**Automatic Garage Door**

**Department of Computer Engineering San Jose State University,**

**San Jose, CA - 95192**

**Kaushal Karinaga Shetter Raju#1, Prince Sharzeel#2, Ninad Dhuri#3**

1[kaushal.karinagashetterraju@sjsu.edu](mailto:kaushal.karinagashetterraju@sjsu.edu),

2[prince.sharzeel@sjsu.edu](mailto:prince.sharzeel@sjsu.edu)

3[ninadsanjay.dhuri@sjsu.edu](mailto:ninadsanjay.dhuri@sjsu.edu)

***Abstract --* The number of devices that are able to interact with objects and connect to the internet is growing rapidly. It is estimated that the number of IoT devices will reach 50 billion by 2025. [1] The goal of this project is to develop garage door gates using the Internet of Things (IoT) and microcontroller-controlled technologies. The design of a prototype while adhering to the steps and practices of research and development constitutes the approach adopted in this study. The three stages of creating the control system are software design, hardware production, and software and hardware integration.**

***Keywords* -- Temperature, Humidity, IoT, Project, Software, Hardware, WiFi, Internet, Planet.**

1. **INTRODUCTION**

People are constantly working to develop technologically-based solutions that will make their daily lives easier. Because technology makes everything easier to do, it is intended to improve the quality of human life [2]. These findings support the growth of technology, which has produced numerous instruments to support human tasks and even take over for humans in some roles [3].

Humans want to do tasks easily because of the modern technology available, and one such easy thing is to use IoT. (Internet of Things). The Internet

of Things (IoT) idea attempts to increase the advantages of permanently connected internet connectivity [4].

In order to safeguard owned assets and privacy, a security system must be installed due to the increased crime rate, technological advancement,

and the current era. It is essential to deploy a security system since it can give people a sense of comfort and security and reduce crime, particularly theft, in the neighborhood.

Thefts like those committed in the garage by breaching the garage door are commonplace in residential settings. A garage door is a door that is often used for mounting entry and egress. Since security is of utmost importance, the key to the garage door is typically given by someone, but occasionally the key may be misplaced or made by careless individuals, and it is simple to pick the garage door lock. People desire a system that makes using the garage door easier and easier, in addition to security. because some individuals are too lazy, especially when it's raining, to control the garage door.

Even while operating the garage door manually, which is still common. One problem is when the owner of the home is too sluggish to exit the car to operate the garage door. Using a camera and a Wi-fi network this issue can be solved. As a result, we require a control mechanism that will open garage doors whenever needed. Wi-fi to send the information to the controller and a camera to capture the image or video of the car.

A radio receiver and transmitter are commonly used to accomplish automatic garage door opening. The user clicks a button on the transmitter once they are close to the garage, which sends a radio signal to the receiver within the garage. The receiver confirms the signal and opens the garage door. This system can be utilized by various users by installing numerous radio transmitters.



Figure 1. Automatic Garage Door

**II. LITERATURE**

Raspberry Pi are solo boards produced by the Raspberry Pi Foundation, an organization whose goals include fostering computer literacy and facilitating access to computer science education.

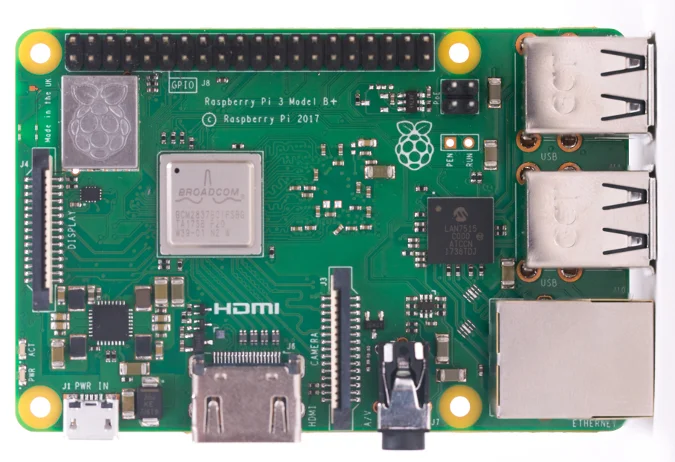


Figure 2. Raspberry Pi

Numerous versions and variations of the Raspberry Pi have been made available since its launch in 2012. The most recent Pi model has a quad-core processor running at over 1.5GHz and 4GB of RAM, whereas the original Pi had a single-core processor running at 700MHz and just 256MB of RAM. The cost of the Raspberry Pi has consistently been in the neighborhood of $100 (typically $35 USD), with the Pi Zero being the most basic model at just $5.



Figure 3. Raspberry Pi 3 and 2

The Pi 3's performance in 32-bit mode is about 50–60% better than the Pi 2's, and it is ten times quicker than the first Raspberry Pi with a single CPU. When video playback is improved by the chip's NEON engine, real-world apps will perform 2.5–20 times better than they did on the original Raspberry Pi.

Infrared cameras are frequently used in commercial license plate recognition systems, and they are filtered to remove the portion of the spectrum where the license plate emits the most intense reflections (infrared). Although the analysis process is much simplified as a result, the system cost is increased. We want to employ a low-cost consumer camera that solely records visible light in order to address this issue.

The majority of algorithms described in the literature approach the problem of recognition in 2 stages: detection of license plate for locating the plate and license plate recognition for isolating and recognizing each character in turn.

A plate is normally standardized for orientation/skew, size, and brightness/contrast when it is discovered. Characters are then identified and segmented. The last step is the application of some (national) syntactical rules. Because cars always travel perpendicular to the camera in our application, normalization is not necessary because plate distortion is constrained.

When using a separate plate reader, the license plate must be found before being recognized. A missed plate detection might occur when a portion of the plate is obscured by the camera's positioning. We indirectly detect license plates, as opposed to a separate plate detector, by detecting specific text characters. Since we don't encrypt some license plate components, extra text strings in the image will also be detected. Only the text from registration plates will be read, though, as characters can only be located in the zones of detected cars.

The system is supposed to identify cars adequately in order to enable additional plate recognition. This suggests the employment of two recognition systems, yet one detection system might be reused for both text and automobile recognition due to the embedded system's efficiency and cost constraints. In this essay, we examine a number of significant research questions. Another issue that needs to be resolved is how to identify and recognize characters accurately enough for the system to be used internationally despite regional variations in character typefaces and plate layouts.

**III. PROBLEM STATEMENT**

With the help of technology, people are constantly attempting to make things that will make their daily lives easier.Because it makes everything easier, technological technologies are made to improve the quality of human life . These findings support the advancement of technology, which has led to the creation of several instruments designed to aid human tasks and even take over for people in some roles.

People want simple processes because of the modern technology available, and one such easy thing is to use IoT. (Internet of Things). The Internet of Things (IoT) idea attempts to increase the advantages of permanently connected internet connection.

Even while operating the garage door manually, which is still common. When the home's owner is too sluggish to get out of the car to open or close the garage door, one problem can occur. Using an Android smartphone and a Wi-fi network, this problem can be resolved. We need a control system that can open and close garage doors wherever and whenever is necessary as a consequence. A wireless data transmission channel known as Wi-fi, or "wireless fidelity," can be used to deliver data and programs quickly.

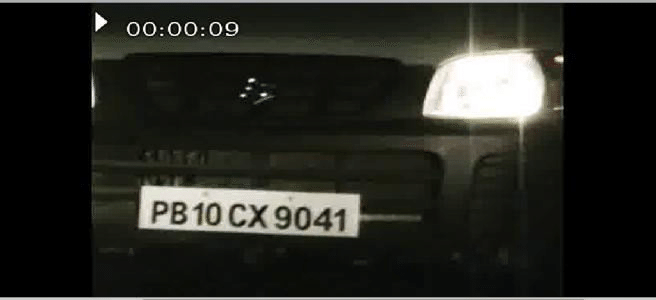
****

Figure 4. Capturing number plate from video

This project aims to develop a control device that unlocks doors with a video camera as an input. The garage door may be opened and closed using Wi-Fi commands from anywhere as long as the Raspberry Pi microcontroller and Wi-Fi signal network are within range. The Raspberry Pi serves as the garage door's CPU and a DC motor serves as its drive, enabling the garage door to open and close.

When the car comes near the garage door the video camera will capture the car's number plate and if the number plate matches with the numbers saved in the database which are the numbers of the registered cars the motor will then open the gate for the car. Wifi will be used to send the data to a database to check whether the car is registered or not and it will send back a response to the Raspberry Pi which will then open or not keep the garage door closed based on the response received over the Wifi.

This system will replace the traditional fob and code system used for opening the garage door.The users are saved from the need to carry a fog or remember the code for the Garage Door. This will also add an extra layer of security as if we use a code to open the door, an intruder can get the code and open the garage door. An intruder can even steal the fob to open the garage door and get inside the house. These situations can be avoided if we use our proposed system and we can further add a system to check for intruders and alert the house owners on their phone regarding the issue. The proposed system has various advantages as it minimizes the need for a human to control it and uses the concepts of Internet of things which connects different electronic devices.The removal of human interference removes the chances of human error and makes the system more robust and secure.

So the proposed system has a lot of advantages over the existing fob and code systems for operating the garage doors.

**IV. PROJECT OBJECTIVE**

The project aims at omitting the dependency on a certain tangible object or any human being for entering a premise. Nowadays, we are highly reliant on RFID keys or humans to do the same which can be replaced with a single system.

This would not only reduce the dependency but will also be cost effective and more secure. The gates can only be opened for vehicles with registered license plates in the system, hence, there is no chance of key theft. Additionally, the manufacturing of RFID or normal keys will be reduced drastically.

Today we also have biometric systems, but the rider has to come out to scan the same out of their vehicle which is another problem this system would solve. Hence, this project can be helpful to ensure security and also help the law enforcement to track down vehicle movements.

The product in an overall sense will increase convenience to the owner and also gives them a huge security to their property. This will save them a lot of time as well which includes entering the password or scanning the RFID tag. This will also remove a lot of dependencies as well like the carrying another key or RFID tag. In case they lose it, they will need to figure out a different way to get in. Whereas, without a system, you will not need anything at all to enter the garage.

**V. METHODOLOGY**

The research and development (R & D) method is used in the design process. The R & D technique is the stage of product design where the product's efficacy is tested. The design stage makes use of a modified version of the Borg and Gall model, which includes the following stages:

A) Information gathering

B) Planning

C) Preliminary product development

D) Preliminary field testing

E) Major product revision

F) Operational field testing

G) Final product revision

Designing the system flowchart, creating the hardware, and integrating the software and hardware are the three steps in the production of control systems.

**VI. SYSTEM DESIGN**

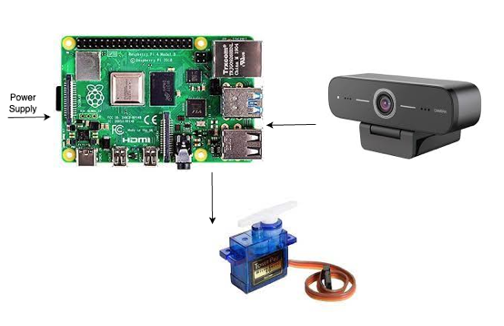


Figure 5. System Architecture

This project includes hardware and software components.

1. *Hardware*

• Raspberry Pi 3 B with Raspbian (Buster)

• Pi camera V2.1

• PIR sensor

• Resistor 220 Ω

• Cover case protection for the Raspberry Pi

1. *Software*

The software is taking frames from Motion project live streams. When the PIR sensor senses an object, it waits for a small delay of up to 1 second and captures a frame. This frame is analyzed with OpenAlpr(Automatic License Plate Recognition Library) to search for license plates. If a license plate (1 or more) is found, the number is inserted into a SQL database and the frame is dumped. If no license is found, the frame is directly dumped.

Components:

• OpenAlpr

• Motion project

• OpenCV

• SQL

• Python 3 scripts

1. *OpenALPR*

OpenALPR is a free and open source library for automatic license plate recognition that has C++, Java, Node.js, Go, and Python bindings. To locate license plates, the library examines photos and video sources. The result is a text representation of any characters from a license plate.

1. *OpenCV*

OpenCV is a real-time optimized Computer Vision library, equipment, and techniques are provided by OpenCV. Additionally, it facilitates the execution of machine learning (ML) and artificial intelligence (AI) models (AI).

1. *Tesseract*

Optical character recognition (OCR) platform Tesseract is open source. OCR translates text from photos and documents without a text layer into new searchable text files, PDFs, or the majority of other widely used formats after extracting the text from them. Tesseract is extremely adaptable and works with most languages in addition to supporting vertical text and multilingual publications. Even though Windows and Linux users may use the program, this lesson will focus on Mac users. This requires the use of the terminal application.

1. *SQLite*

SQLite is a compact, effective, and simple to use SQL database. It performs effectively in applications such as mobile devices, embedded systems, autos, and others where there may not be much CPU power available. It is excellent for Raspberry Pi because of this.

The Raspberry Pi works well with SQLite for:

i) An expedient method for automatically storing data in table format.

ii) while your Raspberry Pi is serving other devices as a server.

iii) Studying relational databases and SQL

A passive infrared sensor (PIR) is an electrical device that detects the infrared (IR) light generated by objects in its range of view (PIR sensor). They are used most frequently in PIR-based motion detectors. PIR sensors are often used in systems for self-contained illumination and security alarms.

PIR sensors can generally detect movement but cannot tell what or who moved. For such, an imaging IR sensor is required.

The abbreviation "PIR" or, less frequently, "PID," which stands for "passive infrared detector," is widely used to refer to PIR sensors. Because PIR devices don't release energy to detect objects, they are regarded as passive devices. The detection of infrared radiation (radiant heat) that is emitted or reflected from objects is their sole means of operation.

The Pi Camera module allows for the capture of high definition video and pictures. We can directly connect the PiCamera module to the Raspberry Pi Board's CSI (Camera Serial Interface) interface. This Pi Camera module can be attached to the Raspberry Pi's CSI port via a 15-pin ribbon cable.

**VII. HARDWARE SPECIFICATIONS**

The project uses Raspberry Pi 3B with Rasbian(Buster) with 4GB Ram, Pi Camera V2.1, PIR Sensor and a servo motor 130 RPM 6V Micro.

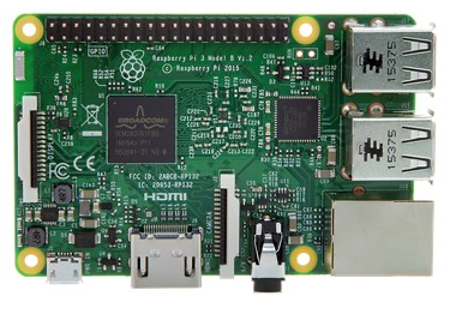


Figure 6. Raspberry Pi 3 B with Raspbian (Buster)

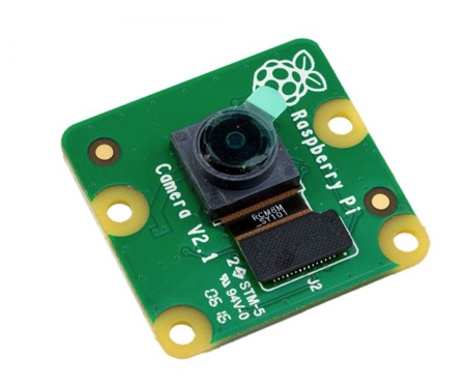


Figure 7. Pi camera V2.1

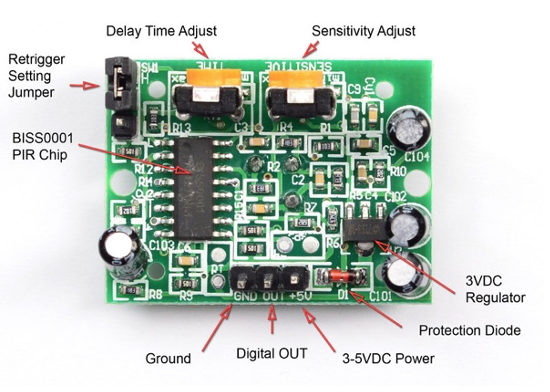


Figure 8. PIR sensor



Figure 9. Resistor 220 Ω



Figure 10. Servo Motor

**VIII. PROJECT ARCHITECTURE**

In this project, we will be using the main 3 functional components:

1. Raspberry Pi
2. A Camera
3. Servo motor

Raspberry Pi is a computational unit/CPU of the system with peripherals like a camera and a servo motor which acts like a gate.

The video feed from the camera is continuously monitored by the raspberry pi. When a vehicle or a number plate is detected, the number plate number is extracted by the algorithm. Using this number plate, a lookup is done on the databases which are on the Raspberry Pi itself. If there is a match for the number plate in the database vs the number of the plate recognised, the gate will be opened. Otherwise, the gate will remain shut. The servo only serves as a gate for demonstration.

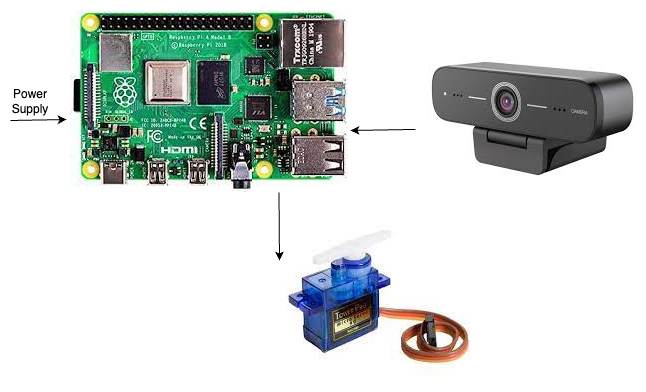
****

Figure 11. Architecture

We chose Raspberry Pi because it’s computational powers are close to a regular computer. Since we also need to run applications like OpenCV and tesseract to process the captured image, Raspberry Pi seemed like a promising platform to build on.

The camera which we are reusing is the Logitech C270 which is an HD camera with 60 degree angle view. The main reason we chose this camera is because it’s video and image qualities are really good and is cheap as well. It is also very easy to integrate with a Raspberry Pi.

**IX. SETUP**

1. *Tesseract Setup:*

The following steps were followed to install Tesseract.

*$ cd /usr/src*

*$ sudo git clone*

*$ https://github.com/tesseract-ocr/tesseract.git*

*$ cd tesseract*

*$ sudo git tag*

*$ sudo git checkout 3.04.01*

*$ sudo ./autogen.sh*

*$ sudo ./configure --enable-debug*

*$ sudo make -j2*

*$ sudo make install*

*$ sudo ldconfig*

*$ tesseract -v*

1. *OpenCV Setup:*

The following steps were followed to install OpenCV:

*$ cd /usr/src*

*$ sudo wget $https://github.com/opencv/opencv/archive/4.2.0.zip*

*$ sudo mv 4.2.0.zip OpenCV-4.2.0.zip*

*$ sudo unzip -q OpenCV-4.2.0.zip*

*$ cd opencv-4.2.0*

*$ sudo mkdir release*

*$ cd release*

*$ sudo cmake -D* CMAKE\_BUILD\_TYPE=RELEASE -D CMAKE\_INSTALL\_PREFIX=/usr/local -D $ ENABLE\_PRECOMPILED\_HEADERS=OFF ..

*$ sudo make -j2*

*$ sudo make install*

1. *OpenALPR Setup:*

*$ cd /usr/src*

*$ sudo git clone https://github.com/openalpr/openalpr.git*

*$ cd openalpr/src*

*$ sudo mkdir build*

*$ cd build*

*$ sudo cmake -D* CMAKE\_INSTALL\_PREFIX:PATH=/usr -D CMAKE\_INSTALL\_SYSCONFDIR:PATH=/etc ..

*$ sudo make -j2*

*$ sudo make install*

*$ sudo ldconfig*

To make sure the script is always running, the following commands were followed for the setup:

We added our command to the crontab to make sure it is always running:

*$ sudo crontab -e*

*$ 0 \* \* \* \* ./path\_to\_our\_cleaning\_script.sh*

We also added our command to rc.local so that on start-up or restart of the system, the script can start running automatically.

*$ sudo nano /etc/rc.local*

*$ python3 /home/pi/myscript.py &*

**X. IMPLEMENTATION**

1. *USE CASE diagram*

The use case diagram shows how people and the system interact with one another. The following figure shows the use case diagram for this project.

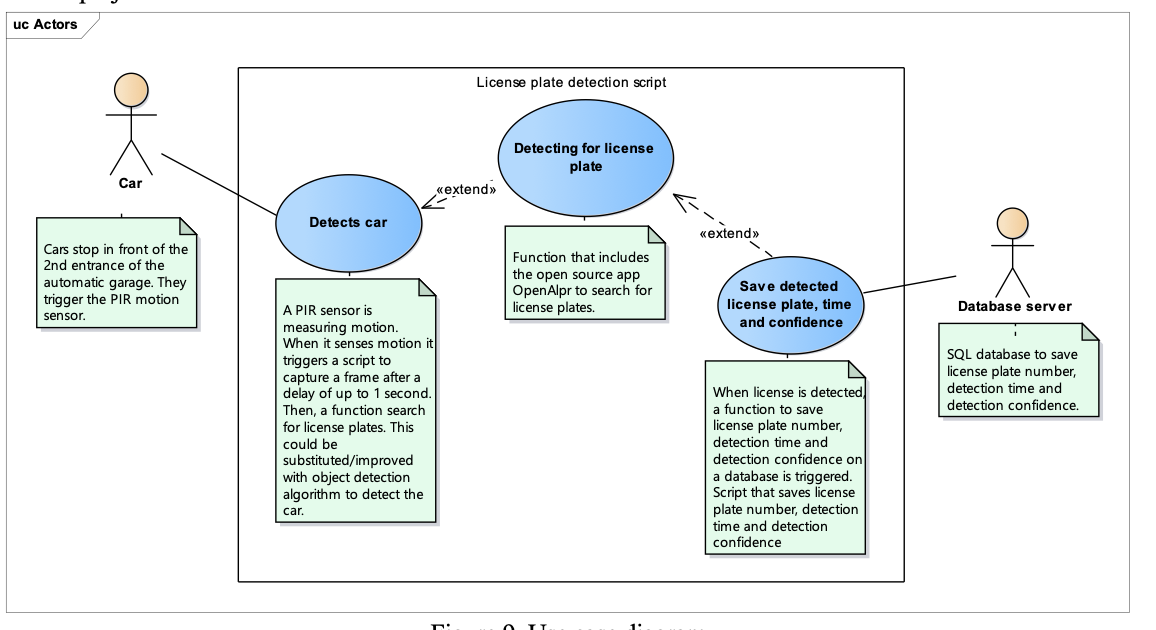


Figure 12. Use case diagram

1. *User roles*

The users are the car and the database.

1. Car: The car will stop in front of the 2nd entrance of the automatic garage. The car will trigger the PIR motion sensor.

2. Database: The SQL database receives/saves the license plate number, detection time, and detection confidence, from the detection function if a license plate is detected.

1. *Activity diagram*

A behavior diagram type called an activity diagram displays the actions taking place during a process. The techniques, partnerships, activities, and state histories of the objects in a system are all displayed in behavior diagrams together with their dynamic behavior. The development of modifications made to a system through time can be defined as the dynamic behavior of the system.

The activities occur for detecting movement when the car is approaching the garage inside the door.

This can be improved by using machine learning car detection models like MobileSSD or tiny YOLO.

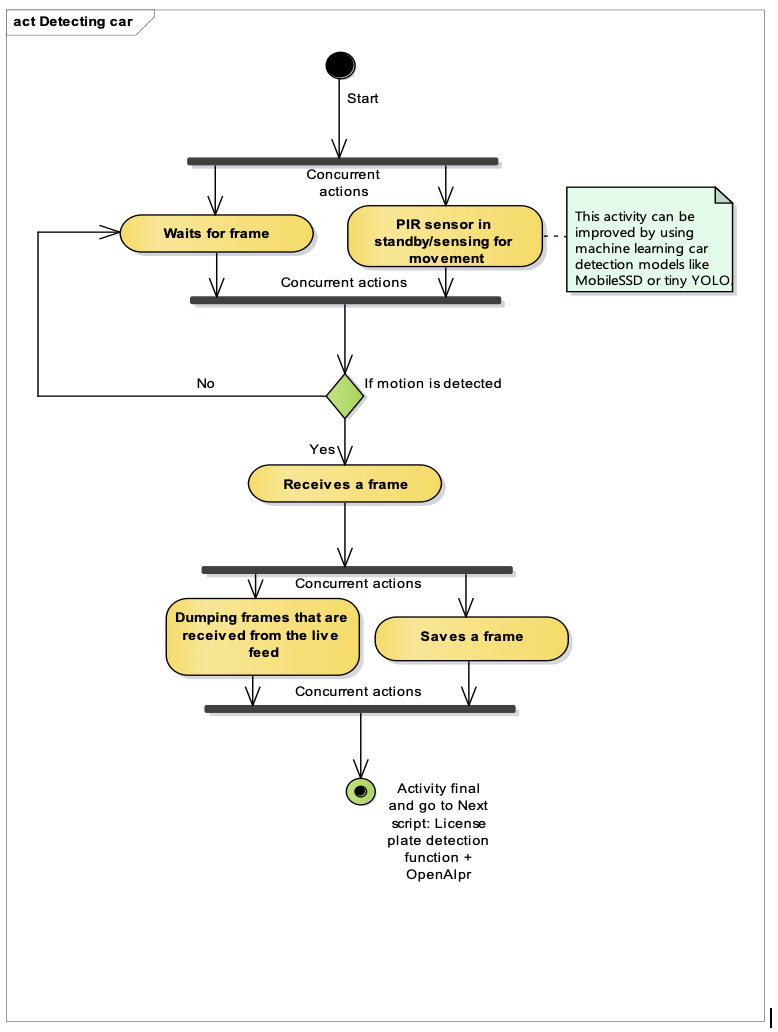


Figure 13. Activity diagram: detecting car

The activities occurring after movement is detected.

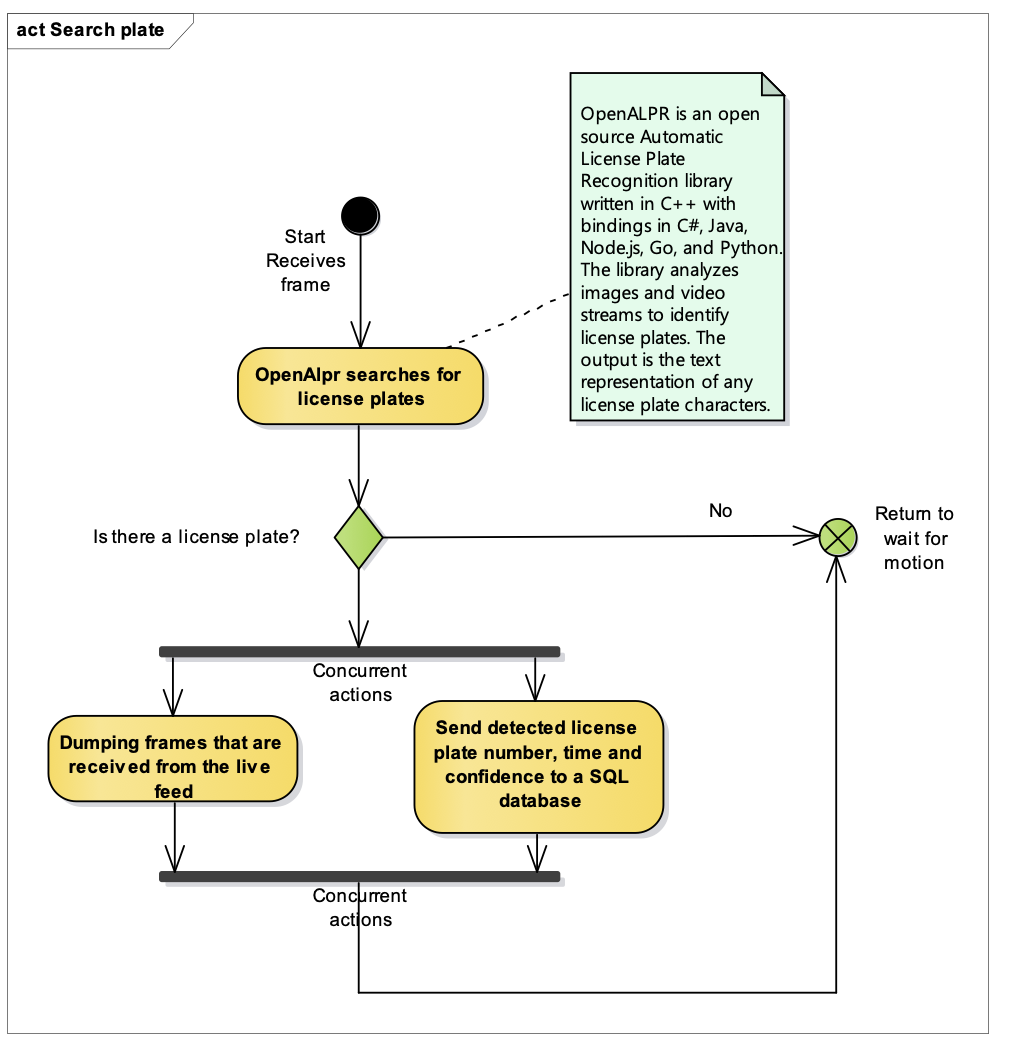


Figure 14. Activity diagram: detecting license plate

1. *Timing diagram*

“Timing diagrams focus on conditions changing within and among Lifelines along a linear time axis. Timing diagrams describe the behavior of both individual classifiers and interactions of classifiers, focusing attention on the time of occurrence of events causing changes in the modeled conditions of the Lifelines”. The basic objective is to show the linear evolution of a lifeline, which symbolizes a Classifier Instance or Classifier Role. Demonstrating how an object's state evolves over time in response to known events or stimuli is the most frequent application. 2017 OMG Object Management Group.

The time that the detection takes is 9.5 seconds.

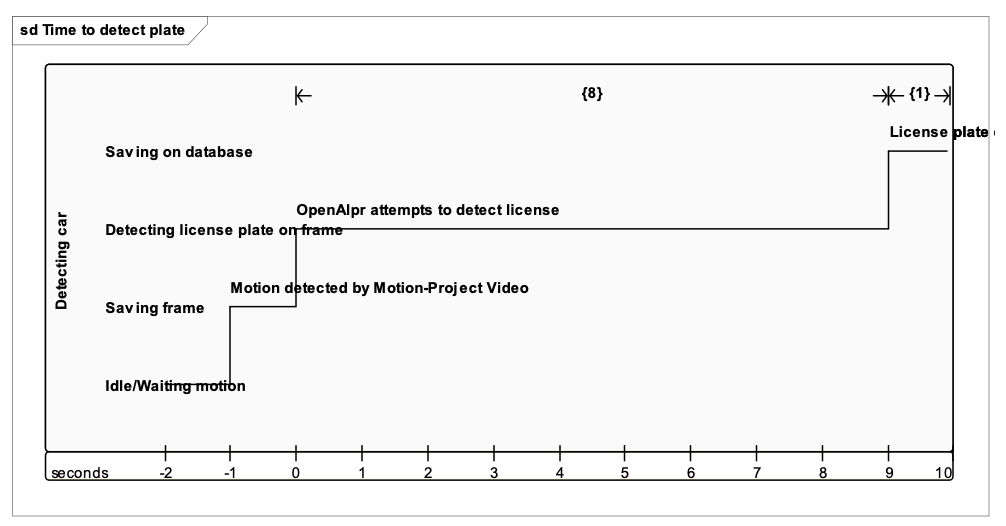


Figure 15. Timing diagram

1. *Flow of data*

i) Images are taken from a live video feed.

ii) The images are analyzed using the license plate detection function.

iii) Eventually OpenAlpr detects if there is a license plate in the image, if any, it's inserted into SQL.

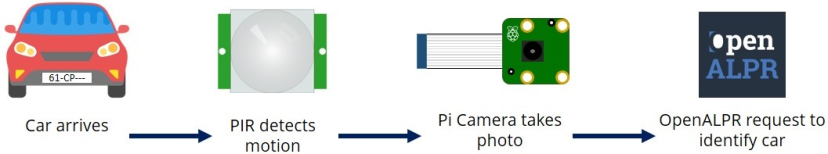


Figure 16. Flow Diagram

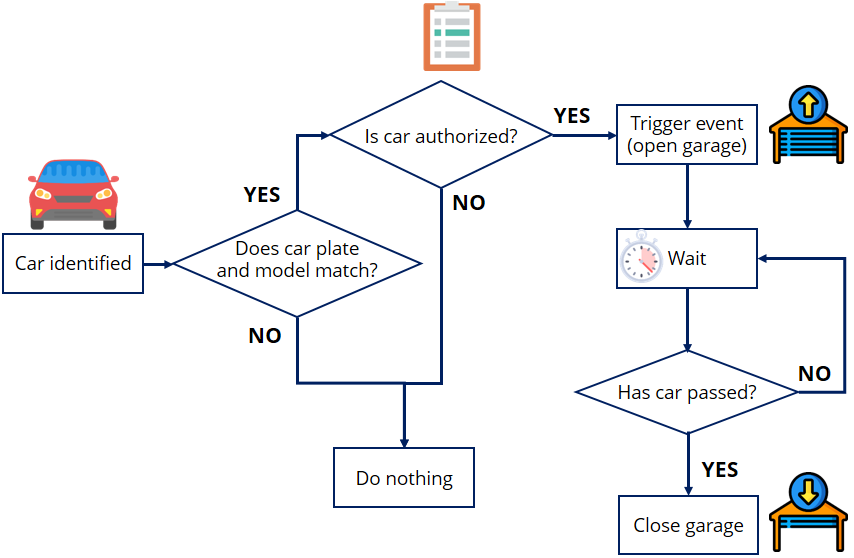


Figure 17. Decision flow

**XI. CAR RECOGNITION**

The Histogram of Oriented Gradients (HOG) technique is used to train the automobile detector. There are 18 orientation bins using the sign, 88 pixel cells, 11 block normalizations, L2 feature normalization, and a 12864 pixel detector size employed. The resulting feature vector has a dimensionality of 2,304.

Our method is trained using the open multi-view car dataset from Kuo and Nevatia. The 2,462 positive car samples are mirrored to produce 4,924 samples. To obtain negatives from the 8,427 non-car PASCAL 2007 samples, we employ bootstrapping. Over the course of 10 epochs, a linear detector is trained using stochastic gradient descent (SGD) with a threshold of =0.001 (a different dataset's training process is detailed in more detail in [XII]).

**XII. LICENSE PLATE RECOGNITION(LPR)**

Characters are retrieved from an image when an automobile is found there.

This is accomplished by sampling the image in increments of 16 pixels on the horizontal axis and 8 pixels on the vertical axis using a sliding detection window with a size of 3232 pixels. The vertical step size is reduced to avoid binarization effects at the boundaries of the license plate. The image is scaled down in 1.5 step intervals to make it easier to distinguish the larger characters.

For recognition, we once more employ the following configuration of the Histogram of Oriented Gradients (HOG) descriptor. We use a description/detection window of 2424 pixels (n=24), 44 pixel cells, 22 cell blocks, and 12 orientations, including sign and L2 normalization. Because of the low resolution of the photographs, smoothing the image during gradient calculation is preferred.As a result, we employed a 33 Sobel filter rather than the 13 filter that was used for the automobiles. SGD is used to train the linear classifier with 20 epochs and =1e-4. Using the 100 positive training samples we acquired from the font files, we trained each classifier. We randomly selected 100 samples from each of 1,000 character-free photos to create the negative samples, yielding a total of 100,000 negative samples. The characters on the plates have the following recognition ratings. For 80% of all plates and photos, we have successfully detected and recognized every character on the plate.

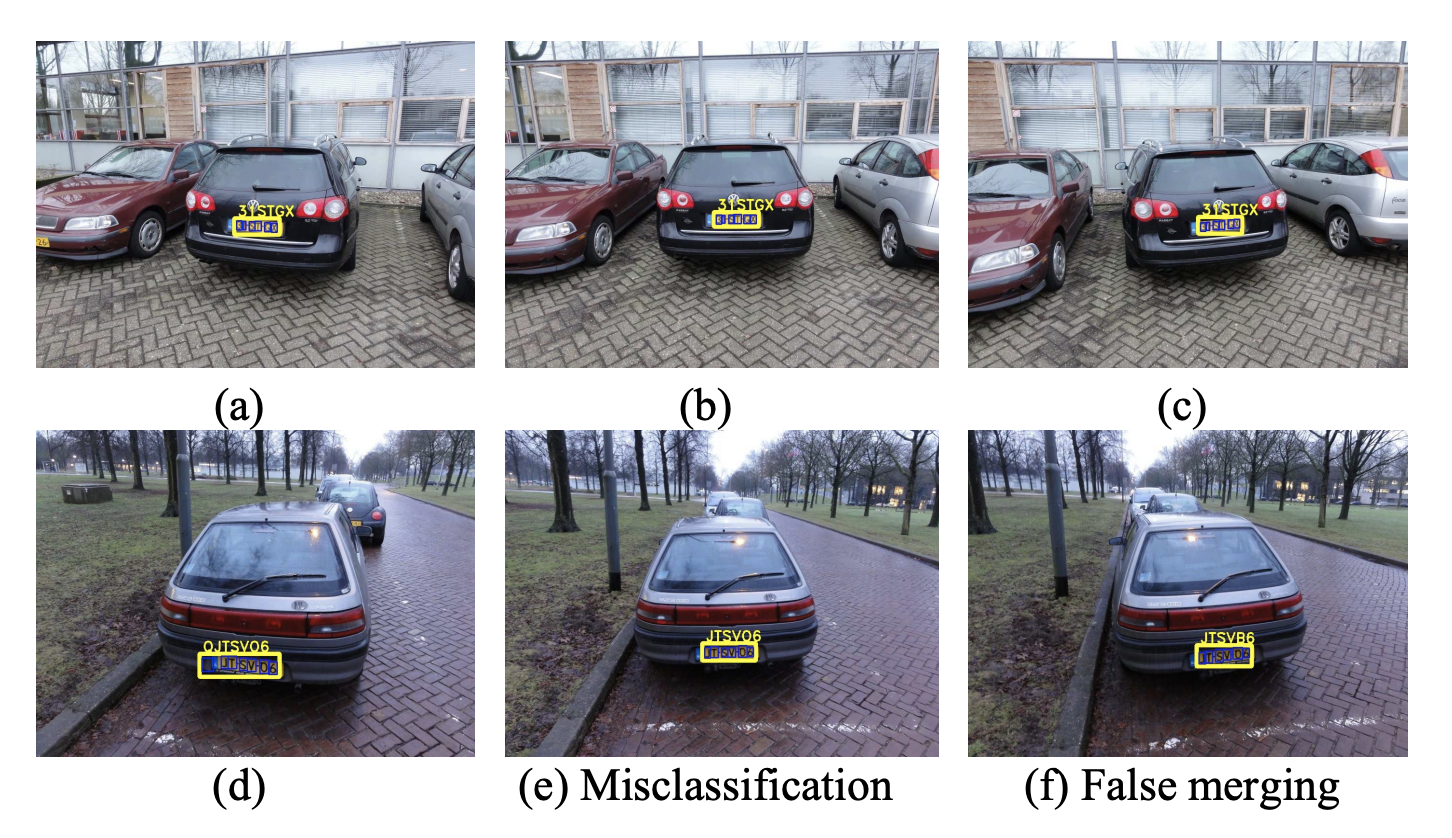


Figure 18. Example character recognition on the LPR dataset. Detections in (a-c,e) are correct, (d) misclassified character, (f) false merge.

**XIII. FALSE DECISIONS FOR AUTHENTICATIONS**

Analyzing the quantity of incorrect judgments is an essential component of any verification system. False positives are vehicles that are wrongly authenticated, whereas false negatives are vehicles that have access but are not authenticated by the system. False positives in the context of the application under consideration prevent the user from entering his garage while false negatives prevent the user from entering his own garage. False positives in car detection are not a problem because the license plate recognition system still needs to identify the plate. The user's identity won't be validated if no plate is discovered. All cars are detected, however false negatives (missed automobile detections) restrict authentication.

Character recognition false positives (extra characters identified) only cause false authentication when they are discovered between the characters on the real plate, which only happens on two plates. Verification mistakes are caused by incorrectly categorized characters, and they occur in 14 plates. In 25 plates, characters are missing, which also results in inaccurate verification. 90% of the license plates in aggregate were authenticated successfully.

Only single photos were used to acquire these performance metrics. Using additional frames for the analysis might be advantageous for the real application. False visual authentication is also not a concern if the computer vision system works in conjunction with the radio-based system as an additional verification step since a radio signal is also required.

**XIV. FUTURE SCOPE**

The project can be extended to make a mobile application that can help monitor the total number of vehicles attempting to enter and vehicles which successfully made it in. It can be used as a management system for vehicles driving in or out of a gated area. Door activation from a mobile application and over the internet so that the owners can remotely allow guests to enter the garage when they are away or even for some scheduled deliveries.

A remote feature to lock the gates for a certain period of time is decided by the owner so that the door never opens. This feature can be turned ON and OFF by the owner itself. This feature will be handy in some scenarios such as a long vacation or in times when the owner is away from the house. Automatic courtesy lights in the driveway so that everything is clearly visible. This light will switch OFF by itself after the garage door completely closes.

For guests to have access to the garage, owners ought to be allowed to set up a guest account. The guest account should automatically be terminated after a certain period of time which the owner decides. This will prevent the owner from coming onto the mobile application again and again when a guest is staying at the place for an extended period of time.

Cloud integrations support all remote operations that an owner chooses to do. The owner can remotely view the video of the garage and the driveway at any time he wants and should be able to make decisions remotely.

The product should support any kind of add-on accessories that is available in the market. The owner should be able to buy any add-on accessories from the market and should be able to seamlessly integrate with the existing products and systems. This integration can be of any type such as courtesy lights or a keypad-controlled garage door for visitors for which the PIN can be remotely changed whenever the owner wants to change.

Make the product highly secure with the help of state-of-the-art technologies that are available on the market or on open source to make the product very secure. The product should be within a private network away from potential hackers. A lot of testing should be done to ensure that the private network is not penetrable.

The product should be always online and should be fault tolerant. No matter what happens, the product should fall back to a safe state. The mobile application should be able to reset to the initial version in case the system feels like the security has been compromised and should notify the owner and also should give him a report on why this happened. The mobile application along with the product and the systems in place should give the user a very smooth user experience. The app or the hardware systems should never get stuck. The user should not feel the application doesn’t respond to the actions that happen on the mobile application. The entire system along with hardware and software should have a very smooth and real-time experience for the users.

Improve the Image Processing algorithm to be able to detect any type of number plates. In case multiple vehicles are seen in the image, the algorithm should be able to detect the car in the front. Even if the number plate is very small, the image processing algorithm should be able to detect the number plate. There should never be a time when the system is not able to detect or process the number plate. The algorithm should recognize the correct number plate number at any cost. This is the primary feature of the product and hence it should never fail.

An alert or notification should be sent to the owner when an unauthorized vehicle or person is trying to enter the garage. The owner should be able to make decisions on his mobile application to let the vehicle in or not. The owner should also have the feature inbuilt into the mobile application to notify the police authorities or call 911 in case of emergencies like this. This will increase the security onsite.

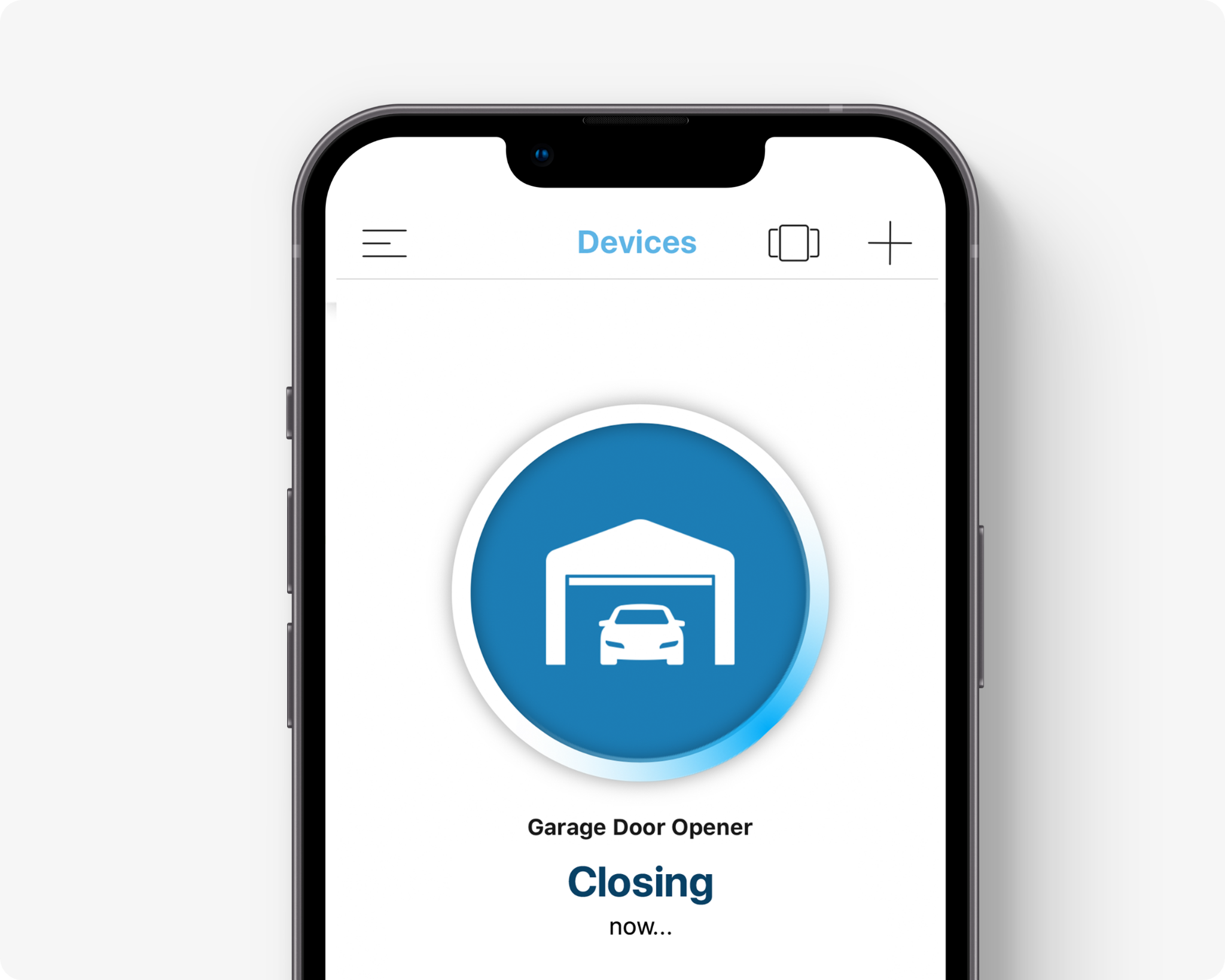


Figure 19. Probable Mobile Application

The owner should be able to run diagnostics on the system whenever he wants and the product itself should run diagnostics and give a report to the owner periodically. This will help the owner to have trust in the system if it’s able to detect failures and notify the owner.

**XV. CONCLUSION**

This project is about an IOT device that helps to open and close the garage door when a vehicle approaches the gate the video camera reads the number plate of the vehicle and if the number plate matches the registered number plate gate is opened by the motor. The smart gate can replace the fob key used to open the gates and there will be no need to carry a key or remember a code to open the gate. The sensors used in these devices are quite simple and are available at low cost which will make them affordable for everyone in the consumer market.

We have implemented a Proof of Concept in this project which can be extended to create a full fledged product from the project which will be very useful in many settings such as apartment complex, universities with reserved parking, or individual houses/places where there is a garage door and saves so much time and hassle to the users.

In this project, we successfully showed the use of image analysis to open a garage door and created a system around it. We will be making the project an open-source project via GitHub upon completion of the project.

**XVI. REFERENCES**

1. Kiana Tehrani A. Michael "Wearable Technology and Wearable Devices: Everything You Need to Know" Wearable Devices Magazine 2014.
2. Ramdhani M A, Aulawi H, Ikhwana A and Mauluddin Y 2017 Model of green technology adaptation in small and medium-sized tannery industry Journal of Engineering and Applied Sciences vol 12pp 954–62.
3. Setiawan S 2014 Miniatur Pengendali Pintu Gerbang Menggunakan Smartphone Via Bluetooth Berbasis Arduino Uno (Universitas Mercu Buana).
4. Hidayatullah N A and Juliando D E 2017 Desain dan Aplikasi Internet of Thing (IoT) untuk Smart Grid Power Sistem VOLT J. Ilm. Pendidik. Tek. Elektro 2 35–44
5. Smart Vision Labs, Smart Vision Labs’ List of 5 Best Smart Glasses.https://[www.smartvisionlabs.com/blog](http://www.smartvisionlabs.com/blog)
6. IoT Based Automatic Garage Door Opening System Prototype on: <https://www.hackster.io/ahmadsatria/iot-based-automatic-garage-door-opening-system-prototype-20a67f>
7. W. Uriawan et al, "Internet of things for automatic garage doors using ESP8266 module," IOP Conference Series.Materials Science and Engineering; IOP Conf.Ser.: Mater.Sci.Eng, vol. 434, (1), pp. 12057, 2018. . DOI: 10.1088/1757-899X/434/1/012057.
8. Wijnhoven, & de With, P. H. N. (2011). Identity verification using computer vision for automatic garage door opening. IEEE Transactions on Consumer Electronics, 57(2), 906–914. https://doi.org/10.1109/TCE.2011.5955239