

## Face Recognition System for Prevention of Car Theft with Haar Cascade and Local Binary Pattern Histogram using Raspberry Pi

<sup>1</sup>Muhammad Hanif Abdurrahman, <sup>2</sup>Haryadi Amran Darwito,  
<sup>3</sup>Akuwan Saleh

Electrical Engineering Department,  
Electronic Engineering Polytechnic Institute of Surabaya,  
Surabaya, Indonesia

E-mail: <sup>1</sup>muhhanifabd@gmail.com, <sup>2</sup>amran@pens.ac.id, <sup>3</sup>akuwan@pens.ac.id

*Received July 5, 2020; Revised August 28, 2020; Accepted December 7, 2020*

### Abstract

In this era, the occurrence of vehicle theft has become a fairly frequent problem, especially in big cities like Jakarta and Surabaya. Although the technology for car security is very sophisticated (e.g. keyless system), but there are many cases that thieves still can break into the system. Once a car was stolen, the whereabouts of the car was unknown and the thief was on the loose. The goal of this research is to overcome this problem. As a device, this research works on Raspberry Pi 3 that connected with the Raspberry Pi Camera. Using the OpenCV library, with the Haar Cascade method for face detection, and Local Binary Pattern Histogram for face recognition. The device must be connected to the internet in order to send the information using a Telegram message. The research results show the success of the device system in face-recognizing between the car owner and car thief with optimal conditions in the morning until the afternoon with the light intensity around 660 to 1000 lux, and best recognizing distance at 50 cm. The success rate for obtaining the car's location for the outdoor condition is 100%. Even if there is a slope or an error data, it can be tolerated because the difference was not too high, about 0.1-1.0 m.

**Keywords:** Face Recognition, Haar Cascade Method, Local Binary Pattern Histogram, OpenCV, Technology.

### 1. INTRODUCTION

The revolution industry 4.0 brings the huge improvement in the technology sector, resulting in rapid improvement on the security systems. The one use of this security system is the use of face recognition using the computer process, called computer vision. Computer vision is a fast-growing field for analyzing, modifying, and high-level understanding of images, e.g. faces. OpenCV is one of the Image Processing library using the Computer

Vision. OpenCV provides libraries for the Haar Cascade and Local Binary Pattern Histogram methods [9].

In this era, the occurrence of motor vehicle theft has become a fairly frequent problem, especially in big cities like Jakarta and Surabaya [15]. For example, in the Cirebon region, the occurrence of car theft in 2017 was only 20 cases, but in 2018 the number of car theft occurrences doubled to 42 cases [2]. Although the technology for car security was very sophisticated, but there are many cases that the thieves can still break into this system [3], [8]. When the car was stolen, the whereabouts of the car was unknown and the thief was on the loose.

This research was designed to make a technology system that can detect and recognize if there is a car theft occurred, by automatically through the face recognition of the driver. This device works on the Raspberry Pi 3 Model B, that connected to the Raspberry Pi Camera Module. The process is done by using the OpenCV library, which provide the Haar Cascade method for face detection, and Local Binary Pattern Histogram method for face recognition of the driver. Lastly, the system is using the Ublox Neo-6m GPS module, and GSM Modem with USB SIM Card.

This research is limited by the Raspberry Pi camera specification, which didn't support the use for night conditions. Furthermore, the device must be connected to the internet for the communication with smartphones using a Telegram chat. For the accessory, it is only use the sunglasses with a clear lens.

## 2. RELATED WORKS

Research conducted by Suhepy Abidin [1] on "Deteksi Wajah Menggunakan Metode Haar Cascade Classifier Berbasis Webcam Pada Matlab" has produced a face detection system that was carried out using Haar features to find out if there are facial features or not in the captured images. From the test, the face can be detected correctly in the perpendicular position, the face tilted by 30° to the right and left, bowed, and raised with a tilt of 15°. Based on the research above, the Haar Cascade Classifier method is ideal for face detection. The OpenCV is able to do the detection for the non-frontal face taking with -74° to +74° angle variances [12], [13].

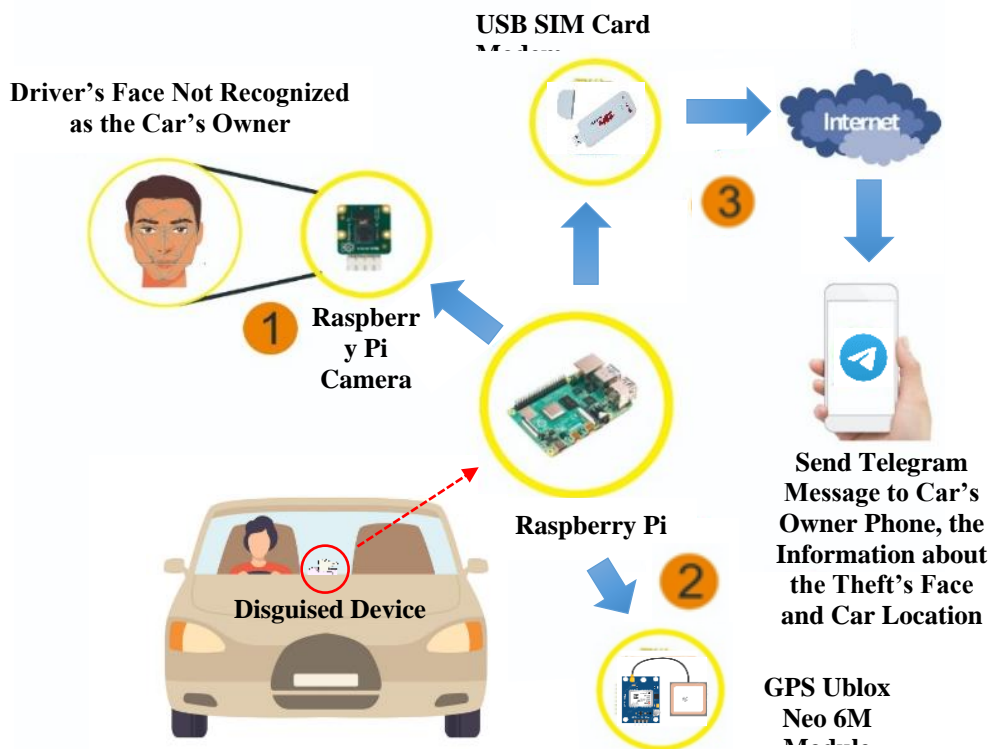
Other research conducted by Ratih Purwati [6] on "Pengenalan Wajah Manusia Berbasis Algoritma Local Binary Pattern" has produced a face recognition system with the Local Binary Pattern (LBP) method with the help of Support Vector Machine (SVM) to predict the classification. The accuracy obtained from the testing with the data is 100.0%. In facial expression testing, the highest accuracy is 100.0% in sad, neutral expressions, with the eyes closed using linear kernels, and in shocked, neutral expressions, with eyes closed using RBF kernels. The lowest accuracy for testing is on happy expressions, with 86.6% accuracy. Based on this research, a facial recognition system using LBP feature extraction can be done well.

### 3. ORIGINALITY

The existing research with 2 different methods above will be combined in this research. The Haar Cascade method will be used for the face detection and Local Binary Pattern Histogram for the face recognition of the driver to prevent the car theft occurrence. This research works on Raspberry Pi with Raspbian Operating System and using the python programming language. The experiment process was performed in several stages, the first stage is for the face detection and recognition system, second is the process to obtain the coordinate location of the car, and the last is the experiment for the device to smartphone communication using Telegram messaging.

### 4. SYSTEM DESIGN

Figure 1 shows the hardware system design that we proposed for the research. There are 3 main processes about the system, and each process will be explained in section 4.1 – 4.3:



**Figure 1.** Hardware System Design

This device uses several components: Raspberry Pi 3 Model B as the microcontroller of the device, that connected to a Raspberry Pi Camera Module (official product from the Raspberry Pi Industry). The camera was placed in the center of the car dashboard. GPS Ublox Neo 6m module for giving the information about the car coordinate location, and GSM USB modem with a SIM Card that used for giving the Raspberry Pi connection to the internet in order to send the information, in the form of Telegram

messages into the car owner's smartphone. The Raspberry Pi power supply comes from the USB port of the car, where the Raspberry Pi will automatically turn on when the car engine is started. When the Raspberry Pi is already turned on, the camera will automatically start to detect and recognize the face of the driver.

There are 3 decisions from the detected and recognized face by the system:

- (1) The driver's face was recognized by the system as the car owner.
- (2) The system recognize the driver's face, but the face was not in the database, so it will be indicated as a car thief.
- (3) The system can't detect any face from the detection process so it's also indicated as a car thief.

For every decision, the system will send different information using the Telegram messages to the car owner based on the condition above.

## **4.1 Face Detection and Recognition System**

### **4.1.1 Driver's Picture Taking**

The first stage carried out by the system was taking the picture of the driver, so the camera needs to be activated. The database preparation is done when setting the device using a monitor, keyboard, and mouse for the Raspberry Pi. The lighting condition when taking the dataset must be bright enough. When the program has started, the device will automatically detect and capture the face as much as 30 photos for every second. The capture process started with the frontal face taking, then the owner must up, right, down, and left with 35 degrees of angle tilt, so that all sides of the face can be captured. The captured images are then stored in a database for each owner.

### **4.1.2 Image Pre Processing**

#### **Scaling Process**

The next step in image pre-processing after taking pictures was scaling. Scaling is the process of resizing digital images, which needs to be done so that all digital images have the same size [4]. The digital image is minimized by the interpolation method. This method uses a region average to represent the region that already exists in the OpenCV library.

#### **Grey scaling Process**

After the scaling process, the next step is grey scaling, where digital images that were originally in RGB (Red, Green, and Blue) format will be converted to the grayscale (black-white) format. This compression process needs to be done to ease the process of the next computing. Because in Haar Cascade method, one of the types of images that can be processed is the grayscale format. Then the formula for calculating the grayscale value of the color image can be calculated by:

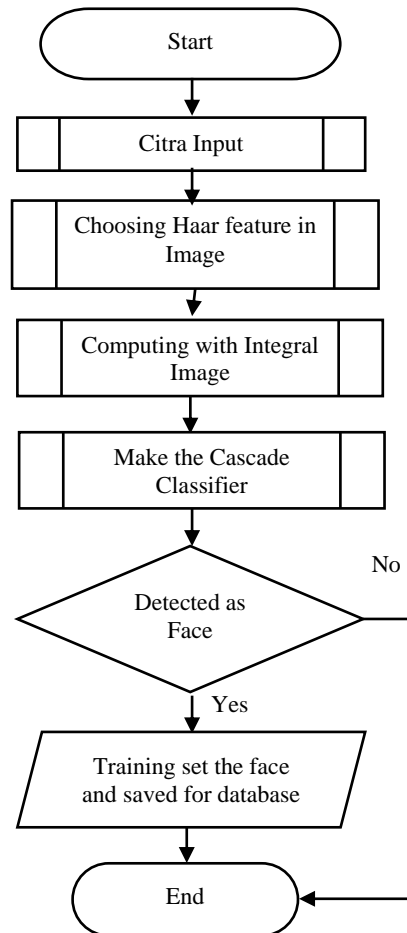
$$W = 0.2999 \times R + 0.5870 \times G + 0.1140 \times B \quad (1)$$

Where:

R = Red; G = Green; B = Blue

#### 4.1.3 Face Detection using Haar Cascade Method

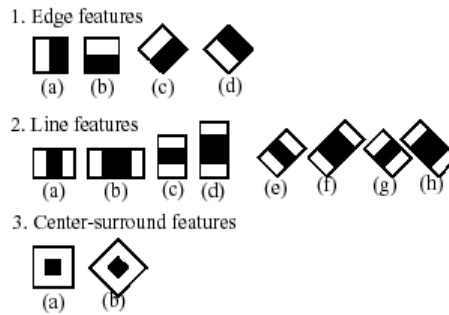
The figure 2 below shows the flowchart of the Haar Cascade Method that used for the face detection.



**Figure 2.** Haar Cascade Method Flowchart

#### Choosing Haar Feature in Image

The Haar Cascade method uses the Haar Feature function which requires training in advance to get a decision tree called Cascade Classifier, to determine whether there are face objects or not in the processed image. Haar feature for face detection is a combination of black and white rectangles. The position of these rectangles is defined relative to a detection window that acts like a bounding box, where the size of each box is 24x24 pixels. There are 3 types of rectangular Haar features:



**Figure 3.** Types of Haar Features

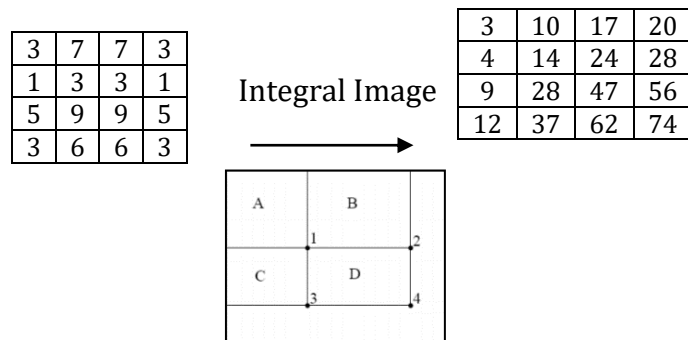
With this 24x24 pixel window size, it can produce more than 160,000 features [11]. Every box is processed by subtracting the value of the black pixel area with the white pixel area. Then the difference value can be calculated, which indicates the dark and light areas. The grey level equation is define with equation (2):

$$F(x) = \text{Sum Black Rectangle} - \text{Sum White Rectangle}. \tag{2}$$

If the difference is higher than the threshold value of face feature, then it can be said that the facial features exist.

**Computing with Integral Image**

To simplify the process of calculating feature values of images, the Haar algorithm uses media in the form called Integral Image. Integral Image is an image where the value of each pixel is the sum of the top left to bottom right pixel values [16]. Suppose an input image has a feature value:



**Figure 4.** Integral Computing

To find out the pixel value of the quadrilateral D in the figure above can be done by using the equation (3):

$$D = (A + B + C + D) - (A + B) - (A + C) + A \tag{3}$$

The value of features will be found, with an easier and faster process.

**Make the Cascade Classifier**

Among all the 160,000 Haar features, most of them were not relevant, it means that many of these features are weak features for the use of face detection. For the process of selecting the best and fast of many features, it

can be achieved using Adaboost (Adaptive Boosting) concept. This algorithm builds a "strong" classifier as a linear combination of several "weak" classifiers [10], [11]. With Adaboost, it gives a reduction from 160,000 features to 6,000 features only.

For the face detection in an image, most parts of the images are non-facial areas. Surely it will take a lot of time to find the facial areas only. A simple method is needed to check whether the window detection is a face area or non-face area. For this, the Cascade Classifiers concept was used. These features are grouped into several classification stages and implemented one by one. If the window fails at the first stage, it will be discarded and the remaining features won't be considered. If passed, the second stage will be applied and so on until the system find a face in image.

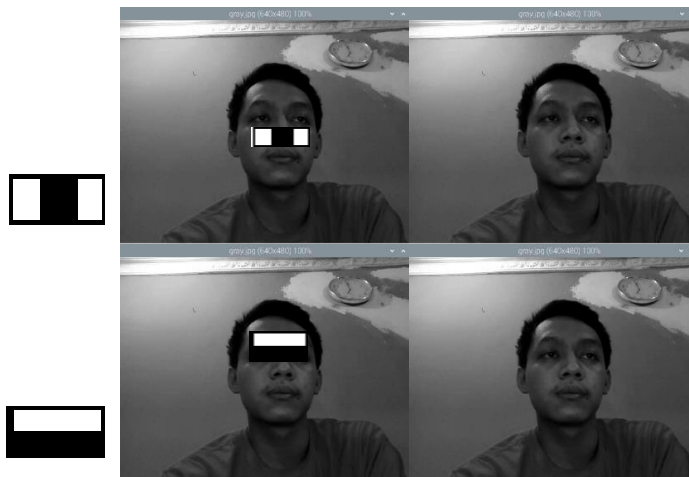
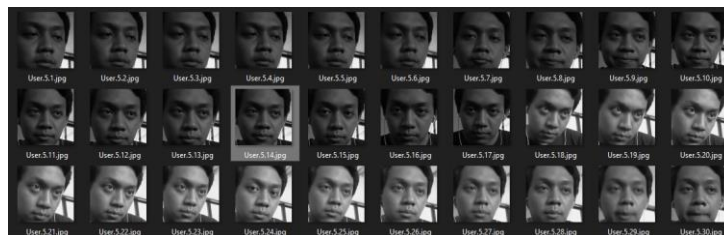


Figure 5. Haar Feature Search Process.

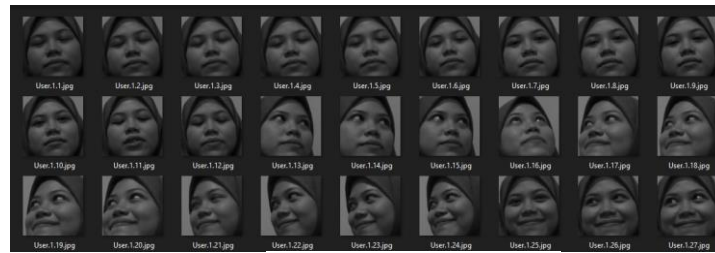


Figure 6. Face Detected using the Haar Feature.

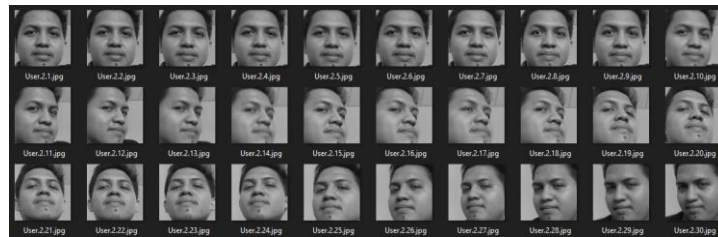
After successfully detecting the face from the captured image, the system will only save the face images in a database/dataset folder. The system will do the process to take the owner's face image data as much as 30 times. Here are some samples of face images stored in the database.



(a) Database with name Hanif



(b) Database with name Septi



(c) Database with name Gilang

**Figure 7.** Owner's Face Sample for the Dataset.

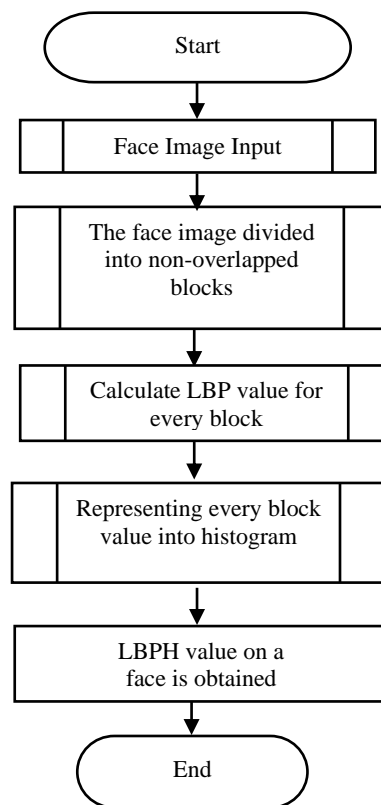
From the figure above, the dataset collection wasn't perfect to get all sides of the face. It happens because there was a delay 2-3 seconds between the real-time face moves with the face captured in the system. The delay happens because the Raspberry Pi 3 specification was not capable enough for the use of image processing.

There were also a different illumination condition happen that makes the dataset different from one to another, because we cannot expect to get a good illumination because the illumination changes are hard to predict and control. To overcome this problem, the Optifuzz method [17] can be used for the further development of the system.

#### 4.1.4 Face Labelling and Recognition using LBPH Method

Local Binary Pattern Histogram (LBPH) method will be used for the labelling process of the car owner's face label in the database, and also for recognizing face image data that detected in real-time capture. The flowchart of the LBPH method is given for figure 8 below:





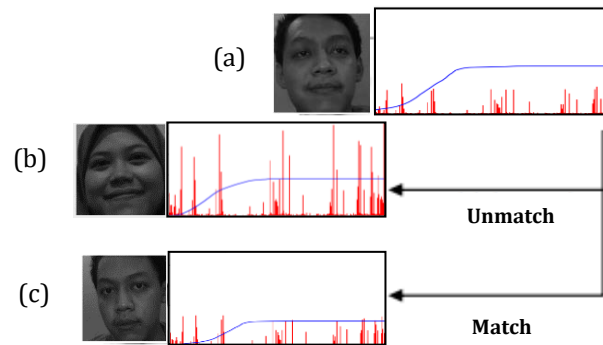
**Figure 8.** Flowchart of LBPH method

In Local Binary Pattern Histogram Method, the image will be divided into several block sections, without crossing the boundary of the image. Then the feature extraction process will be carried out to get the histogram value of each block, by changing each block into pixels of an image with a size of 3x3, then taking one pixel in the centre to become a threshold.

For each neighbour of the threshold, a new binary value is set. 1 for a value that is equal to or higher than the threshold, and 0 for a lower value than the threshold. After the binary number is found, it will be converted to decimal value, which is then formed in the histogram form by concatenate the value of the previous block.

After the LBP histogram value obtained, the histogram results will be compared with the histogram results of database. So it can be compared whether the image has the same characteristics as the recognition results or not.

Supposed there were 3 pictures: (a) Hanif real-time captured face; (b) Septi face on the database; (c) Hanif face on the database; that has been captured and the LBP value in the form of histogram have been obtained. The system will process the captured image, whether it has a match with the image in the database or not. If the system found a match, it can be concluded that the face was registered in the database and successfully recognized as the car owner.



**Figure 9.** Matching the Image Histogram

OpenCV has 4 functions that can be implemented to perform a Histogram comparison, they are Correlation, Chi-Square, Intersection, and Bhattacharyya distance. For this research, the Chi-Square formula is used:

$$d_{\text{chi-square}}(H_1, H_2) = \sum_i \frac{(H_1(i) - H_2(i))^2}{|H_1(i) + H_2(i)|} \quad (4)$$

The  $H_1$  is the LBP Histogram value for the real-time detected face, and  $H_2$  is the LBP Histogram value (e.g. Hanif's face) from the trained face on the database. The lower the Chi-Square result value (d), the more accurate the match and so it will be indicated as the owner's face (e.g. Hanif's face). Chi-Square measurements are effective if there are similarities between a pair of histograms [14].



**Figure 10.** Face Recognition Results from the System

#### 4.2 Get the Device Location

The device location information will be sent continuously to the Raspberry Pi so it can get the car direction in case the car was stolen. By using the Ublox Neo 6m GPS module, a program was designed to obtain the location of the device in the car that will be used to get the coordinates of the car.

Data obtained through the GPS module comes in the form of NMEA data. NMEA stands for the National Marine Electronics Association. NMEA is the standard message format for almost all GPS receivers. The Neo 6m GPS

module updates this information once per second (1Hz frequency) by default. Below is the NMEA data obtained from Ublox Neo GPS module when observing:

**\$GPGGA,062208.00,0721.57091,S,11245.33633,E,1,04,3.17,27.3,M,17.0,M,,,\*75**

### 4.3 Device to Smartphone Communication

To connect the Raspberry Pi with an internet connection, a USB SIM Card modem was used. Using Telethon, a Python library for using the Telegram API. Several steps must be taken to be able to use the Telegram API:

- a. Install telethon using pip.
- b. Log in to the Telegram account to make Telegram API on <https://my.telegram.org>.
- c. Go to API Development tools and make the API.
- d. Save the `api_id` and `api_hash` for user authentication.
- e. Then we will be asked the Telegram account number for the robot and the authentication code.
- f. The Telegram API can be used to send any multimedia media files such as text, images, audio, and video without requiring authentication anymore.

There are 3 main information form that will be sent:

- Text; about the condition of the car (safe or unsafe)
- Images; from the thief face as much as 20 photos (for the stolen car decision)
- Text; information about the coordinate location of the car that updated for every 30 seconds (for the stolen car decision)

## 5. EXPERIMENT AND ANALYSIS

This section explained about the research experiment and the analysis from the research results. The experiment is done to get the performance of the system. The test results will give an idea whether the research is corresponding as the expected result or not. The experiment is done for every process inside the system.

### 5.1 Face Recognition Based on Light Intensity

The data collection method is done by registering the image of the car owner for the database. The test was conducted on 1 male as a car owner named **Hanif**, and 1 male as a car thief named **Hasan**. The analysis is performed after carrying out the testing process with various conditions. Each test in each condition is done for 5 times with 3 expressions like the table below.

**Table 2.** Expression Scheme for the Experiment

Experiment-	1	2	3	4	5
Expression	N	S	T	S	T

Table Description:

N : Normal (Face Expression in Normal Condition)

S : Smile (Face Expression in Smile Condition)

T : Teeth (Face Expression in Smile with Teeth)

The experiment based on the light intensity is done with 3 conditions, in the morning, afternoon, and night. The morning experiment is done at 07.00-09.00 WIB. The afternoon experiment is done at 11.00-13.00 WIB. And night experiment is done at 18.00-20.00 WIB. The results of this test will give an overview performance of the device in different lighting conditions.

The morning experiment is done at 07.00-09.00 WIB with a light intensity of around 660 lux. Data collection in this test was carried out 5 times for each face data.

**Table 3.** Experiment at Morning

Experiment -	Name: Hanif		Name: Hasan	
	Description	Similarity (%)	Description	Similarity (%)
1	Detected	45	Detected	0
2	Detected	50	Detected	0
3	Detected	47	Detected	0
4	Detected	45	Detected	0
5	Detected	48	Detected	0
Success: 100%			Success: 100%	

The afternoon experiment is done at 11.00-13.00 WIB with a light intensity of around 1000 lux. Data collection in this test was carried out 5 times for each face data.

**Table 4.** Experiment at Afternoon

Experiment -	Name: Hanif		Name: Hasan	
	Description	Similarity (%)	Description	Similarity (%)
1	Detected	51	Detected	0
2	Detected	55	Detected	0
3	Detected	53	Detected	0
4	Detected	49	Detected	0
5	Detected	48	Detected	0
Success: 100%			Success: 100%	

The night experiment is done at 18.00-20.00 WIB with a light intensity of around 50 lux. Data collection in this test was carried out 5 times for each face data.

**Table 5.** Experiment at Night

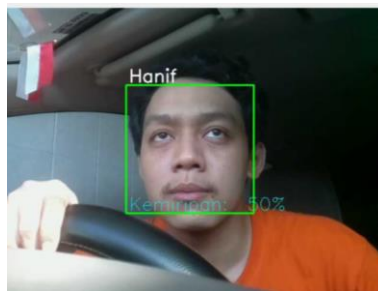
Experiment -	Name: Hanif		Name: Hasan	
	Description	Similarity (%)	Description	Similarity (%)
1	Not Detected	-	Not Detected	-
2	Not Detected	-	Not Detected	-
3	Not Detected	-	Not Detected	-
4	Not Detected	-	Not Detected	-
5	Not Detected	-	Not Detected	-
Success: 0%			Success: 0%	

Based on the experiment with morning lighting conditions, the success of the device to detect the both faces of the car owner and the car thief shows a 100% success rate in both faces. For facial image recognition testing, the car owner's (Hanif) face has a 100% success rate, with a fairly large degree of similarity, based on the data that has been registered in the database. While the device testing to recognize the car thief (Hasan) also has a 100% success rate, with a similarity rate is 0% on all tests. This level of similarity means that no face in database that matches with the face of Hasan, so it will be indicated as a thief.

In the afternoon experiment, the success of the device to detect both faces shows a 100% success rate in both faces. Facial image recognition for the car owner's also has a 100% success rate, with a larger degree of similarity than the morning experiment. While the device testing to recognize the car thief (Hasan) also has a 100% success rate, with a similarity rate is 0% on all tests.

In the night experiment, the success of the device to detect both faces shows a 0% success rate in both faces. This indicates that face detection in night lighting conditions cannot be done at all. Therefore, the next stage of testing, facial recognition also cannot be done. It can be happened by one important factor, that is the lack of lighting intensity, coupled with the factor of the window film on the car, which caused the inside of the car gets darker and the camera cannot perform both detection and recognition.

The maximum and best accuracy is only 55%, it can be happen because there are differences in data collection conditions (such as distance and light intensity) between the taking face images for the database and when the taking in real time at the car.

**Figure 11.** Face recognition at morning condition with 50 cm taking picture distance

## 5.2 Face Recognition Based on Taking Picture Distance

The experiment is done with 3 different testing distances at 50 cm, 80 cm, and 100 cm. The results of this experiment will perform the device performance for different taking picture distance conditions.

**Table 6.** Experiment at 50 cm

Experiment -	Name: Hanif		Name: Hasan	
	Description	Similarity (%)	Description	Similarity (%)
1	Detected	45	Detected	0
2	Detected	50	Detected	0
3	Detected	47	Detected	0
4	Detected	45	Detected	0
5	Detected	48	Detected	0
Success: 100%			Success: 100%	

**Table 7.** Experiment at 80 cm

Experiment -	Name: Hanif		Name: Hasan	
	Description	Similarity (%)	Description	Similarity (%)
1	Detected	24	Detected	0
2	Detected	29	Detected	0
3	Detected	23	Detected	0
4	Detected	30	Detected	0
5	Detected	22	Detected	0
Success: 100%			Success: 100%	

**Table 8.** Experiment at 100 cm

Experiment -	Name: Hanif		Name: Hasan	
	Description	Similarity (%)	Description	Similarity (%)
1	Detected	9	Detected	0
2	Detected	14	Not Detected	-
3	Detected	7	Detected	0
4	Not Detected	-	Detected	0
5	Detected	10	Detected	0
Success: 80%			Success: 80%	

Based on the experiment above, it can be concluded that the highest similarity value obtained is at 50 cm testing distance, with a similarity value is 50%. At the 50 cm and 80 cm test, the face recognition still can be done with a 100% success rate. While at the 100 cm test, during the 5 experiments, it has one failed detection. The value of similarity also gets smaller and decreases. It can be concluded that the farther the taking picture distance, the smaller the similarity value.



**Figure 12.** Face recognition based on taking picture distance

**5.3 Face Recognition Based on the Use of Accessories**

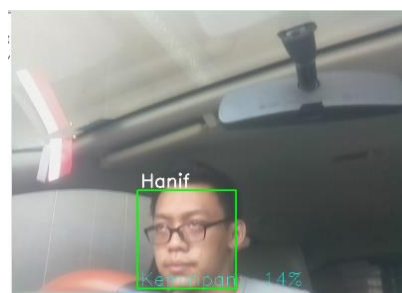
This experiment is done by using additional accessories, eyeglasses with the clear lens. This test is done with a distance of 80 cm between the face and the camera, inside the car. Data collection in this test was carried out 5 times for each face data.

**Table 9.** Experiment with the use of Accessories

Experiment -	Name: Hanif		Name: Hasan	
	Description	Similarity (%)	Description	Similarity (%)
1	Detected	14	Detected	0
2	Detected	16	Detected	0
3	Detected	13	Detected	0
4	Detected	20	Detected	0
5	Detected	14	Detected	0
Success: 100%			Success: 100%	

Based on the results above, the success of the device to detect the both faces of car owner and car thief has a 100% success rate. For facial image recognition testing, the car owner's (Hanif) face has a 100% success rate, even the similarity is not as good as without using eyeglasses. While the device testing to recognize the car thief (Hasan) also has a 100% success rate, with a similarity rate is 0% on all tests. This level of similarity means that no face in database that matches with the face of Hasan. So it can be concluded that the facial detection and recognition with the use of eyeglasses can be done well.

This success rate is because the eyeglasses used have clear lenses and the size of eyeglasses do not cover the face area. So the facial recognition process can be done successfully.



**Figure 13.** Face recognition with the use of eyeglasses

#### 5.4 Experiment for Getting the Device Location

This experiment is done to determine the GPS accuracy and the ability of the device to obtain the car coordinate. This test was carried out in the author's house at Waru, Sidoarjo and on 2 highways. For each experiment is done 3 times.

**Table 10.** Experiment in the Author's House

Experiment-	Description	Accuracy
1	Can't Get Location	-
2	Can't Get Location	-
3	Can't Get Location	-
Success: 0%		

**Table 11.** Experiment in Jl. Raya Tropodo, Waru Sidoarjo

Experiment-	Description	Accuracy
1	Success Get Location	Different 1 m
2	Success Get Location	Appropriate
3	Success Get Location	Different 0.4 m
Success: 100%		

**Table 12.** Experiment in Jl. Brigjend Katamso, Waru Sidoarjo

Experiment-	Description	Accuracy
1	Success Get Location	Appropriate
2	Success Get Location	Appropriate
3	Success Get Location	Different 0.4 m
Success: 100%		

Based on the results above, when the experiment is done at the author's house, the device was unable to obtain the location of the car. This can be happened because the GPS module could not receive any data from the GPS satellites, so that no data can be retrieved. While on the experiment at two highways, the GPS module successfully obtained the location data with 100% success rate. But there were a differences for the obtained data between the experiments and the real locations. This can be caused due to signal bending (due to gravity or reflection). However, the difference still can be tolerated because it was not too far away. From this experiment, it can be concluded that the GPS module works better for obtain the data when done in outdoors.

#### 5.5 Device to Smartphone Communication

This experiment was carried out to determine the reliability of the device for sending information in the form of text and image. The device will send a photo of the driver face and the location of the car to the car owner smartphone using Telegram application. Each experiment for send the information will be done 3 times.

Texts sent for the experiment:

1. "Warning! The Car is not safe! This is the Driver! "
2. "Car's Coordinate: 7.359515 S, 112.755606 E."



Images sent for the experiment:

**Table 13.** Images Sent for Experiment



**Table 14.** Sent Information in the Form of Texts

Text 1		Text 2	
Experiment-	Description	Experiment-	Description
1	Success	1	Success
2	Success	2	Success
3	Success	3	Success
Success : 100%		Success : 100%	

**Table 15.** Sent Information in the Form of Images

Image 1		Image 2	
Experiment-	Description	Experiment-	Description
1	Success	1	Success
2	Success	2	Success
3	Success	3	Success
Success : 100%		Success : 100%	

Based on the experiment results from 2 tables above, for send the Telegram information in the form of text and image has a 100% rate of success. The texts and images received on the smartphone of the car owner also match with the text or image sent. So it can be concluded that the device can send information in any form of media using the Telegram application correctly without any misinformation.

**6. CONCLUSION**

According to the experiment result, it can be concluded that the combination of the Haar Cascade method and the Local Binary Pattern Histogram for the face detection and recognition was good and promising. The device was able to distinguish between the car owner and the car thief. The optimal light intensity for face recognition is in morning until afternoon with the light intensity value around 660 to 1000 lux. At night conditions, with a light intensity of around 50 lux, the device cannot perform both detection and recognition at all. The best distance in taking pictures for the face detection and recognition is in the range of 50-60 cm from the camera.

The device can also recognize well the car owner and the car thief when using a clear lens eyeglasses accessories with 100% success rate. For obtaining the location of the device, the device must be able to receive the data from the GPS satellites first. From the experiment results, the device cannot receive location data when done in indoor area, but in outdoor area it can be done with 100% success rate.

## REFERENCES

- [1] Suhepy Abidin, **Deteksi Wajah Menggunakan Metode Haar Cascade Classifier Berbasis Webcam Pada Matlab**, *Jurnal Teknologi Elekterika* Vol. 15, No. 1, 2018
- [2] Cecep Nacepi, **Pencurian Mobil Naik 100 Persen, Tahun 2018 Terdata Sebanyak 42 Kasus**, Radar Cirebon, Report Number: -, 2019.
- [3] Aditi Shrikant, **A new report shows that most keyless cars are easier to steal than previously thought**, Vox Journalism, Report Number: -, 2019.
- [4] Nahdi Saubari, **Deteksi Citra Wajah Dengan Metode Haar Feature Selection**, *Jurnal Teknologi Informasi Universitas Lambung Mangkurat (JTIULM)* 4 (1), 7-12. 2019
- [5] R. Purwanti, G. Ariyanto, **Pengenalan Wajah Manusia Berbasis Algoritma Local Binary Pattern**, *Jurnal Emittor* Vol. 17 No. 02, 2017
- [6] Septian Adi Wijaya, **Perbandingan Metode Pengenalan Wajah Secara Real Time Pada Perangkat Bergerak Berbasis Android**. *Jurnal Sains, Teknologi dan Industri*, Vol 14, No 1. 2016
- [7] Oktri Mohammad Firdaus, **Analisis Implementasi Global Positioning System (GPS) pada Moda Transportasi di PT. "X"**, *Seminar on Application and Research in Industrial Technology, SMART*, ISBN 978-602-97567-4-6
- [8] Hesti Rika, **Mudahnya Sistem 'Keyless' Mobil Dibobol Pencuri**, CNN Indonesia, Report Number : 22, 2019
- [9] K. Mistry, A. Saluja, **An Introduction to OpenCV using Python with Ubuntu**, *International Journal of Scientific Research in Science, Engineering and Information Technology*, Vol. 1, No. 2, pp. 65-68, 20
- [10] Ruihu Wang, **AdaBoost for Feature Selection, Classification and Its Relation with SVM\***, *International Conference on Solid State Devices and Materials Science*, Physics Procedia 25 ( 2012 ) 800 – 807
- [11] J. Kaur, A. Sharma, **Performance Analysis of Face Detection by using Viola-Jones algorithm**, *International Journal of Computational Intelligence Research*, ISSN 0973-1873 Volume 13, Number 5 (2017), pp. 707-717
- [12] D. A. Prasetya, I. Urviyanto, **Deteksi Wajah Metode Viola Jones Pada OpenCV menggunakan Pemrograman Python**, *Simposium Nasional RAPI XI FT UMS*, ISSN : 1412-9612

- [13] A. A. S. Gunawan, R. A. Prasetyo, **Face Recognition Performance in Facing Pose Variation**, *CommIT (Communication & Information Technology) Journal* 11(1), 1-7, 2017
- [14] G. Bradski, A. Kaehler, **“Learning OpenCV”**, O’Reily Media, Inc (U. S. America), pp. 202-203, 2008
- [15] Wan Ulfa Nur Zuhra, **“Jakarta, Kota dengan 9 Kasus Pencurian Setiap Hari”**, Tirto.id, Report Number: -, 2019
- [16] D. Bradley, G. Roth, **“Adaptive Thresholding Using the Integral Image”**, *Journal of Graphics Tools*, Volume 12, Issue 2, pp 13-21. 2007. NRC 48816.
- [17] B. S. B. Dewantara, J. Miura, **“A Robust Illumination Invariant Face Recognition System and Its Implementation”**, *Machine Vision and Applications*, vol. 27, no. 6, pp. 877-891, 2016.