Journal of Data Science, Technology, and Artificial Intelligences
ISSN XXXX-XXXX
Vol. 1, No. 1, July, 2024, pp. 33-37
Journal homepage: https://journal.itisd.org/index.php/ditech

Improvement of Face Recognition Algorithm in Smart Home Security System

Abdul Ghofir[1*], Rusdianto Roestam[2] , Insidini Fawwaz[3]

[1] President University, Jababeka Education Park, Cikarang, Bekasi 17530, Indonesia
[2] President University, Jababeka Education Park, Cikarang, Bekasi 17530, Indonesia
[3] Program Studi Teknik Informatika, Universitas Prima Indonesia, Medan 20118, Indonesia

Corresponding Author Email: geoff@president.ac.id

https://doi.org/xx.xxxxx/ditech.xxxxxx

Received : 24 March 2024
Revised : 26 July 2024
Accepted : 28 July 2024
Availabe online : 31 July 2024

Keywords:
Face recognition, euclidean distance, microcontroller, internet of thing, home security system.

ABSTRACT

At present, human intervention is still needed in most security systems to control their functions. The implementation of machine learning plays an important role in smart home security systems for better control functions. The system will have the ability to learn user behaviour, which then represents it in the form of control of the system. One of the important capabilities possessed by a security system is to recognize the face of everyone who accesses a secured place. This paper introduces a face recognition algorithm which is enhanced through a filtration of its controlled Euclidean Distance. The Success Rate Formula is also added and applied for more convincing results. All required system functions are identified and registered as the first system development step. The type of sensor for each function is determined as input data for machine learning processing. Designing and coding the system is carried out on Arduino as its core physical control system before testing and evaluating the system.

1. INTRODUCTION

Security systems for housing are a common need for modern society today. The protective measures implemented are not only limited to intruders trying to enter the house, but also protect all equipment used inside and outside the home (i.e. in the yard). Control of all equipment is carried out using a control system which usually uses a microcontroller connected to a smartphone via the internet. An implementation scheme like this is also called IoT.

Safe action for intruders is carried out through the use of cameras or other sensors such as infrared, ultrasonic, motion detectors. The data captured by the camera is sent to the owner using a smartphone connected to the internet. This is one example of implementing a security system for residential homes. In this case, the owner must have the opportunity to access or view his smartphone in his awareness of all possible intrusion attempts. All movements of the intruder will be monitored by the owner so that the necessary action can be taken in a timely manner.

The example above is clear that human intervention is still needed in implementing a security system. This happens to almost all security systems that are implemented at home. The existence of machine learning techniques can and has provided opportunities to enhance the capabilities of a security system. This research discusses more about machine learning techniques in security systems. The purpose of using machine learning technology is to reduce the dependence of the system on humans in its implementation. The system will have the ability to study all input images, in particular to recognize all authorized persons. Each control measure will be carried out based on the output provided by the learning system. To enhance recognition capability of the security system, a controlled Euclidean Distance filtration is introduced here. A Success Rate formula is also added to provide more convincing results.

2. RESEARCH METHOD

The system being developed includes an application program that is supported by hardware in the form of a microcontroller connected to a camera and several sensors. The first step of developing this system is to ensure that all the required hardware components are functioning properly and are interconnected. For application software development, the Extreme Programming method or simply called XP is used with some modifications [1]. This development method is an agile process model that promotes rapid development of application software.

All requirements for a home security system are identified in advance as the basis for determining the application process or function as well as quickly designing the application software. All necessary system controls must be able to carry out the specified system functions. Code writing is done as soon as the system design is complete. Writing code followed by system testing is carried out repeatedly until all identified system requirements are met. The function of the microcontroller in moving the camera and other sensors is also included in this test. This includes the improved function to recognize a face.

3. LITERATURE REVIEW

The characteristics of a modern home security system have implemented a microcontroller as the main controller of the whole system[14]. Arduino Uno is a microcontroller which is implemented through the use of an IDE (integrated development environment) as a means of programming the microcontroller [11]. Arduino hardware provides several pins as input and output connectors that can be connected to any sensor or hardware [2]. Serial communication with USB (universal serial bus) is carried out using the Software Serial Library [3]. Servo motors and cameras are types of hardware that can be connected and controlled by Arduino [4].

The use of cameras has an important role to play in identifying intruders. The image captured by the camera will later become solid evidence for the owner when all precautions fail. As stated earlier, introducing machine learning algorithms for recognizing face will be an added value for security [5][12]. For quick application based machine learning development, OpenCV is implemented because it provides the haarcascade_facefrontal_default.xml classifier to detect and recognize faces [6][7]. To train classifiers, positive and negative images are used in machine learning through the Haar cascade classifier [8].

Local Binary Pattern Histogram (LBPH) is one of the algorithm methods available in OpenCV for face recognition [9]. A local binary operator and one of the best texture descriptors are the basis of the method to create a combination of the Local Binary Pattern with histogram. The use of the histogram here is to find the frequency of the similarity of values so that the process is faster. A part from that, this method is also easy to understand and easy to use as illustrated below. Through the data ID (identifier) which represents the image in the dataset, it is used as a reference to be compared with the input image. Known or unknown faces can be created by matching the trained data.

4. RESULT AND DISCUSSION

The discussion of Results contains into three parts namely: System requirement, System design and implementation, and System testing. The improvement of algorithms in recognizing faces is elaborated in the System Testing part in which Success Rate Formula plays an important role in providing more convincing results

4.1 System Requirement

As is the case in any system that uses a microcontroller, several sensors and devices are required to obtain control signals and carry out control actions. Motion detection can be carried out via a laser beam directed at the LDR (Light Dependent Resistance) connected to the Arduino. Control action will be generated when the laser beam is blocked by an object. Opening or closing the door (via servo motor), sounding an alarm and contacting the police are all examples of control measures. Cameras can also be used as sensors to detect intruders.

This paper will highlight the use of machine learning in Home Security Systems. Thus, general control measures will not be discussed in detail here. Camera is required to capture "face" images as input data for the face recognition process. Machine learning algorithm is implemented in the process to recognize "face" image input. Such a process needs to be trained with several authorized "face" images before being tested for its ability in recognizing "face". As happens in other control measures, any detected unauthorized face will cause the control action to be generated in response to the emergency condition. The Success Rate formula is introduced to improve the ability of face recognition processes.

4.2 System Design and Implementation

System contains five blocks which are NodeMCU Microcontroller (include WiFi Module), Camera, LDR, Servo motor and Mobile Device as shown in Figure 1. This diagram represents the whole functions of the system in which the Microcontroller receives a signal for control processing and the result of control processing is delivering a signal for related control measures. Arrows represent the direction of movement of data or control signals. Circuit diagram of the system is shown in Figure 2.

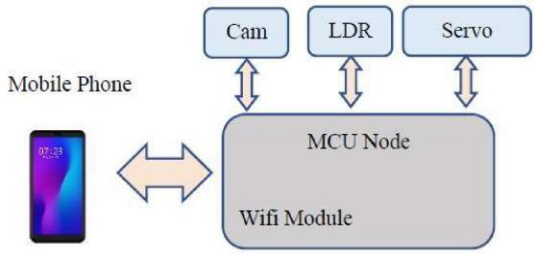


Figure 1. System block design

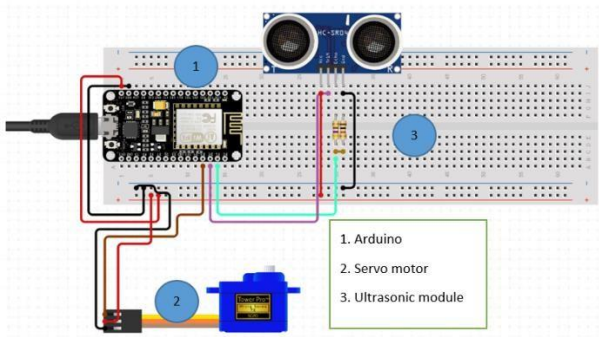


Figure 2. Circuit diagram of the microcontroller system

Furthermore, the high level view of the system, especially in highlighting the implementation of machine learning algorithms, is shown in the use case illustrated in Figure 3. Five activities can be carried out. The first one is getting a WiFi connection in order to communicate with the IoT. Multiple trials can be applied until the request is fulfilled. The machine learning use is initiated by "Capture Face" use case activation. Use case "View Face" activation allows a camera to capture images and, in turn, recognition process is executed through the use of "Verify Descriptor" use case. The rest is the control actions as a result of a recognition process which all will be carried out by NodeMCU through "Command by Voice" and "Tap Button".

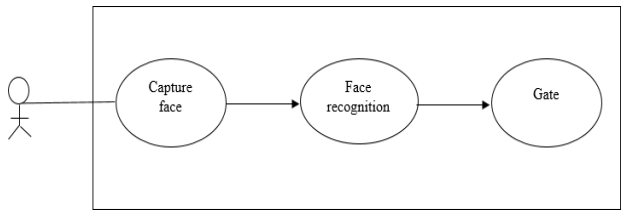


Figure 3. Use case diagram of the system

As shown in Figure 3, the camera will capture a "face" image and also send it to the Arduino for control processing. The servo motor receives control signals from the Microcontroller to implement related control actions such as opening or closing the gate. Controlling the system from a mobile device is done via a WiFi module. In this case, the mobile device sends a control signal to the Microcontroller and a notification is sent to the mobile device as a result of control processing in the Microcontroller.

A Camera is the primary device for implementing machine learning in home security systems [13]. All training "face" images are obtained from the camera and stored in a directory called "dataset" for training purposes. Each image in the dataset needs to have an ID label for the algorithm process in recognizing input images and providing an output. As stated by Mahmoud (2020), 4 parameters (Radius, Neighbours, Grid X, and Grid Y) are used in implementation of LBPH algorithm. The first step for the training is to create an intermediate image that describes the original image in a better way by highlighting the facial characteristics. The concept of sliding window is used in the algorithm through radius and neighbour parameters.

Several steps are taken for the process of recognizing face image as illustrated Figure 4. The Linear Support Vektor Machine (SVM) Formulation (Binary Classification) is adopted for the process of face recognition. The linear decision function for a binary classification problem *f(x)* is defined in equation 1[15].

$$f(x) = w.x + b \quad (1)$$

where the weight vector is *w*, the input feature vector is *x* and the bias term is *b*. The sign of the decision function is used to determine the predicted class which is defined as equation 2[15].

$$y = sign(f(x)) \quad (2)$$

When f(x) is greater than or equal to zero, then y is equal to 1, but when f(x) is less than zero, then y is equal to -1. The distance between the decision boundary and the nearest data point from either class is represented by the margin which is defined as equation 3[15].

$$M = \frac{1}{||w||} . abs(f(Xi)) \quad (3)$$

where x_i is a given data point. A Soft margin needs to be implemented to maximaze the margin while minimazing misclassifications, which is defined as equation 4[15].

$$min_{w,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{N} \max(0,1 - y_i . f(x_i)) \quad (4)$$

Where:

- $||w||^2$ is the regularization term that penalizes large weights.
- C is the regularization parameter controlling the trade-off between maximizing the margin and minimizing the classification error.
- N is the number of training samples.
- y_i is the true class label of the i^{th} data point.

The image part is represented as a window of 3x3 pixels. Another representation is a 3x3 matrix consisting of 9 squares, where each square contains a number representing the pixel intensity (0 ~ 255). In this matrix, the threshold (90) is obtained from its central value. This threshold will determine the binary of the next matrix squares. Any number above 90 will be converted into a binary 1 whereas a binary 0 is obtained from any number below 90. This conversion yields a Binary 10001101 matrix which represents Decimal 141 (expressed in the final matrix).

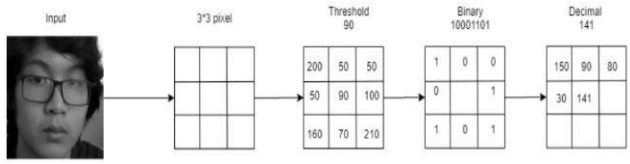


Figure 4. Procedure for better presenting original image characteristics

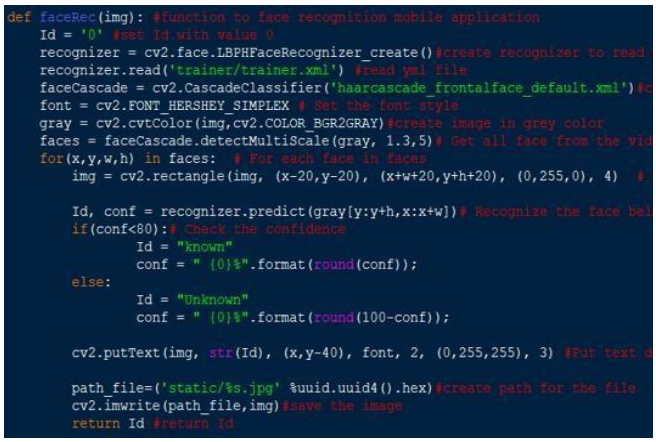
The Grid X and Grid Y parameters will be used to define the image into multiple grids. Each grid is then represented as a histogram. These histograms are then put together to form a larger histogram which at the end will represent the original face. The "trainer.xml" file is used in this research to store all training images of the face. Face recognition is carried out through a comparison of two histograms where the closest histogram represents the known face.

The XML file (haarcascade_frontalface_alt.xml) needs to be loaded before it detects the presence of a face and creates an identifier. In this case, the Local Binary Pattern Histogram (LBPH) method is used to load the "trainer.xml" file as a reference for the face recognition process captured by the camera. Code implementation for detecting and recognizing faces is shown in Figure 5.

```
recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainer/trainer.xml')
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath);
```

Figure 5. Code implementation for detecting face with "trainer.xml" file as reference

The face recognition process in mobile applications is represented by the faceRec() function as shown in Figure 6. The LBPH algorithm is used to recognize the face in the database and this is represented in the code recognizer = cv2.face.LBPH FaceRecognizer_create(). Recognizer classifier is loaded using the code faceCascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml'. Code img = cv2.rectangle() is used to detect the face after converting image to grayscale(cv2.cvtColor) and obtaining faces (faceCascade .detectMultiScale). The statement of "if-else" is used to recognize the image. The smaller the percentage value (represented in the conf variable), the more similar the face will be. Finally, the image is saved in the static directory through the use of codes path_file=('static/%s.jpg' %uuid.uuid4().hex) and cv2.imwrite(path_file,img) as shown in Figure 6.

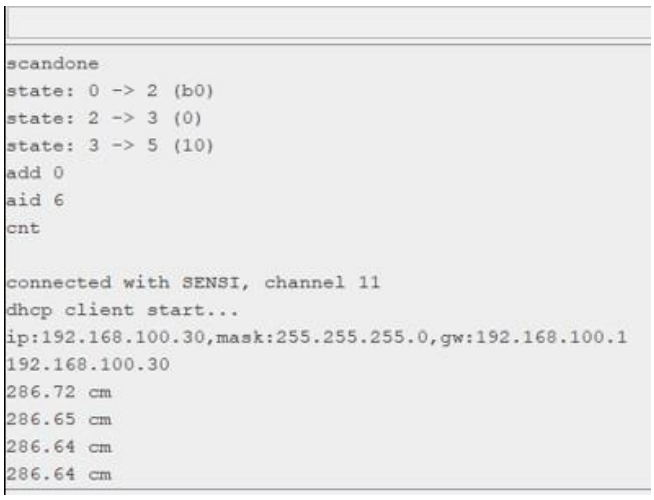


```
def faceRec(img): #function to face recognition mobile application
    Id = '0' #set Id with value 0
    recognizer = cv2.face.LBPHFaceRecognizer_create()#create recognizer to read
    recognizer.read('trainer/trainer.xml') #read yml file
    faceCascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')#c
    font = cv2.FONT_HERSHEY_SIMPLEX # Set the font style
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)#create image in grey color
    faces = faceCascade.detectMultiScale(gray, 1.3,5)# Get all face from the vid
    for(x,y,w,h) in faces:  # For each face in faces
        img = cv2.rectangle(img, (x-20,y-20), (x+w+20,y+h+20), (0,255,0), 4)  #

        Id, conf = recognizer.predict(gray[y:y+h,x:x+w])# Recognize the face bel
        if(conf<80):# Check the confidence
                Id = "known"
                conf = " {0}%".format(round(conf));
        else:
                Id = "Unknown"
                conf = " {0}%".format(round(100-conf));

        cv2.putText(img, str(Id), (x,y-40), font, 2, (0,255,255), 3) #Put text o

        path_file=('static/%s.jpg' %uuid.uuid4().hex)#create path for the file
        cv2.imwrite(path_file,img)#save the image
        return Id #return Id
```

Figure 6. Code implementation for face recognitizing in mobile application

4.3 System Testing

The test implementation begins with the need for communication between the Arduino and the Android application on a mobile device. The WiFi module attached in NodeMCU Arduino functions to send and receive data from mobile devices. Figure 7 shows an IP address 192.168.100.30 is detected. This indicates that the WiFi module is working properly. Whereas Figure 8 shows "running on http//192.168.100.249:5000" displayed on the screen and this is also an indication that the mobile application has obtained a connection from the WiFi module attached in NodeMCU Arduino. Making this all work, the Android app has a wireless connection for control measures from streaming camera-captured images to face recognition processes.



```
scandone
state: 0 -> 2 (b0)
state: 2 -> 3 (0)
state: 3 -> 5 (10)
add 0
aid 6
cnt

connected with SENSI, channel 11
dhcp client start...
ip:192.168.100.30,mask:255.255.255.0,gw:192.168.100.1
192.168.100.30
286.72 cm
286.65 cm
286.64 cm
286.64 cm
```

Figure 7. Testing result for WiFi module function

Euclidean Distance formula [10] written as equation 1 is adopted for calculating the distance between two histograms. The face recognition algorithm applies this formula in which the Euclidean Distance needs to be controlled in its filtration process. To obtain better results in recognizing faces, this filtration process needs to be applied in such a way that the Euclidean Distance value can be controlled.

$$D(a,b)\sqrt{\sum_{i=1}^{n}(ai-bi)^2} \qquad (5)$$

The filtration process running in the application can be described as follows. Testing images are represented in "ai" and "bi" where the smaller the distance between the two histograms will represent the higher the level of resemblance to the original face as stated in Equation 5. The filtration process needs to maintain the Euclidean distance of an image as small as possible with the reference image. Success rate is introduced to measure the similarity level of an image being tested against the original or reference image. This is represented in Equation 6. Success rate is calculated from the following scenario. The minimum Euclidean Distance resulted from the testing is considered as having a 100% success.

The Euclidean Distance of each image resulting from the test needs to be calculated first to get the image with the smallest Euclidean Distance value. The success rate is calculated from how much the similarity value is between a certain image and a reference image multiplied by 100%. This can be represented as the mathematical formula as Equation 6.

$$Success\ rate = (1 - \frac{Image\ x - Image\ R}{Image\ R}) * 100\% \qquad (6)$$

Where Image X is the image being calculated for its similarity level against the reference image (Image R). Both images are represented in Euclidean Distance, where the Image R is the image identified as having the smallest Euclidean Distance. Success rate is represented in percent (%) in which the greater percentage of the success rate, the higher the level of similarity with the original image.

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:20:19) [MSC v.1925 32 bit (In
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\NITRO\Downloads\Programs\Thesis\Web udah jalan - Copy - Copy
\app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployme
nt.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://192.168.100.249:5000/ (Press CTRL+C to quit)
```

Figure 8. Testing result for obtaining WiFi connection from android app

Table I shows the calculation results of Euclidean Distance (ED). It lists four Euclidean Distance results for four different images. Image_1 is identified having 48 ED as the smallest Euclidean Distance, while the rest (image_2, image_3, and image_4) are having Euclidean Distance of 66ED, 122 ED and 116 ED consecutively. Image_1 is identified as having the smallest ED, henceforth, image_1 is used as a reference image to determine other images to be tested for the level of similarity with the original image. The difference of Image_4 from Image_1 is (116 ED – 48 ED) which is equal to 68 ED. The difference of Image_3 from Image_1 is (122 ED – 48 ED) which is equal to 74 ED. whereas the difference of Image_2 from Image_1 is (66 ED – 48 ED) which is equal to 18 ED.

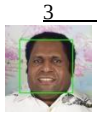
After all differences are obtained, the Success Rate can then be calculated as follows. Using the formula or Equation 2 above for Image R=48 ED and Image X=Image_2 = 66 ED, the calculation result of Success Rate for Image_2 is 62.5 %, while for Image_3 and Image_4, it shows that the results of the calculation of the Success Rate are negative or less than zero percent. For Image_2 with the Success Rate 62.5%, it is considered that Image_2 is having closer similarity compared to both Image_3 and Image_4. As stated above, both Image_3 and Image_4 are having negative Success Rates. These results indicate that the two images have a "Totally Difference Face" and this can also be said to be "failed" or "not success" as shown in Table 1. Thus, to make a more convincing result in

recognizing a face, the difference between Image R and Image X should be maintained as small as possible so that the Success Rate will be positive and higher.

Table 1. The calculation result of Eucledean Distance

Subject	1	2	3	4
Testing image				
Euclidean distance	48ED	66ED	122ED	116ED
Success rate	100%	62.5%	Failed	Failed

To further ensure this enhanced algorithm works properly, a test has been applied directly to the security system in which its camera is used to capture a facial image. The main Gate is then controlled based on the results of the facial recognition process. The test results of the face recognition process and the control process are shown in Table II. When the authorized face is captured by the camera, then the system will indicate as “known” (representing that the face recognition process is working properly) and the “gate opened” (representing that the control process is working properly). However, when an unauthorized face is captured on camera, the reverse result will be an indication of "unknown" and "gate not open".

Subject	1	2	3	4
Capture face				
Face recognition	Known	Known	Unknown	Unknown
Control action	Gate opened	Gate opened	Gate not open	Gate not open

5. CONCLUSION

As stated earlier, the aim of this study is to show that, in the system authorization process, face recognition through the use of machine learning algorithms can be applied to home security systems. Euclidean Distance filtration which is complemented by the implementation of the Success Rate Formula shows more convincing results in the process of recognizing faces. This is represented in the test in which the process of face recognition and control are working properly as a result of indication “Known” and “Gate Opened”. Thus, the appropriate control measures with improved facial recognition algorithms can realize more convincing system security.

In terms of security systems, currently it only implements a learning algorithm to recognize faces. Further development needs to be done to improve the capabilities of this home security system, especially when a learning algorithm is adopted in the system. Recognizing intruders' behaviour in the yard captured by cameras can be a warning sign before an intruder approaches the access gate. Recognizing the behaviour of the homeowner in the house can also be used as a sign to turn on certain household electronic appliances such as lights, air conditioners, TVs and DVD players. Cameras can also be used to capture dangerous patterns that occur in the house such as the appearance of a fire in the room.

AKNOWLEDGEMENT

Author acknowledge Michael for his assistance in preparing and carrying out this time-consuming and tedious test to support the research. Without his help, this research would not have produced valuable data and been completed on time

REFERENCES

[1] Roger S. Pressman, “Software Engineering: A Practitioner's Approach”, Eight Edition, McGraw-Hill Education, New York, 2015.

[2] Massimo Banzi, Michael Shiloh, “Make: Getting Started with Arduino”, 3rd Edition, Maker Media Inc., Sebastopol, CA 2014.

[3] Arduino, "SoftwareSerial Library," [Online]. Available:https://www.arduino.cc/en/Reference/softwareSerial.

[4] R. Roestam, N. Hadisukmana, “CARDUINO: An Effort Towards Commercial Autonomous Public Vehicles Based On Arduino,” 1st International Conference on Sustainable Engineering and Creartive Computing (ICSECC), 2019.

[5] Adam Geitgey, “Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning” Medium Corporation, 2016. {Online}.

[6] P. Viola, M. Jones, “Rapid object detection using boosted cascade of simple features,” IEEE Conf. Proceedings on Computer Vision and Pattern Recognition, 2001

[7] E. Villa, "Haar Cascade", Medium Corporation, 2019. [Online]

[8] N. Hadisukmana, A. Yudianto, “Paper Money Recognizer Using Feature Descriptor,” International Journal of Electrical Engineering and Computer Science, Vol. 12 No. 1. 2018

[9] Mahmoud Harmouch, “Face Recognition Based On LBPH Algorithm”, Dev Genius. [Online]

[10] Abdo Y. Alfakih, “Euclidean Distance Matrices and Their Applications in Rigidity Theory”, Springer, 1st ed. 2018

[11] Teuku, H. R., Ghofir, A., Roestam, R. (2019). Smart Postpaid Electricity Meter using Arduino. In International Conference on Sustainable Engineering and Creative Computing

[12] A S Romadhon (2018). System Security and Monitoring on Smart Home Using Android. Journal of Physics: Conference Series

[13] Nico S, Wingky RW (2018). Design of Smart Home Security System using Object Recognition and PIR Sensor, 3rd International Conference on Computer Science and Computational Intelligence 2018

[14] Syedzainnasir, Smart Home Security System using Arduino, The Engineering project, 2021 [Online]

[15] Y. Wang, Q. Wu, “Research of Face Recognition Technology Based on PCA and SVM”, IEEE Xplore: 2021.