

Additionally, testing was done in the same room between the hours of 16.00-18.00, when the lighting was beginning to dim. Each subject performed the experiment ten times at three different distances: 30 cm, 50 cm, and 80 cm. At a distance of 30 cm, the accuracy value was 96.6%, at a distance of 50 cm, it was 76.6%, and at a distance of 80 cm, it was 73.3%. After doing the experiment, it was obvious that the accuracy value was starting to decrease.

The results of the 60 tests conducted on the representation of outsider are shown Table IV.

TABLE IV
UNTRAINED FACIAL RECOGNITION TESTING RESULT

| No. | Image | Time | Distance | Accuracy |
|---|---|---|---|---|
| 1 |  | 12.00 | 30 cm | 100% |
| 2 | | - | 50 cm | 100% |
| 3 | | 15.00 | 80 cm | 100% |
| 4 | | 16.00 | 30 cm | 100% |
| 5 | | - | 50 cm | 100% |
| 6 | | 18.00 | 80 cm | 100% |

The experiment was done from 12 a.m. to 15 a.m. in a room with windows and ventilation to get a good lighting. The experiment was run at distances of 30 cm, 50 cm, and 80 cm. Following the experiment, it was discovered that the accuracy percentage value at each distance was highly accurate, reaching 100%.

Later, testing was done in the same room from 16:30 to 18:30 with lighting that was starting to decrease. The experiment was repeated ten times at distances of 30 cm, 50 cm, and 80 cm, respectively. Following the trial, it was discovered that each value obtained was extremely accurate, reaching 100%.

After testing, it can be seen that the accuracy of facial identification is known by the distance and intensity of the light. This can be seen in the table, when the distance is getting farther and the light received begins to dim, the degree of accuracy decreases. But the system is able to recognize both listed and unregistered faces in the database well. It can be concluded that the accuracy is better when the light brighter and the distance closer.

Additionally, facial recognition processing speed testing was done to see how rapidly the system could process facial recognition. With three different distances and two different times, this test was run up to ten times with the correct data. After testing the delay with three distance differences and two time spans, an average delay of 1.563 s was obtained.

### *F. System Testing for Mask Detection*

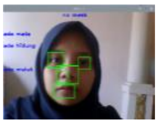
In the mask detection test, tests for mask detection accuracy will be conducted at a variety of distances, namely 30 cm, 50 cm, and 80 cm, and during two different time windows, namely 12.00-15.00 and 16.00-18.00. The categories of mask detection tests that make use of a confusion matrix are as follows Table V.

TABLE V
FACEMASK DETECTION MATRIX CONFUSION

| Category | Description |
|---|---|
| True Positive (TP) | - Detected the usage of a mask on the masked face. |
| True Negative (TN) | - The face is not masked; it is detected that it is not wearing a mask.<br>- The face was masked inappropriately. Incorrect use of masks was detected. |
| False Positive (FP) | - The face is not masked, detecting the use of a mask.<br>- Masked face wrong, detected the use of a mask. |
| False Negative (FN) | - The face is not masked. The wrong use of the mask is detected.<br>- Inappropriately masked face, detected unmasked |

The table VI shows the results of facial recognition identification.

TABLE VI
RESULT OF FACE RECOGNITION

| No | Figure | Condition |
|---|---|---|
| 1 |  | Using Mask |
| 2 |  | Using Mask |
| 3 |  | Bad Mask |
| 4 |  | No Mask |

After conducting the test, the accuracy of the black and white mask detection is calculated and the results are displayed in the Table VII.

TABLE VII
MASK DETECTION ACCURACY RESULT

| No | Mask | Time | Distance | Accuracy |
|---|---|---|---|---|
| 1 | Black | 12.00 - 15.00 | 30 cm | 100% |
| 2 | Black | 12.00 - 15.00 | 50 cm | 90% |
| 3 | Black | 12.00 - 15.00 | 80 cm | 100% |
| 4 | White | 12.00 - 15.00 | 30 cm | 90% |
| 5 | White | 12.00 - 15.00 | 50 cm | 100% |
| 6 | White | 12.00 - 15.00 | 80 cm | 100% |
| 7 | Black | 16.00 - 18.00 | 30 cm | 100% |
| 8 | Black | 16.00 - 18.00 | 50 cm | 100% |
| 9 | Black | 16.00 - 18.00 | 80 cm | 100% |
| 10 | White | 16.00 - 18.00 | 30 cm | 100% |
| 11 | White | 16.00 - 18.00 | 50 cm | 100% |
| 12 | White | 16.00 - 18.00 | 80 cm | 100% |

Following the experiment, it is known that the accuracy percentage value for black and white masks at each distance was perfect, reaching a value of 100 percent. The accuracy rate is 90 percent at a distance of 50 cm when a black and white mask is used, however.

The percentage of detection accuracy of the incorrect use of masks and unmasked faces is then calculated. The results of the calculation of its accuracy can be seen in the Table VIII.

TABLE VIII
BAD MASK AND NO MASK DETECTION ACCURACY

| No | Mask | Time | Distance | Accuracy |
|---|---|---|---|---|
| 1 | Bad Mask | 12.00 - 15.00 | 30 cm | 100% |
| 2 | Bad Mask | 12.00 - 15.00 | 50 cm | 100% |
| 3 | Bad Mask | 12.00 - 15.00 | 80 cm | 100% |
| 4 | No Mask | 12.00 - 15.00 | 30 cm | 100% |
| 5 | No Mask | 12.00 - 15.00 | 50 cm | 100% |
| 6 | No Mask | 12.00 - 15.00 | 80 cm | 90% |
| 7 | Bad Mask | 16.00 - 18.00 | 30 cm | 100% |
| 8 | Bad Mask | 16.00 - 18.00 | 50 cm | 90% |
| 9 | Bad Mask | 16.00 - 18.00 | 80 cm | 90% |
| 10 | No Mask | 16.00 - 18.00 | 30 cm | 100% |
| 11 | No Mask | 16.00 - 18.00 | 50 cm | 90% |
| 12 | No Mask | 16.00 - 18.00 | 80 cm | 100% |

After conducting the experiment, it can be seen that the percentage value of accuracy to detect the wrong use of masks and unmasked faces at each distance reached a perfect value of 100%. However, in some conditions, its accuracy value reaches only 90%. This shows that the system is able to detect the use of masks well. However, the light intensity and the distance between the face and the camera also affect the level of accuracy, besides the stability of the microcontroller also affects the results.

In order to determine how quickly the system could identify the use of masks, a mask detection delay test was also performed. Three different distances and two different times were used in this test, which was performed up to ten times with accurate data. After testing the delay with three distance differences and two time spans, an average delay of 0.894 s was obtained.

### G. *Overall System Testing*

With the use of this comprehensive system test, we can determine whether or not each system function has been carried out correctly and effectively. The Table IX contains the results of the system's overall testing.

TABLE IX
OVERALL SYSTEM TESTING RESULT

| Testing | Activity | Desired Result | Status |
|---|---|---|---|
| Push Button | Activate the camera and take a picture. | Display an image. | Success |
| Webcam (IN) | Identifies the faces listed in the train data. | Displays the name and a green box. | Success |
| | Identify unrecognized faces. | Displays the words "Unknown" and a red box. | Success |
| Webcam (Out) | Detects masked faces. | Displays the inscription "Mask". | Success |
| | Detects non-masked faces | Displays the inscription "No Mask". | Success |
| | Detect faces whose masks are used incorrectly. | Displays the inscription "Bad Mask". | Success |
| Buzzer | Identifies the faces listed in the train data. | The buzzer sounded. | Success |
| | Identify unrecognized faces. | The buzzer did not sound. | Success |
| | Detects masked faces | The buzzer sounded. | Success |
| | Detects non-masked faces. | The buzzer did not sound. | Success |
| | Detecting the wrong use of masks. | The buzzer did not sound. | Success |
| Infrared | Detects the presence of passing objects. | The LED lights up. | Success |
| | Detects the absence of passing objects. | The LED does not light up. | Success |
| Solenoid Door Lock | Identifies the faces listed in the train data. | Solenoid door lock unlocked the door. | Success |
| | Identify unrecognized faces. | Solenoid door lock lock locked the door. | Success |
| | Detects masked faces. | Solenoid door lock unlocked the door. | Success |
| | Detects non-masked faces. | Solenoid door lock lock locked the door. | Success |
| Driver Motor and Motor DC | Solenoid door lock unlocked the door. | The moving wheel opened the door. | Success |
| | Solenoid door lock locked the door. | The wheels do not move. | Success |
| | Infrared sensors detect the presence of objects. | The moving wheel closes the door. | Success |
| | Infrared sensors do not detect the presence of objects. | The wheels do not move. | Success |

It is possible to determine that the system is capable of functioning properly after doing an overall system test.

In order to determine whether the Android App can function properly, the entire application system is tested. To do this test, first open the application and then individually operate each of its components. The results of testing the application completely are listed in Table X.

It may be concluded from the test results shown in Table X that every component of the program can function properly and is as expected.

TABLE X
APPLICATION TESTING RESULT

| Testing | Success | Not Success |
|---|---|---|
| Open the App | ✓ | - |
| Button “DATE” | ✓ | - |
| Selecting date | ✓ | - |
| Button “SEARCH” | ✓ | - |
| Button “EMERGENCY” | ✓ | - |
| Fill the password | ✓ | - |
| Display the Card View | ✓ | - |

## IV. CONCLUSION

The system consists of the Raspberry Pi, a webcam, a push button, a solenoid door lock, a buzzer, an infrared sensor, and an integrated DC motor gearbox. This system includes an android application that provides as the user's interface with the system. The conclusion that each component employed can function properly. From the results of facial recognition testing, it can be concluded that the Viola-Jones algorithm in synergy with LBPH is able to recognize faces with an average accuracy rate of 95.25% and an average process time of 1,563 s. Meanwhile, from the results of mask detection testing, it can be concluded that the Viola-Jones algorithm is able to detect the use of masks with an average accuracy of 97.5% with an average process time of 0.894 s. From the results of testing the application system with various features, it can be concluded that the Android application can run well and as expected.

## REFERENCES

[1] H. Nishiura, N. M. Linton, and A. R. Akhmetzhanov, “Serial interval of novel coronavirus (COVID-19) infections,” Int. J. Infect. Dis., vol. 93, pp. 284–286, 2020.

[2] Wu, Yi-Chia; Chen, Ching-Sunga; Chan, Yu-Jiuna,b,c,*. The outbreak of COVID-19: An overview. Journal of the Chinese Medical Association 83(3):p 217-220, March 2020.

[3] T. Alam and S. Qamar, “Coronavirus Disease (COVID-19): Reviews, Applications, and Current Status,” J. Inform. Univ. Pamulang, vol. 5, no. 3, p. 213, 2020.

[4] D. Chen et al., “Recurrence of positive SARS-CoV-2 RNA in COVID-19: A case report,” Int. J. Infect. Dis., vol. 93, pp. 297–299, 2020.

[5] S. M. Situmeang, “Fenomena Kejahatan Di Masa Pandemi Covid-19: Perspektif Kriminologi,” Maj. Ilm. UNIKOM, vol. 19, no. 1, pp. 35–43, 2021.

[6] E. Fadly, S. A. Wibowo and A. P. Sasmito, "Sistem Keamanan Pintu Kamar Kos Menggunakan Face Recognition Dengan Telegram Sebagai Media Monitoring dan Controlling," *JATI (Jurnal Mahasiswa Teknik Informatika),* vol. 5, no. 2, pp. 435-442, 2021.

[7] M. Arsal, B. A. Wardijono and D. Anggraini, "Face Recognition Untuk Akses Pegawai Bank Menggunakan Deep Learning Dengan Metode CNN," *Jurnal Nasional Teknologi dan Sistem Informasi,* vol. 06, no. 01, pp. 055-063, 2017.

[8] M. N. Baay, A. N. Irfansyah and M. Attamimi, "Sistem Otomatis Pendeteksi Wajah Bermasker Menggunakan Deep Learning," *JURNAL TEKNIK ITS,* vol. 10, no. 1, pp. 64-70, 2021.

[9] K. Usha, B. Sudeepthi, D. Mahathi and P. Shravya, "Face Mask Detection Using Feature Extraction," in *Data Engineering and Communication Technology*, Mumbai, Springer, 2020, pp. 437-447.

[10] F. L. Ahmad, A. Nugroho and A. F. Suni, "Deteksi Pemakai Masker Menggunakan Metode Haar Cascade Sebagai Pencegahan COVID-19," *Edu Elektrika Journal,* vol. 10, no. 1, pp. 13-18, 2021.

[11] S. A. Sanjaya and S. A. Rakhmawan, "Face Mask Detection Using MobileNetV2 in The Era of COVID-19 Pandemic," in *2020 International Conference on Data Analytics for Business and Industry Way Towards a Sustainable Economy (ICDABI)*, IEEE, 2020.

[12] I. D. Wijaya, U. Nurhasan and M. A. Barata, "Implementasi Raspberry Pi Untuk Rancang Bangun Sistem Keamanan Pintu Ruang Server Dengan Pengenalan wajah Menggunakan Metode Triangle Face," *Jurnal Informatika Polinema,* vol. 4, no. 1, pp. 9-16, 2017.

[13] S. R, A. Sridhar, R. T. P, S. Farheen and S. Tameem, "Smart Door Lock/Unlock Using Raspberry Pi," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology,* vol. 6, no. 3, pp. 543-548, 2020.

[14] I. Novansyah, T. B. Utomo and M. Y. Fadhlan, "Realisasi Prototype Sistem Smart Door Lock dengan Pengenalan Wajah Terintregasi Telegram Messenger Berbasis Internet Of Things," in *Prosiding The 12th Industrial Research Workshop and National Seminar*, Bandung, 2021.

[15] J. Baikerikar, V. Kavathekar, Y. Agarwal, S. Bhat, C. Polly and S. Juwatkar, "Home Security System Using Face Recognition," in *Proceedings of the 2nd International Conference on Advanced Computing Technologies and Applications*, Mumbai, SVKM's Dwarkadas J. Shangvi College of Engineering, 2020, pp. 303-310.