IT Journal Research and Development (ITJRD)
Vol.7, No.1, August 2022, E-ISSN : 2528-4053 | P-ISSN : 2528-4061
DOI : 10.25299/itjrd.2022.9958

Analysis of Combination of Parking System with Face Recognition and QR Code using Histogram of Oriented Gradient Method (Case Study: Institut Teknologi Sumatera)

Ilham Firman Ashari[1], Mohamad Idri[2], M. Anas Nasrulah[3]
Department of Informatics Engineering, Institut Teknologi Sumatera
Firman.ashari@if.itera.ac.id[1], Mohamad.idris@if.itera.ac.id[2], manas.14117028@student.itera.ac.id[3]

Article Info

Article history:

Received Jul 03, 2022
Revised Jul 14, 2022
Accepted Aug 25, 2022

Keyword:

Security
Face Recognition
Dlib
HOG Method
QR Code

ABSTRACT

Security is very important everywhere, including in the campus environment. To provide security and comfort for those who park their vehicles, a parking application is needed that can provide vehicle security while undergoing academic activities on campus. QR code (Quick Response Code) is a technology for converting written data into a two-dimensional code, which is printed on a more compact medium capable of storing various types of data. The most common individual part used to identify a person is a face because it has the unique characteristics of everyone. Histogram of Oriented Gradient (HOG) is a feature extraction used for face identification based on the histogram of gradient orientation and gradient magnitude. This application is implemented using the Dlib library for facial recognition. The implementation of this method is expected to improve parking security and provide a record of parked vehicles. The results of testing the implementation of facial recognition methods into android applications show very satisfactory results. With the results of testing the QR code scanning accuracy of 100% and an accuracy of 90% for a 7% damage rate and an accuracy of 85% for a 15% damage rate, and the results of facial recognition testing of 90% on face photos wearing helmets and an accuracy of 92% on a photo of the face without a helmet. The test results comparing the old and new systems show that there is an increase in efficiency of 32%.

Corresponding Author:

Ilham Firman Ashari
Departement of Informatics
Institut Teknologi Sumatera
Jl. Terusan Ryacudu, Lampung Selatan, Indonesia
Email: firman.ashari@if.itera.ac.id

1. INTRODUCTION

Technological developments that continue to grow over time have influenced civilizations that allow many jobs to be completed quickly and efficiently. Along with the development of technology today, many companies are interested in using technology to increase their effectiveness in doing work. One of them is a university in terms of parking. One of the problems that often occur in universities is the parking system.

Parking is a condition of a stationary or immobile vehicle that is temporary because it is abandoned by the driver [1]. Without a good parking system, it can cause various problems. Common problems with the parking system are related to security issues and long queue waiting times when exiting the parking loT [2]. As is the case with the parking system in buildings C and D of the Institut Teknologi Sumatera, which currently uses a manual checking system by showing a Vehicle Registration Certificate (STNK), as well as the absence of recording of incoming and outgoing vehicles, which makes it difficult to monitor parked vehicles. Of course, this can cause loss problems and complicate the checking process when the driver forgets or does not bring the vehicle registration certificate.

For that, we need an application that can overcome these problems [3][4]. It takes an android application that can record vehicle data and facilitate the process of checking the vehicle when the driver is about to get out of the parking lot. So, the researchers have the idea to design and build an android application by utilizing Quick Response (QR) Code technology for the vehicle checking process and Face Recognition (Face Recognition) to identify the driver through his face. The selection of the QR code and the use of the android application in this study were to solve problems at a low cost. QR code is a technique for converting written data into codes in the 2-dimensional form that are printed into a more concise medium [5]. QR codes can store all types of data, such as numeric, alphanumeric, and binary data. By using a QR code, the inspection process time when leaving the parking lot becomes more efficient [6]. The QR code with an error correction rate of up to 30% (High) has a physical damage tolerance of about 10% [7]. Face Recognition (face recognition) is an identification technique in biometric technology to identify a person through his face as the main parameter [8]. In general, the face recognition process has two stages. The first stage is face detection, which is the stage in searching for human faces to ensure the presence of a detected face. The second stage is face recognition, namely the face identification stage by comparing the results of the extraction of detected facial features with faces that have been stored in the database.

This research is a development of Arief Budiman and Joko Triono's research which developed an Android-based QR code scanning parking application used by parking attendants which refer to security because the entire process of entering and leaving the vehicle will be scanned with a QR code [9]. Adlan et al [10], in their research, developed a parking application using a QR code with a validation or scanning process carried out twice when entering or leaving the parking lot. Scanning is carried out on the QR code located on the student card or employee card for driver identification and the QR code affixed to the vehicle for vehicle identification, then both data are stored in the database. When the driver exits the parking lot, a check of the suitability of the driver and vehicle data will be carried out. The time required for the process of scanning two QR codes (driver and vehicle) is 10 seconds. Subsequent research by Saeful Bahri and Heri Kusindaryadi [11]. Their research aims to create an attendance system with face recognition using the Histogram of Oriented Gradient (HOG) method. The system will detect student faces using the HOG method by taking pictures through raspberries. The system will perform face matching on the database and send the facial recognition results to a PC/laptop. When the shooting process has a student's face that is not detected, the system will take pictures and match the face to the database again until the student's face is detected according to the registered face database. With the results of testing the system can run well when the minimum lighting level is 80 lux and the maximum 300 lux and the distance of students from the raspberry camera is a maximum of 2 meters and the minimum distance is 10cm. The last research was conducted by Ilham et al [12], where in his research he made a prototype of a parking system using Face Recognition and ESP32 CAM, facial recognition was carried out using Amazon Web Service. The results of the study show the system can work effectively and an increase of 21%.

Then in this study, the author combines QR code technology with face recognition for vehicle and driver identification using the Histogram of Oriented Gradient (HOG) method. The histogram of Oriented Gradient (HOG) method is a feature extraction that can be used to detect faces by calculating the gradient orientation in a face image. Data is important for processing in this research [13][14], the data used is image data as many as 80 data. In its development, this research will use the help of the Dlib library to implement face recognition in the application to be developed. This application requires the user to register by filling in the required data first and pasting the generated QR code on the vehicle that has been obtained from the application, then when the user will enter the parking lot, he must scan the QR code provided. QR Code is one of the security methods used as credentials for users [15]. In its use, the data is stored in database firebase realtime database, Firebase Realtime Database is an online database as a data storage medium for applications that can be used to store application data in JSON form and can synchronize in real time to each connected client [16]. Then when going out of the parking lot, the officer or security guard will scan the QR code on the vehicle and then scan the driver's face, if the face is identified by the database, it is allowed to leave the parking lot. With this application, it is hoped that motorized vehicle users can easily park and provide control for parking security in the ITERA area [17].

2. RESEARCH METHOD

2.1. Face Detection Stages

Face detection is a process to detect human faces in computer technology by determining the position and size of the human face in a digital image. This technology can detect faces through facial characteristics and human facial characteristics by ignoring other objects such as the human body itself [18]. At this stage of detecting faces with Dlib, the shape predictor model of 5 face landmarks is provided and has been trained by Dlib.

2.2. Steps of Feature Extraction with Histogram of Oriented Gradient (HOG)

At the feature extraction stage, it is a process to determine the characteristics of an image that can distinguish faces from one another. In the feature extraction process with the Histogram of Oriented Gradient (HOG), the gradient and orientation of the gradient will be calculated and then converted into a histogram which will produce an image feature vector 16]. An example of a HOG extraction calculation is as follows, for example, a matrix A with dimensions of 8 x 8 which represents a face image.

$$A = \begin{bmatrix} 49 & 148 & 172 & 61 & 104 & 39 & 70 & 83 \\ 153 & 51 & 87 & 111 & 83 & 98 & 40 & 175 \\ 56 & 122 & 59 & 76 & 83 & 55 & 100 & 91 \\ 90 & 42 & 54 & 47 & 45 & 71 & 83 & 39 \\ 77 & 101 & 71 & 153 & 102 & 165 & 43 & 102 \\ 82 & 89 & 134 & 172 & 65 & 65 & 55 & 62 \\ 114 & 74 & 48 & 53 & 87 & 56 & 101 & 160 \\ 107 & 39 & 73 & 66 & 57 & 88 & 170 & 153 \end{bmatrix}$$

The next process is to calculate the horizontal and vertical gradients of the matrix using equation 1 for the horizontal gradient and equation 2 for the vertical gradient.

$$f_x(a,b) = I(a+1,b) - I(a-1,b) \quad (1)$$

$$f_y(a,b) = I(a,b+1) - I(a,b-1) \quad (2)$$

An example calculation using pixel 51 at coordinates 1.1 is as follows:

fx = 87 – 153 = -66

fy = 148 – 122 = 26

So that the results of the calculation of the horizontal gradient (fx) and vertical gradient (fy) are obtained as follows:

$$fx = \begin{bmatrix} 65 & 123 & -87 & -68 & -22 & -34 & 44 & -21 \\ -124 & -66 & 60 & -4 & -13 & -43 & 77 & -113 \\ -31 & 3 & -46 & 24 & -21 & 27 & 36 & -44 \\ 3 & -36 & 5 & -9 & 24 & 38 & -32 & 7 \\ -1 & -6 & 52 & 31 & 12 & -59 & -63 & 34 \\ 27 & 52 & 83 & -69 & -107 & -10 & -3 & 27 \\ -86 & -66 & -21 & 39 & 3 & 14 & 104 & 13 \\ -114 & -34 & 27 & -16 & 22 & 113 & 65 & -63 \end{bmatrix}$$

$$fy = \begin{bmatrix} -46 & -12 & -14 & -45 & -26 & -10 & 130 & -22 \\ -7 & 26 & 113 & -15 & 21 & -16 & -30 & -8 \\ 63 & 9 & 33 & 64 & 38 & 27 & -43 & 136 \\ -21 & 21 & -12 & -77 & -19 & -110 & 67 & -11 \\ 8 & -47 & -80 & -125 & -20 & 6 & 28 & -23 \\ -37 & 28 & 23 & 100 & 15 & 109 & -58 & -58 \\ -25 & 50 & 61 & 106 & 8 & -23 & -115 & -91 \\ 65 & -74 & -124 & -8 & -17 & 17 & 31 & 77 \end{bmatrix}$$

Then after obtaining the gradient, the next step is to calculate the magnitude (m) with equation 3 and the angle value (θ) with equation 4 as follows:

$$\mathrm{m(a,b)} = \sqrt{(a)^2 + (b)^2} = \sqrt{(-66) + 26^2} = 70{,}94 \quad (3)$$

$$\theta = tan^{-1}\left(\frac{26}{66}\right) = 21{,}50143 \quad (4)$$

So that the magnitude (m) and angle value (θ) are obtained from the calculation results which will be used to determine the direction of bin orientation can be seen in figure 1.

$$m = \begin{bmatrix} 79{,}6304 & 124{,}1974 & 70{,}21396 & 21{,}2132 & 8{,}062258 & 45{,}80393 & 89{,}56004 & 131{,}2288 \\ 123{,}584 & 70{,}93659 & 9{,}486833 & 41{,}67733 & 47{,}38143 & 59{,}05929 & 82{,}80097 & 81{,}43709 \\ 88{,}11924 & 127{,}9414 & 56{,}61272 & 13 & 95{,}41488 & 86{,}12781 & 64{,}51356 & 126{,}9055 \\ 81{,}5414 & 15{,}52417 & 68{,}35203 & 77{,}52419 & 128{,}7866 & 121{,}4949 & 112{,}9469 & 17{,}88854 \\ 34{,}05877 & 24{,}69818 & 43{,}41659 & 30{,}61046 & 23{,}32381 & 108{,}0463 & 8{,}544004 & 27{,}80288 \\ 35{,}44009 & 45{,}88028 & 38{,}18377 & 116{,}3787 & 59{,}3043 & 109{,}4578 & 26{,}92582 & 114{,}2716 \\ 137{,}2443 & 82{,}63776 & 56{,}0803 & 74{,}24958 & 68{,}942 & 58{,}07753 & 155{,}0516 & 72{,}01389 \\ 30{,}41381 & 113{,}2828 & 142{,}9405 & 13{,}0384 & 41{,}04875 & 63{,}97656 & 91{,}92388 & 99{,}48869 \end{bmatrix}$$

$$\theta = \begin{bmatrix} 35{,}28675 & 3{,}23101 & 63{,}79989 & 81{,}8699 & 82{,}87498 & 53{,}88066 & 16{,}20902 & 29{,}69073 \\ 5{,}572198 & 21{,}50143 & 71{,}56505 & 30{,}25644 & 82{,}725 & 28{,}30076 & 37{,}14669 & 65{,}32314 \\ 9{,}14164 & 62{,}03288 & 35{,}65533 & 67{,}38014 & 56{,}97613 & 15{,}4885 & 71{,}00335 & 77{,}71604 \\ 33{,}49518 & 75{,}06858 & 69{,}44395 & 33{,}33334 & 76{,}07166 & 55{,}39432 & 69{,}80019 & 26{,}56505 \\ 49{,}76364 & 58{,}24052 & 61{,}07357 & 38{,}36749 & 59{,}03624 & 7{,}980114 & 69{,}44395 & 37{,}69424 \\ 16{,}38954 & 20{,}40988 & 45 & 70{,}9423 & 5{,}806727 & 84{,}75818 & 58{,}67131 & 8{,}555558 \\ 71{,}30102 & 21{,}28641 & 50{,}06362 & 64{,}4703 & 23{,}96249 & 87{,}03906 & 47{,}87545 & 25{,}49755 \\ 46{,}33222 & 4{,}049582 & 72{,}07208 & 57{,}52881 & 34{,}0772 & 65{,}03721 & 81{,}8699 & 50{,}71059 \end{bmatrix}$$

Magnitude				43,41659					
Bin	0	20	40	60	80	100	120	140	160

Figure 1. The Magnitude and Angle Nilai Value

The results of the bin orientation that has been determined from the magnitude (m) and the angle value (θ) are as follows:

Magnitude	94,19336	61,8945	62,5782	70,53195	43,63153	43,64399	142,9405	137,2443	86,48639
Bin	0	20	40	60	80	100	120	140	160

From the results of the bin orientation, the histogram normalization will then be carried out using equation 5

$$V_i = \frac{v_i}{\sqrt{\|v\|_i^2 + \varepsilon^2}} \quad (5)$$

$$V_i = \frac{94{,}19336}{\sqrt{72150{,}51055}} = \frac{94{,}19336}{268{,}6084707} = 0{,}350671593$$

So that the vector results from normalization have been carried out as follows:

V = [(0,350671593), (0,230426481), (0,232971813), (0,262582757), (0,162435414), (0,162481815), (0,532152045), (0,510945568), (0,321979385)]

The vector histogram above can be used for classification by calculating the Euclidean distance of the 2 vectors generated by the target face and the driver's face during facial scanning.

2.3. Face Recognition Classification Stages

The step after feature extraction is faced recognition classification by calculating Euclidean Distance from the vector generated from feature extraction. At this stage, we will only compare 2 faces, namely the target face in the database and the driver's face which was taken during facial scanning. With the same classification of faces, if the resulting Euclidean Distance is less than 0.6 and if it is more than 0.6 then the faces are categorized as not the same. With the implementation of the application using the dlib face recognition resnet model v1 which has been trained and provided by Dlib [20]. The determination of the Euclidean distance of 0.6 here is based on the model used in the face recognition process. In the following, 3 examples of histogram vectors are given, with vector 1 as the face that will be used as the comparison face data.

V_1 = [(0,045418263), (0,032463761), (0,097873388), (0,38916587), (0,053491575), (0,012272767), (0,03528628), (0,209349295), (0,124678802)]
V_2 = [(0,350671593), (0,230426481), (0,232971813), (0,262582757), (0,162435414), (0,162481815), (0,532152045), (0,510945568), (0,321979385)]
V_3 = [(0,008509616), (0,015356646), (0,277300638), (0,545070513), (0,071915353), (0,006340938), (0,005221022), (0,04533381), (0,024951463)]

By using equation 6, it can be seen that the Euclidean distance between vector 1 and vector 2 is as follows:

$$d(p,q) = d(q,p) = \sqrt{\textstyle\sum_{i=1}^{n} (q_i - p_i)^2} \quad (6)$$

$$d(v_1, v_2) = \sqrt{0{,}577838641} = 0{,}760156985$$

By using equation 6, it can be seen that the Euclidean distance between vector 1 and vector 3 is as follows:

$$d(v_1, v_3) = \sqrt{0{,}096280461} = 0{,}310290929$$

From the results of the Euclidean distance calculations that have been carried out, it can be concluded that vector 3 is categorized the same as vector 1 because of the results of the Euclidean distance <0.6. While vector 2 is categorized as not the same or different because the results of the Euclidean distance are> 0.6.

2.4. Architectural Design

The system architecture design is used to provide an overview of the application workflow. The following is a picture of the system architecture design that will be developed when the driver enters the parking lot:

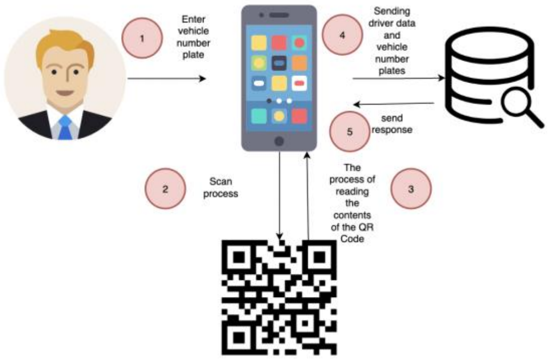

Figure 1. The architectural design of the parking lot entry system

Figure 2 is a design when the driver enters the parking lot where the driver has installed the application on his smartphone. The driver inputs the vehicle number plate and then scans the QR code available in the parking lot, then the data will be sent and stored in the database. However, if the driver has not registered his vehicle, he can register the vehicle first or if the driver brings another person's vehicle, he can enter the vehicle number plate after scanning the QR code. The data sent and stored in the database includes the vehicle number plate as the main data and the identity of the driver. Figure 3 is the process flow when the driver enters the parking location:

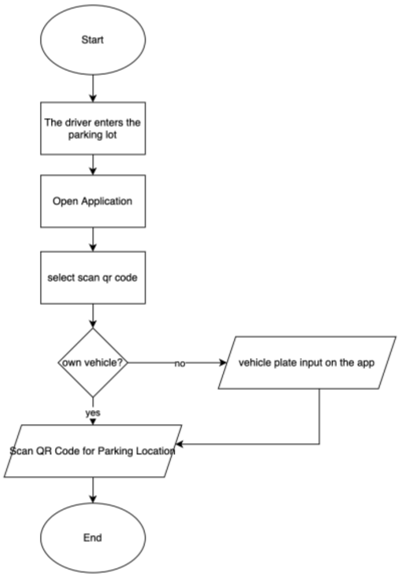

Figure 2. Flowchart of drivers entering the parking lot

Figure 3 describes the process when a driver enters a parking location. The driver can scan the QR code found at the parking location and the vehicle number plate and driver's identity will be recorded in the database, if the vehicle used by the user is not his vehicle, the user can input the vehicle number plate before scanning the QR code. If the driver does not have a user application account, he can register the user first. The process flow when the driver registers a user can be seen in figure 4:

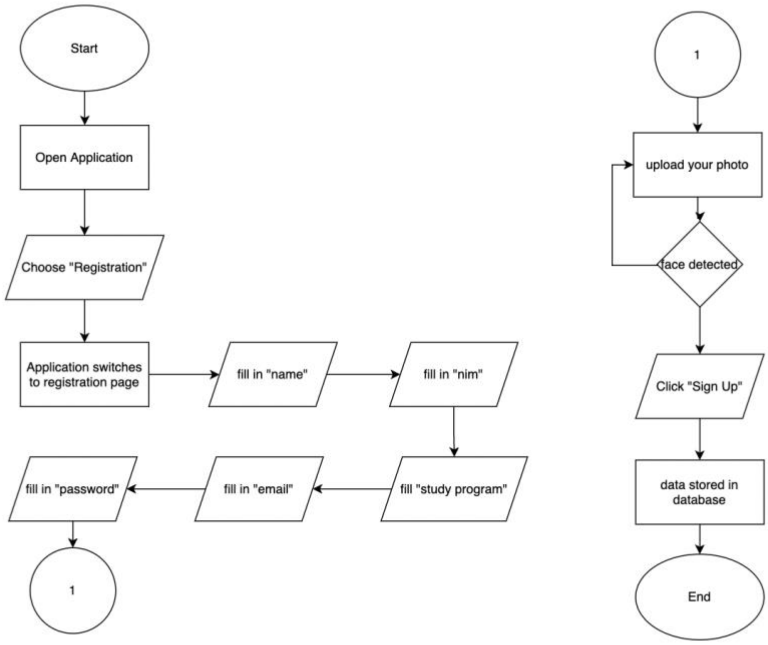

Figure 3. User registration flowchart

Figure 4 describes the flow of the user registration process. Users can register themselves by selecting "Registration" on the login page. Then the user can fill in their identity such as name, NIM, study program, email, password, and upload user photos. In the process of uploading a user's photo, a face detection check will be carried out using the Histogram of Oriented Gradient (HOG) method whether the photo contains a detected face. If a face is detected in the photo, it can continue registration, but if it is not detected, the user must take a photo and re-upload the photo. After all, data has been filled in, it is continued by pressing the "Signup" button to process registration, and user data is stored in the database. After registering an account on the user application, drivers can register their vehicles independently. The following is a picture of the process flow when the driver registers his vehicle:

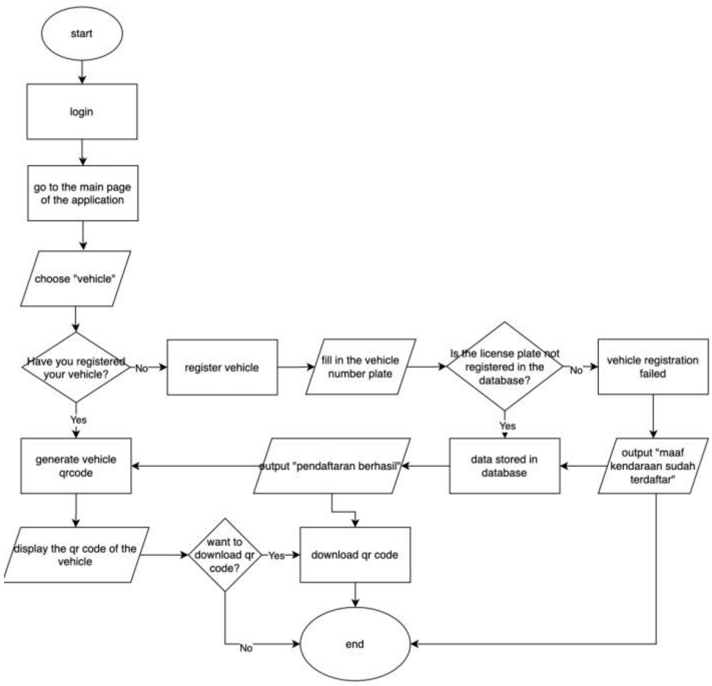

Figure 5. Vehicle registration flowchart

Figure 5 describes the flow of vehicle registration. Users can register their vehicles through the vehicle page. If the user has registered his vehicle, the application will display the generated QR code and the user can download the QR code. If the user has not registered his vehicle, the application will lead to the vehicle registration page, where the user can fill in the number plate and vehicle name data. Then the application will check the vehicle number plate in the database, if it is not registered then the vehicle registration is successful and if the vehicle number plate is registered then the registration fails. Then the application will generate a QR code and display it on the application, and the user can download the QR code to be attached to the vehicle. The following is a drawing of the system architecture design that will be developed when the driver exits the parking lot:

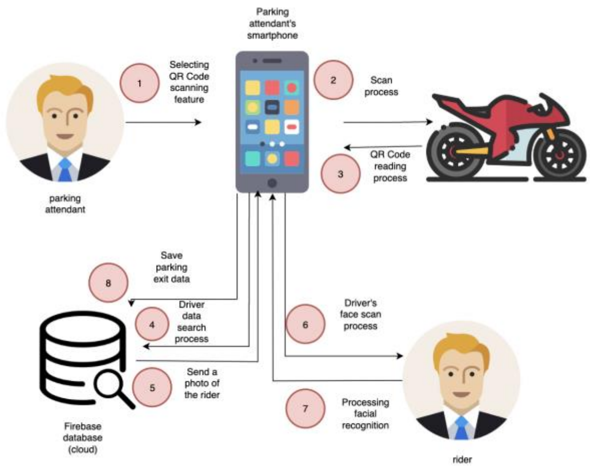

Figure 6. The architectural design of the parking lot driver system

Figure 6 is the design when the driver exits the parking lot. The parking attendant will scan the QR code that has been affixed to the vehicle or if the driver has not pasted the QR code on the vehicle, the driver can show the QR code in the application, the QR code on the vehicle is a representation of the vehicle's license plate. After scanning the vehicle's QR code, the result of reading the QR code in the form of a vehicle number plate will be sent to the database and look for the appropriate data when the vehicle enters the parking lot, then the database sends photo data of the driver's face to the officer's smartphone. Then the parking attendant can scan the driver's face to check whether the driver is the one who brought the vehicle in. The application will display the results of the facial scan, the driver can be asked to leave if the identity of the driver is the same as when entering the parking lot and the exit data will be stored in the database. The flow of the process for motorists leaving the parking lot can be seen in figure 7:

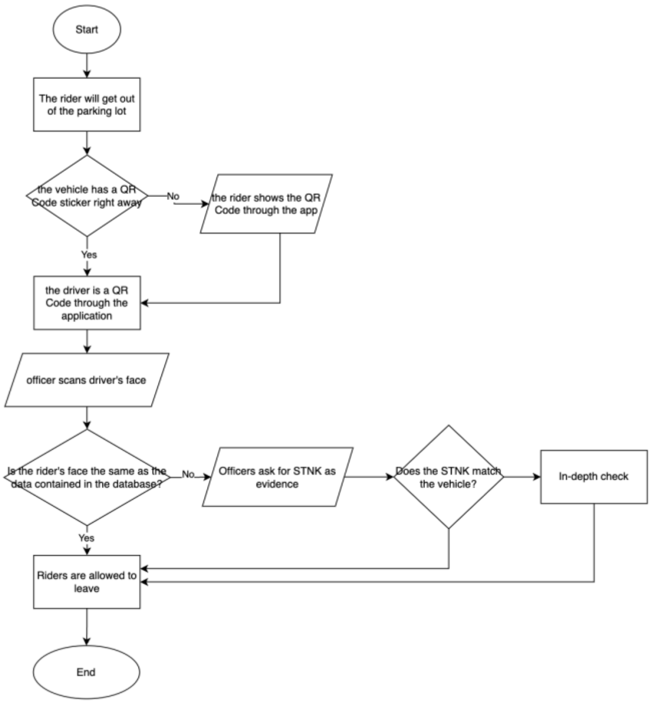

Figure 7. Flowchart of drivers leaving the parking lot

Figure 7 describes the process flow when the driver will leave the parking location. When leaving the parking location, the parking attendant will scan the QR code attached to the vehicle to check the vehicle. If the vehicle's QR code has not been affixed to the vehicle, the user can show the vehicle's QR code through the application. After the officer scans the vehicle's QR code, the application will immediately direct to the driver's face scan page to check whether the face of the driver is the same as the face that brought the vehicle into the database. If the identity of the driver is the same as when he entered the parking location, the parking attendant will invite the driver to leave the parking location. However, if the identity of the driver is not the same as when he entered the parking location, further checking is necessary, such as checking the vehicle registration certificate.

3. RESULTS AND ANALYSIS (11 Pt)

3.1. Implementation of Parking Lot Entry System Design

The system workflow process when the driver has entered into a pre-designed parking space. This process is carried out by the driver using a user application. After the driver enters the parking lot, the driver can scan the parking QR code available at the parking location. However, before scanning the parking QR code, the driver must select the vehicle number plate, if the vehicle used does not belong to the driver, the driver can fill in the vehicle number plate in the column provided. The process of selecting or filling in vehicle number plates can be seen in Figure 8.

Figure 8. Choosing the number plate of the vehicle used

Figure 9. Scan parking QR code

The parking QR code scan is carried out after the driver selects or fills in the vehicle number plate used. Drivers can scan the parking QR code available at the parking location. After the driver scans the QR code, the driver data and the vehicle used will be stored in the Firebase Database and the data can be viewed through the parking history page in the user application. The parking QR code scanning process can be seen in Figure 9.

Figure 10. Incorrectly scanned parking QR code

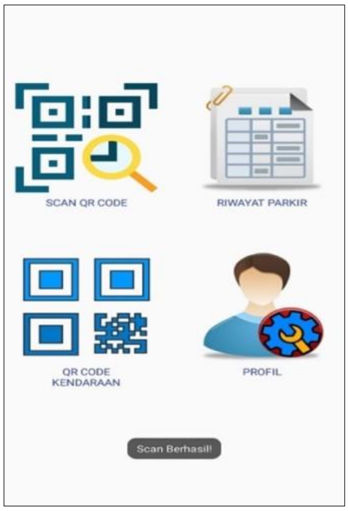

Figure 11. The parking QR code scan was successful

When the scanned QR code is not a parking QR code or another QR code, the application will give a notification that the QR code is incorrect as shown in Figure 10. While in Figure 11 display when the driver successfully scans the parking QR code.

3.2. Implementation of Parking Lot Exit System Design

The process of scanning the vehicle when the driver leaves the parking lot is carried out by the parking attendant using the admin application with a pre-designed process flow. The officer can scan the vehicle's QR code that has been affixed to the vehicle, if the vehicle's QR code has not been affixed to the vehicle, the officer can ask the driver to show the vehicle's QR code through the user application. The vehicle scanning process can be seen in Figure 12.

Figure 12. Scanning vehicle QR code

Figure 13. Driver's face scan

The admin application will redirect to the face scan page when the QR code scan is successful as shown in Figure 13. The rider's face scan is done by pointing the smartphone at the rider's face to take a photo of the rider. After the photo-taking process, the application will process and compare the photo with the driver's photo contained in the database. The facial recognition process uses the Histogram of Oriented Gradient (HOG) method with the help of the Dlib library and the dlib face recognition resnet model v1 which was previously trained by Dlib.

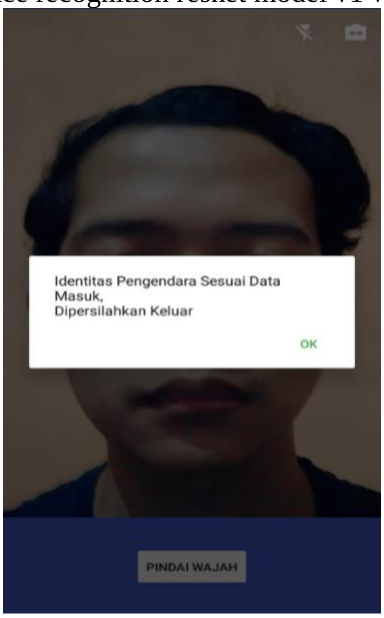

Figure 14. The face of the rider is recognized and according to the data entered

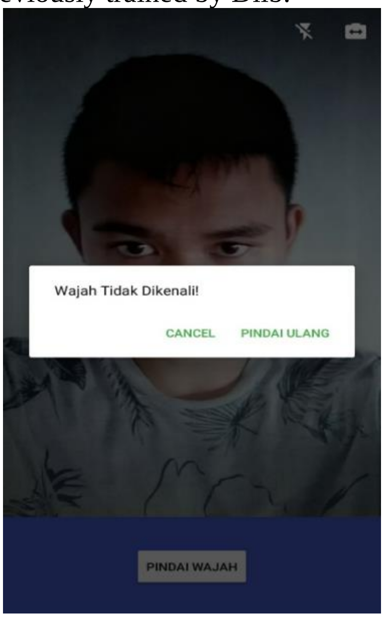

Figure 15. The face of the rider is not recognized or does not match

Figure 14 shows that the face of the driver is the same as the face of the driver who brought the vehicle in. When the driver's face being scanned is not the driver who brought in the vehicle, the application will not recognize the driver's face, and the officer can re-scan or return to the QR code scanning page by pressing the "cancel" button on the pop-up dialog that is displayed. The process of scanning the face is not recognized can be seen in Figure 15.

3.3. *QR Code Scanning Accuracy Test*

This QR code scanning accuracy test is carried out to determine the level of accuracy of normal QR code scanning without any damage and the accuracy of QR code scanning with damage. Testing the QR code under normal conditions uses 20 vehicle QR code data and testing the level of damage using a QR code that has been damaged by covering part of the QR code as shown in Figure 16.

Figure 16. Example of a QR code that has been damaged

The QR code scanning process in this test was carried out several times to get maximum results. Testing the QR code that has been damaged is used to determine the ability of the error correction level of the QR code to damage that can still be read or scanned. The level of damage used in this test is 7%, 15%, 25%, and 30% according to the error correction level of the QR code. The results of testing the accuracy of QR code scanning with and without damage can be seen in Table 1 below.

Table 1. The results of the QR code scanning accuracy test

Testing	True	False	Accuracy
QR code without damage	20	0	100%
7% damage	18	2	90%
15% damage	17	3	85%
25% damage	5	15	25%
30% damage	2	18	10%

To reduce the possible risk of damage to the QR Code, the motorcyclist must be affixed with a sticker of good quality and anti-rain.

3.4. Facial Recognition Accuracy Test

Testing the accuracy of face recognition applications using two types of photos, namely photos of human faces and photos using helmets. The data used consists of 50 photos of human faces with 50 different individuals and 30 photos of faces using a helmet with an open helmet. With each individual having 2 photos, the first photo is used as comparison data in the database, the second photo will be used as facial recognition through facial scanning with a smartphone camera. The second photo or photo taken by facial scanning through the camera will be compared with the first photo contained in the database. The face scanning test process can be seen in Figure 17 and Figure 18.

Figure 17. Face scan test

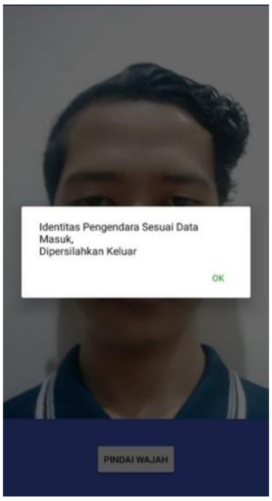

Figure 18. The face scan test was successful

Figure 19. Face scan test wearing a helmet

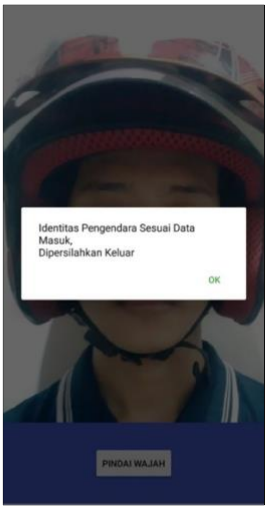

Figure 20. Face scan test wearing a helmet is successful

Figure 19 and Figure 20 are face scan tests wearing a helmet. The face recognition process is carried out on a smartphone using a face photo taken from the database according to the rider's data with a face photo taken through a face scan on the application. The app will compare the two photos and determine if they have the same face. In the face recognition testing process, this is done several times until the face is recognized by the application. The results of the facial recognition accuracy test can be seen in Table 2.

Table 2. The results of the facial recognition accuracy test

Test Result	Photo of the face without a helmet	Photo of a face wearing a helmet
True	46	27
False	4	3
Total	50	30
Accuracy	92%	90%

3.5. *Discussion of QR code Scanning Accuracy Test Results*

Based on the QR code scanning accuracy test that has been carried out in the previous subsection, QR code scanning can read QR codes well. By calculating the accuracy using equation 7.

$$Accuration = \frac{correct\ result}{number\ of\ tests} \times 100\% \tag{7}$$

The accuracy value is 100% for a normal QR code or without damage, and the accuracy of the QR code with a damage rate of 7% has an accuracy of 90% and a damage rate of 15% has an accuracy of 85%. However, the 25% and 30% damage levels have very low accuracy, namely 25% and 10%. From the results of the QR code scanning test with damage, it is known that the QR code can still be scanned and read properly when the QR code damage level is 15% and a QR code with damage above 15% has a very small possibility to be scanned and read properly. The small accuracy of scanning the QR code can be caused by the amount of damage received by the QR code, the greater the damage, the more data will be lost in the QR code.

3.6. Time-related test

Tests related to scanning and response times, by comparing the conditions of the manual and automatic parking systems. The test results can be seen in table 3.

Table 3. Scan and response time test results

	No.	Entry	Estimate Time	Out	Estimate Time
Previous System (Manual Checking)	1.	Data 1	9 seconds	Data 1	18 seconds
	2.	Data 2	4 seconds	Data 2	19 seconds
	3.	Data 3	7 seconds	Data 3	18 seconds
	4.	Data 4	6 seconds	Data 4	20 seconds
	5.	Data 5	7 seconds	Data 5	18 seconds
	6.	Data 6	8 seconds	Data 6	22 seconds
	7.	Data 7	7 seconds	Data 7	25 seconds
	8.	Data 8	6 seconds	Data 8	24 seconds
	9.	Data 9	6 seconds	Data 9	23 seconds
	10.	Data 10	7 seconds	Data 10	21 seconds
	11.	Data 11	5 seconds	Data 11	28 seconds
	12.	Data 12	5 seconds	Data 12	18 seconds
	13.	Data 13	7 seconds	Data 13	19 seconds
	14.	Data 14	9 seconds	Data 14	20 seconds
	15.	Data 15	8 seconds	Data 15	21 seconds
Designed System	1.	Data 1	5 seconds	Data 1	15 seconds
	2.	Data 2	5 seconds	Data 2	17 seconds
	3.	Data 3	5 seconds	Data 3	15 seconds
	4.	Data 4	5 seconds	Data 4	13 seconds
	5.	Data 5	5 seconds	Data 5	12 seconds
	6.	Data 6	5 seconds	Data 6	15 seconds
	7.	Data 7	5 seconds	Data 7	13 seconds
	8.	Data 8	5 seconds	Data 8	15 seconds
	9.	Data 9	5 seconds	Data 9	16 seconds
	10.	Data 10	5 seconds	Data 10	12 seconds
	11.	Data 11	5 seconds	Data 11	13 seconds
	12.	Data 12	5 seconds	Data 12	11 seconds
	13.	Data 13	5 seconds	Data 13	11 seconds
	14.	Data 14	5 seconds	Data 14	12 seconds
	15.	Data 15	5 seconds	Data 15	16 seconds

Entry Time (previous method) $= \frac{9+4+7+6+7+8+7+6+6+7+5+5+7+9+8}{15} = \frac{101}{15} = 6.73$ seconds

Entry Time (proposed method) $= \frac{5+5+5+5+5+5+5+5+5+5+5+5+5+5+5}{15} = \frac{75}{15} = 5$ seconds

Out Time (previous method) $= \frac{18+19+18+20+18+22+25+24+23+21+28+18+19+20+21}{15} = \frac{314}{15} = 20.93$ seconds

Out Time (proposed method) $= \frac{15+17+15+13+12+15+13+15+16+12+13+11+11+12+16}{15} = \frac{206}{15} = 13.73$ seconds

Total Time Difference Between old and new system = (20.93 + 6.73) - (5+13.73) = 27.66 - 18.73 = 8.93

effectiveness = 8.93 / 27.66

= 0.32 * 100%

= 32 %

The test results show that there is an increase in efficiency of 32%.

3.7. Discussion of the results of the facial recognition accuracy test

Based on the facial recognition accuracy test that has been carried out in the previous subsection, the application can recognize the rider's face quite well. By calculating the accuracy using equation 7. The test accuracy value is 90% for face photos wearing helmets with 30 facial photo data and an accuracy of 92% for photos of faces not wearing helmets with data as many as 50 photos of faces. From the results of the accuracy test that has been carried out, facial recognition can still recognize the rider's face well as long as the rider's face is not blocked by the helmet glass. The failure of facial recognition in testing wearing a helmet or without wearing a helmet is generally experienced by photos of faces wearing glasses, especially dark-colored glasses. It is evident from the test on photos that wear transparent glasses, they can still be recognized well, while some photos that wear dark glasses fail to be recognized by the application. From this analysis, the failure to recognize the target's face can be caused by the presence of objects or accessories such as dark-colored glasses that block the face such as the eyes, as well as the small resolution of the photo used so that it can affect the results of the gradient calculation obtained using this Histogram of Oriented Gradient.

4. CONCLUSION

Based on the research and testing that has been done on the QR code parking application and face recognition using the Histogram of Oriented Gradient (HOG) method, the following conclusions can be drawn:

1. This study succeeded in making a parking application using a QR code as vehicle and parking identification that can be implemented and run well in the scanning process and generate QR codes. With the results of the QR code scanning test accuracy of 100% and the QR code can still be scanned and read properly at a damage rate of 7% with an accuracy of 90% and a damage rate of 15% with an accuracy of 85%.
2. Face recognition was successfully implemented using the Histogram of Oriented Gradient method with the help of the Dlib library. The facial recognition process can run well even though the rider wears a helmet with the condition that the helmet glass must be open, the facial recognition accuracy test gets 90% accuracy for face photos wearing a helmet and 92% accuracy for face photos without wearing a helmet.
3. Efficiency testing showed an increase of 32% from the old system.

REFERENCES

[1] L. Gustiaji, D. Herianto, and Y. M. Hadi, “Studi Optimalisasi Perparkiran Dan Pedestrian

Di Fakultas Teknik Jurusan Teknik Mesin, Teknik Elektro, Teknik Kimia Dan Teknik Geofisika Universitas …," *J. Rekayasa Sipil Dan …*, vol. 4, no. 1, pp. 33–42, 2016, [Online]. Available: http://journal.eng.unila.ac.id/index.php/jrsdd/article/viewFile/344/pdf.

[2] A. R. Mazidah, A. Sumarsono, and D. Djumari, "Evaluasi Panjang Antrian Kendaraan Pada Pelayanan Pintu Keluar Parkir Di Hartono Lifestyle Mall Solo Baru," *Matriks Tek. Sipil*, vol. 3, no. 2, p. 485, 2015, [Online]. Available: https://jurnal.uns.ac.id/matriks/article/view/37204.

[3] I. F. Ashari, "Implementation of Cyber-Physical-Social System Based on Service Oriented Architecture in Smart Tourism," *J. Appl. Informatics Comput.*, vol. 4, no. 1, pp. 66–73, 2020, doi: 10.30871/jaic.v4i1.2077.

[4] I. F. Ashari, A. J. Aryani, and A. M. Ardhi, "DESIGN AND BUILD INVENTORY MANAGEMENT INFORMATION SYSTEM," vol. 9, no. 1, pp. 27–35, 2022.

[5] A. Norhadi, A. Marzuki, L. Wicaksono, and R. Addetya Yacob, "Studi Debit Aliran pada Sungai Antasan Kelurahansungai Andai Banjarmasin Utara," *J. Poros Tek.*, vol. 7, no. 1, pp. 1–53, 2015.

[6] A. Fernando Sianipar and N. Lutfiani, "Sistem Parkir Berbasis QR Code Pada Perguruan Tinggi Raharja," *Semin. Nas. Multi Disiplin Ilmu*, vol. 1, no. November 2017, pp. 43–54, 2017, [Online]. Available: https://www.researchgate.net/publication/331429646.

[7] E. F. Nurdiansyah, I. Afrianto, and S. V. Tiket, "IMPLEMENTASI QRCODE SEBAGAI TIKET MASUK EVENT DENGAN MEMPERHITUNGKAN TINGKAT KOREKSI KESALAHAN," *J. Teknol. dan Informasi2*, vol. 7, no. 2, pp. 25–44, 2017.

[8] B. M. Susanto, F. E. Purnomo, and M. F. I. Fahmi, "Sistem Keamanan Pintu Berbasis Pengenalan Wajah Menggunakan Metode Fisherface," *J. Ilm. Inov.*, vol. 17, no. 1, 2017, doi: 10.25047/jii.v17i1.464.

[9] A. Budiman and J. Triono, "Sistem Informasi Parkir Kendaraan Bermotor Berbasis Android," *J. Ilm. Ilmu-Ilmu Tek. (Pilar Teknol.*, vol. 1, no. 1, p. 42, 2016.

[10] A. B. Pradana, C. Ma'rifadiyah, D. Jatinugroho, and F. Z. Abidin, "Perancangan Sistem Perparkiran Rendah Biaya Berbasis Ponsel Cerdas Android," *J. Tek. Elektro*, vol. 11, no. 1, pp. 31–35, 2019, doi: 10.15294/jte.v11i1.21202.

[11] S. Bahri and H. Kusindaryadi, "Rancang Bangun Pemantauan Absensi Mahasiswa dengan Menggunakan Sidik Wajah secara Simultan Melalui CCTV Ruang Kelas," *Resist. (elektRonika kEndali Telekomun. tenaga List. kOmputeR)*, vol. 3, no. 1, p. 37, 2020, doi: 10.24853/resistor.3.1.37-44.

[12] I. F. Ashari, M. D. Satria, and M. Idris, "Parking System Optimization Based on IoT using Face and Vehicle Plat Recognition via Amazon Web Service and ESP-32 CAM (Case Study : Institut Teknologi Sumatera)," vol. 11, no. 2, pp. 137–153, 2022.

[13] I. F. Ashari and V. Adhelia, "Expert System and IoT for Diagnose of Feline Panleukopenia Virus Using Certainty Factor," *Matrik J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 21, no. 2, pp. 451–462, 2022, doi: 10.30812/matrik.v21i2.1517.

[14] I. F. Ashari, M. Alfarizi, M. N. K, and M. A. H, "Vulnerability Analysis and Proven On The neonime . co Website Using OWASP ZAP 4 and XSpear," *J. Teknol. Komput. dan Sist. Inf.*, vol. 5, no. 2, pp. 75–81, 2022.

[15] I. F. Ashari, "Graph Steganography Based On Multimedia Cover To Improve Security and Capacity," in *2018 International Conference on Applied Information Technology and Innovation (ICAITI)*, 2018, no. April 2019, pp. 194–201.

[16] E. A. W. Sanadi, A. Achmad, and Dewiani, "Pemanfaatan Realtime Database di Platform Firebase Pada Aplikasi E-Tourism Kabupaten Nabire," *J. Penelit. Enj.*, vol. 22, no. 1, pp. 20–26, 2018, doi: 10.25042/jpe.052018.04.

[17] I. F. Ashari, A. G. Manalu, and R. Setiawan, "Analysis of Security Guard Scheduling System Using Genetic Algorithm and Tournament Selection (Case Study : Institut Teknologi Sumatera)," vol. 5, no. 2, pp. 202–207, 2021.

[18] M. B. Pranoto, K. N. Ramadhani, and A. Arifianto, "Face Detection System Menggunakan Metode Histogram of Oriented Gradients (HOG) dan Support Vector Machine (SVM)

Face Dtection System using Histogram of Oriented Gradients (HOG) Method amd Support Vector Machine (SVM),” *e-Proceeding Eng.*, vol. 4, no. 3, pp. 5038–5045, 2017, [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/5441.

[19] Y. F. Munawaroh and I. Salamah, “Analisa Perbandingan Algoritma Histogram of Oriented Gradient (HOG) dan Gaussian Mixture Model (GMM) Dalam Mendeteksi Manusia,” *Semin. Nas. Inov. dan Apl. Teknol. Di Ind. 2018*, vol. 4, no. 2, pp. 251–255, 2018.

[20] D. E. King, “Dlib-ml: A machine learning toolkit,” *J. Mach. Learn. Res.*, vol. 10, pp. 1755–1758, 2009.