Face Identification Lock

ECE 445 Final Paper

Kaiji Lu, Zan Chen, Zekun Hu

Group 69

TA: Jacob Bryan

May. 1st, 2018

1 Introduction

1.1 Objective

It is quite annoying when we are at the front door of our houses with hands full of things from the supermarket, but we can not enter in since we still have to take out the key from our packet to unlock the door. Just like we have different approaches to unlock our phone other than entering pins, physical keys can be more and more replaceable now. A face-identification lock is one solution to the problem of finding keys and losing keys. Instead of putting things on the ground and then use our free hand to get the key and unlock the door, we can easily stand in front of a camera and then the door will automatically unlock after it identify us.

The trend of facial recognition systems have never been higher these days, especially after the release of Apple's newest product: iPhone X. The designers discarded the home button that the company has been using on every single iPhone before iPhone X. To replace the fingerprint sensors on the home buttons they used on previous iPhones, they introduced face ID: a biometric authentication technology that scans the user's face to achieve identification

The project we propose seizes the achievement of turning a cheap camera into an intelligent guard. Nowadays, face identification has been an active area with many deep learning methods. Specifically, a neural network approach is good to tackle the face detection while maintain a relatively high accuracy. According to a research conducted by Stanford University, their neural network algorithm is able to produce a 0.0884% false positive rate and overall 91% accrаucy [1]. Even though the system does not necessarily seem more secure than a regular lock, it does provide us a much more convenient way to open the door when our hands are full while the tradeoff of security is in an acceptable level.

1.2 Background

At present, people are paying increasing attentions to home securities. The market of home security products will possibly reach 12 billion USD by 2025 [2]. In particular, the image processing technology has made more and more security solutions possible such as smart monitor systems or motion triggered cameras. As an alternative to traditional security systems that use passwords, cards, or keys, face recognition lock is one of the image processing technology that both offers users great convenience of hands free operation and ensures the sensitive area to be monitored and controlled[3]. However, we have to agree that there is no exact formula or rule to define one person's face. Thus we can not directly tell a computer to recognize a human being's face. Instead, we choose machine learning to approach such problem. We let the computer to learn from the experience by feeding with input and hand-designed features or multiple layers of features. By mapping from those features, the computer is able to provide us the output. Particularly, in our algorithm, we will use a fully-connected layer and a dropout layer.

Our project only requires a simple USB camera, so it is cost effective and reliable compared to the average price of 400 USD of existing products in the market.

1.3 High-level Requirements List

- The false negative rate of our face identification system must be below 1%, which implies intruders have little chance to lock the door. Besides, overall accuracy rate must be above 85%, which means the lock works as intended at more than 85% of the time.
- The lock will respond to the user within 5 seconds. That is, either the LED shows the face identification fails or the door unlocks.
- The microphone should be able to detect an individual sound “hey” and activated face identification system in the correct situati

2 Design

Our design is make up of four components: a power unit, a recognition unit, a actual lock system and a printed circuit board (PCB). A power unit ensures the power supply to all of the other components so that the door lock system can work continuously all day. A recognition unit is the core of computing our convolutional neural network algorithm and feeding back our face identification outcome to the control unit in PCB. The PCB contains multiple chips and helps connect the other components. It detects sound and receives image signal if activated. Besides, it process the image into data we need for the computing part. It also contains a microprocessor to receive signals from the recognition unit and send to the lock system unit. The lock system unit contains a motor driver which helps unlocking the door if it receives a signal from the PCB.

In our design, the camera, will firstly take multiple pictures of the users and then transmit them to the recognition unit. The recognition unit has a pre-processed training model stored in a SD card which uses a convolutional neural network algorithm. Our camera data will go into the recognition unit as test inputs and the unit generates a true/false signal back to the control unit indicating whether the face identification success or failure. Here is a flow chart of the design, please refer to the state diagram to see more about how decision is made in the last step.

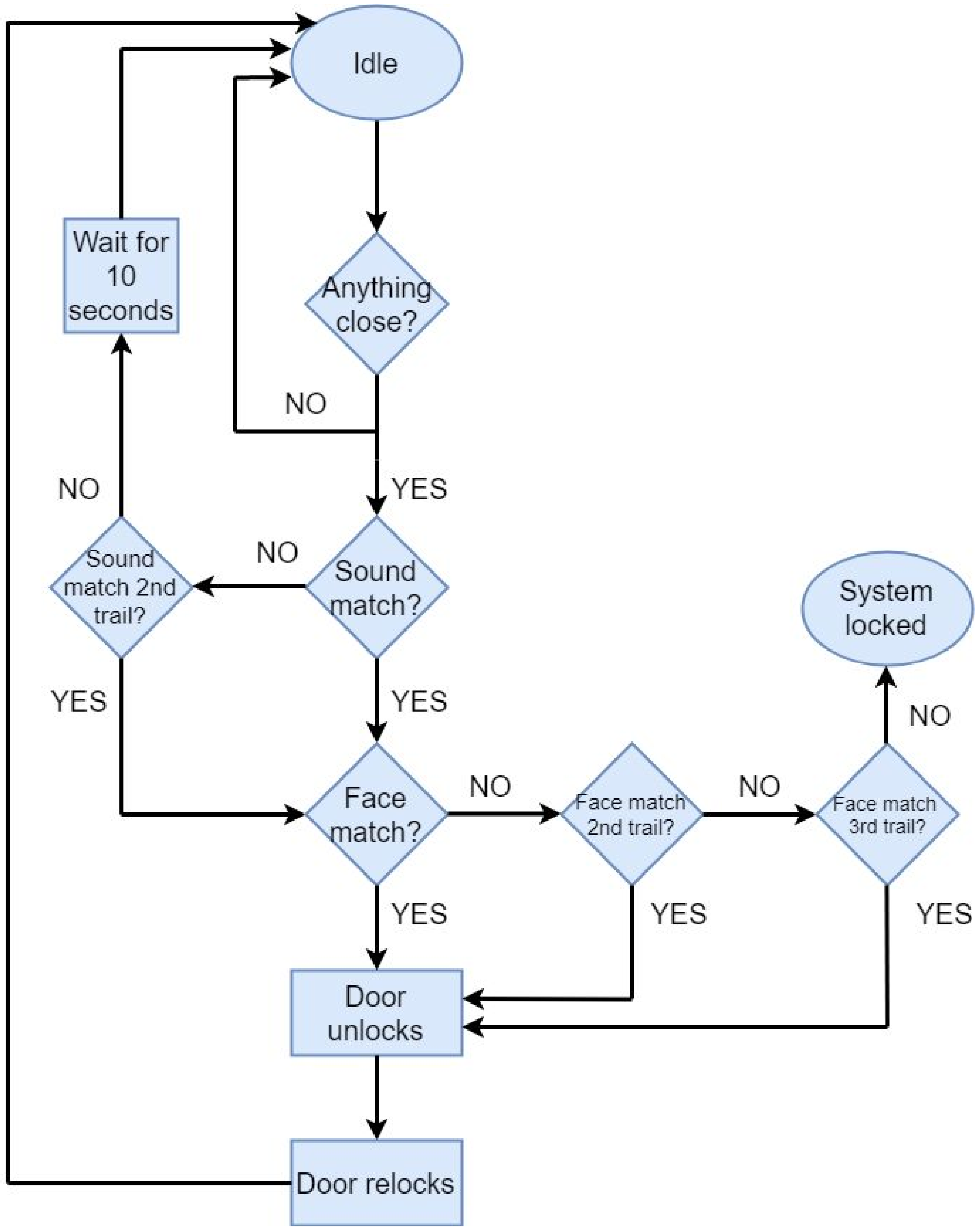


Figure 1. Flow Chart

Block Diagram:

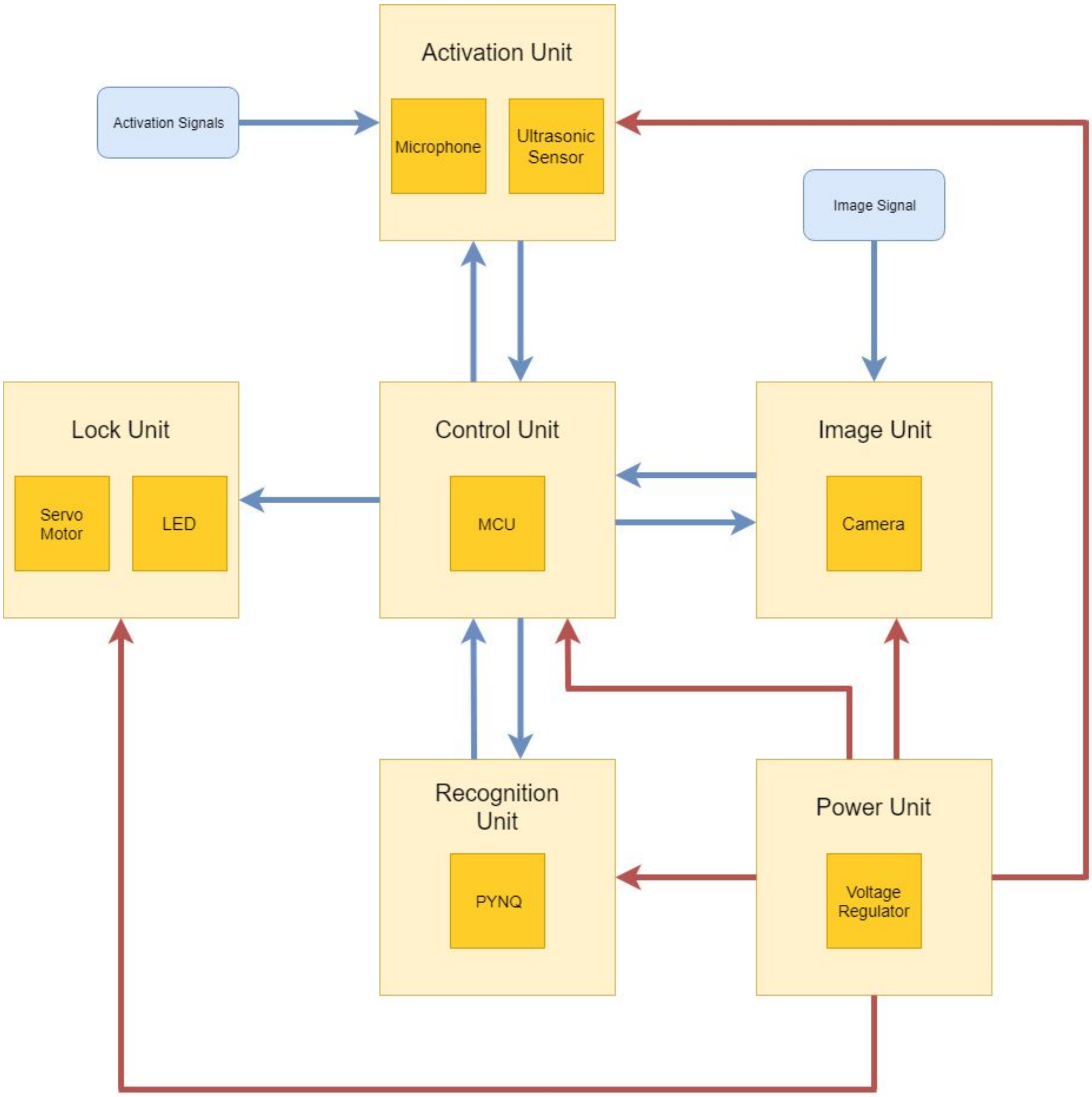


Figure 2. Block Diagram of the System

2.1 PCB

2.1.1 Proximity sensor

The ultrasonic sensor helps to detect whether there is anything in front of the door. We use it to provide unnecessary power consumptions of the camera and the microphone.
It will emit an 40,000Hz ultrasound and bounce back hits an object. On Arduino, it is able to output the time duration of the traveling sound wave. Then we can use the below formula to calculate the distance between our sensor and detected object:

$$distance = \frac{speed\ of\ sound\ \times time\ taken}{2}$$

An alternative for ultrasonic sensor would be a infrared sensor. However, unlike ultrasonic sensors, infrared sensors do not work well in dark environment. Also, ultrasonic sensors are completely insensitive to light, dust, smoke, vapor, lint, etc..

2.1.2 Microphone

The microphone is activated only after the ultrasonic sensor has detected an object with 30cm. If it detects a short individual sound (i.e "hey") trained by the user , it will outputs a signal to activate the camera. In this way, we help to avoid a problem which some noises accidentally activate our face identification system and "double-safe" our system by adding a sound detection before the face detection.

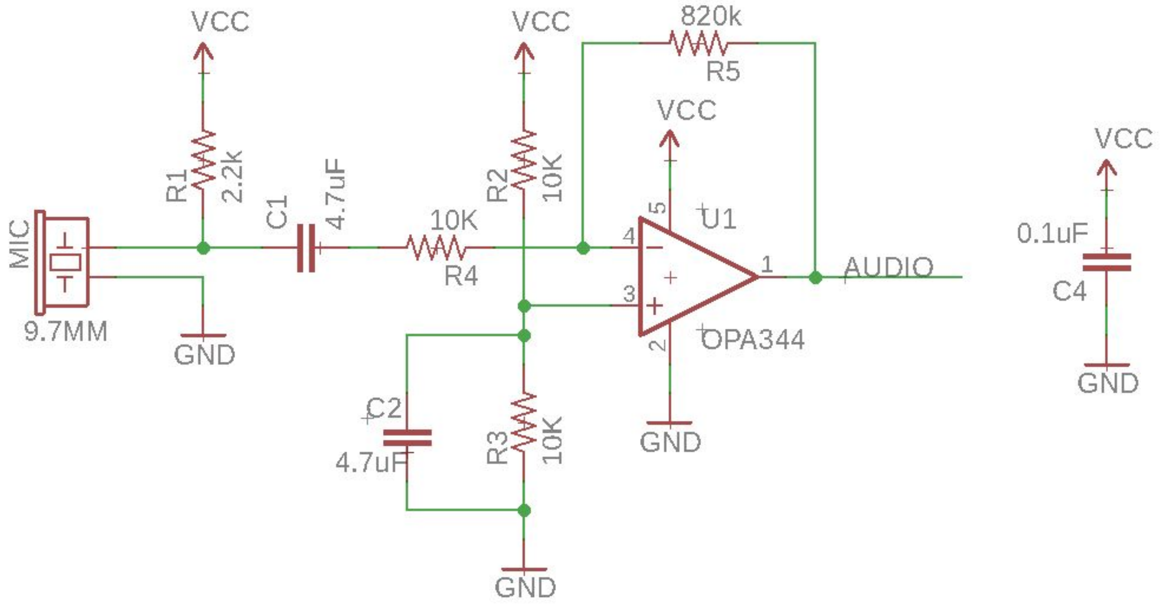


Figure 3. Microphone Circuit Schematic

The designed circuit above has a voltage gain of 82V/V, which is approximately 80dB. And we place a decoupling capacitor near to OPA344 to prevent some circuit interference. The output AUDIO has a voltage of half the supply voltage, which is ready to input to the ADC of our microcontroller.

2.1.3 Microprocessor

We plan to use ATMega328P microprocessor that controls the operation of the whole system: activation of image sampling, signal of computing algorithm on face identification, and the control of motor driver. The microprocessor has processing power of 20 MIPS at 20MHz.

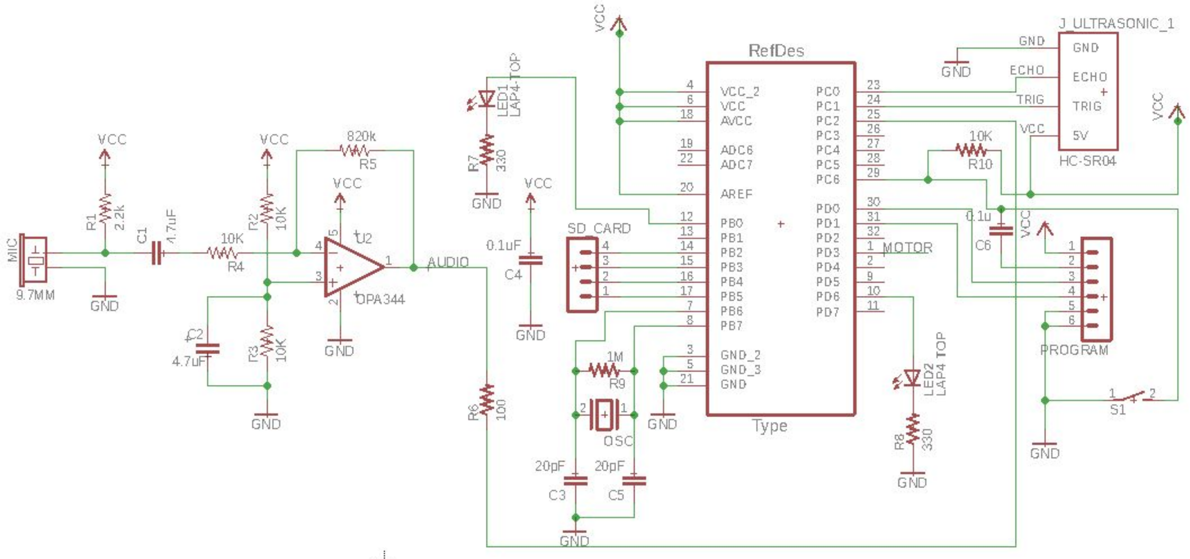


Figure 4. Microcontroller Circuit Schematic

Figure 4 above shows the input and output from or to other modules of our circuit. We will need a crystal oscillator built inside as a generated clock signal to synchronize the circuit. We will first build this design with our dev board, and then program it using kit in the lab.

On software side (Arduino) **,** it needs to regulate all the signals and give instructions to lock system accordingly (See Appendix A) Besides, it will use Fast Fourier Transform (FFT) to extract sound features and check if the sound is matched (i.e “hey”). By using FFT, we are able to tell what frequencies are dominant in a sound and therefore we can compare two FFTs to see whether microphone detects a wanted sound.

In details, we use 128 samples as our sample rate and 2000 Hz as our sample frequency. The reason why we only use 128 samples is because that the sample number must be a power of 2 and Arduino Uno does not have enough memory for storing 512 doubles (if use 256 samples, we need array for 256 real components and 256 imaginary components) We have Zan speak “hey” for 50 times, computing FFTs each time and store the average of the result into Arduino. After we collected the data through training, we can now get FFT of the input audio if it hits a certain threshold and compare it with the trained data. If it has above 70% similarity we classify the sound as “hey” from Zan.

2.1.4 Power Regulation Unit

We plan to use LM317 voltage regulators to manage out voltage supply for different components on PCB as they have different voltage input requirements. Our microcontroller and ultrasonic sensor need 5 volts, other components (microphone, LEDs, servo motor) all require 3.3 volts.

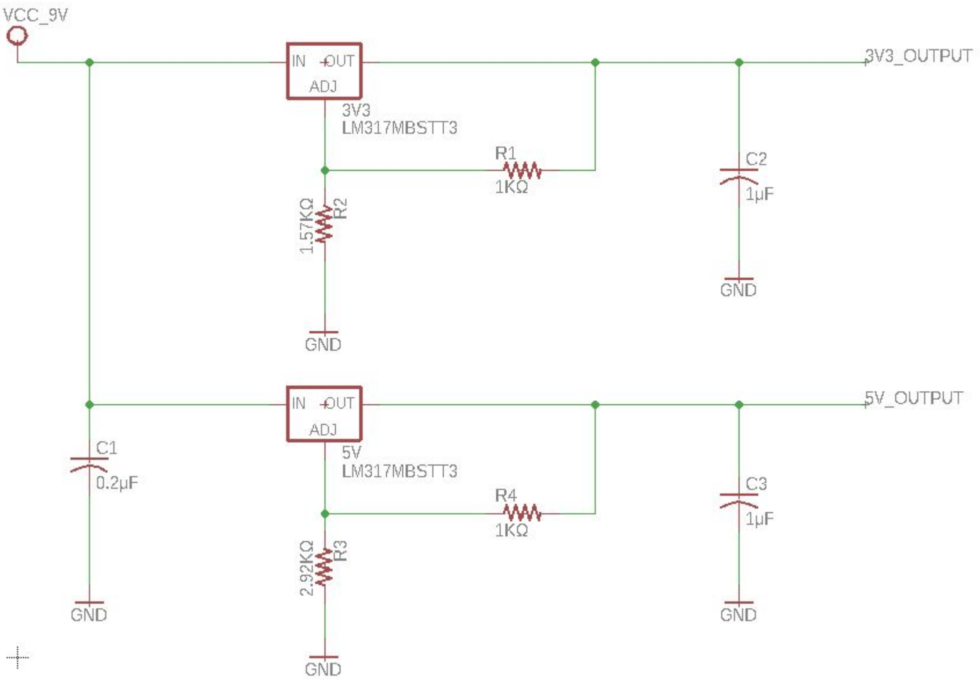


Figure 5. Power Regulator Unit Circuit Schematic

2.2 Lock System

2.2.1 Servo Motor

The servo module serves as a “mini lock” in our project. It will start rotating after face detection successes and after three seconds delay it will rotate back.

2.2.2 LED

The LED takes signal input from MCU, indicating all kinds of results during detections. Details also shown in Appendix A.

2.3 Power Unit

Power unit supplies DC power to all the other units. It is connected to a AC wall plug, with a AC to DC converter. We also need to design a regulator that stabilizes the circuit.

All of the input voltage to the chips and units should be strictly controlled within limit range. The power regulation unit we designed needs a range from 11.5~12.5V input converted from AC wall power.

2.4 Recognition Unit

Face identification runs on a PYNQ FPGA board. Its inputs are the raw images from the camera. The input test images should be tested by the pre-trained classifier, and the recognition unit should be able to output identification result that is either true or false. The output signal will go back to the control unit.

2.4.1 PYNQ

PYNQ board is based on ZYNQ chip. We choose this board because its computation capability with Cortex ARM A9 processor and its python enabled environment.

2.4.2 Camera

We used Logitech C920 USB camera to take images. The camera is connected directly to and powered by the PYNQ board. It is initialized to output RGB images with size 640 * 480 pixels. We control the camera to take one frame at a time until there is a good face detected by our face detection algorithm.

2.4.3 SD card

SD card can be used as memory once formatted. It should be big enough to store about hundreds of images as trained data.

2.5 Tolerance Analysis

Since our project is a security device, lowering the error rate is extremely essential. There are two types of error: a false negative and a false positive. A false negative will lock the user from entering the door, but could be simply solved by a key. Just like the face id system of iPhone 10: a negative result of recognition would prompt the password option. A false positive, however, is way more dangerous. It would allow intruders to use this glitch to unlock the door.

With our current algorithm and design, the total error rate is less than 15%. False negative rate is estimated around 10%, therefore, false positive rate is less than 5%. Our USB camera will take 60 pictures in one second. We will first rule out bad inputs like blurred pictures. Then we will randomly pick 10 out of the good inputs to test. If 9 out of 10 good inputs

match the pre-stored data, the door unlocks. Also, the user has three chances to unlock the door, if all three failed, the system will enter protected state, which disables the face identification system. The user can only unlock the door using a key.
The probability for an intruder to pass a test is $5\%^9$. We assume this probability to be independent of the others because each sample will be randomly chosen by our algorithm. The probability of an intruder successfully breaks in should be: P = first test passed + first test did not pass but second test passed + first and second tests did not pass but third test passed. Since there is 5% false positive rate and 85% true positive rate, we can calculate P like this:

$$P_{BreakIn} = (5\%)^9 + (85\%)^2 * (5\%)^9 + ((85\%)^2)^2 * (5\%)^9 \ = 4.384 * 10^{-12}$$

We will also continue to try to improve our facial recognition algorithm to reduce false positive and false negative rate.

On the other hand, we have to also make sure the completion time for one operation cycle is short enough. It is clearly not a good idea to let the user wait for a couple of minutes before they can enter their place.

First user activates the system using voice command over 60 decibels, which is the average level of human voice volume. Since the SEN12642 sound detector uses analog signal, we consider it to be real time.

After the activation signal is sent, USB camera starts to take pictures. It takes one picture in around 0.01667 second. For aforementioned reason, we would like it to take more than a few pictures. So we would give it one second to take pictures.

The pictures are sent to SD card in PYNQ module. The write speed and read speed of our SD card, according to the datasheet, should be around 50MB/sec and 90MB/sec. The amount of data we write and read is:

$$D \ = \ 3 * 640 * 480 * 60 \,/\, 10^6 \ = \ 55.629MB$$

Thus the total read and write time is 1.73 seconds.

Then we want to make sure the computation time of our algorithm is acceptable for a real-time face identification system. Since we have not implemented the code completely, we can only offer an estimation based on our experience. With trained model, our algorithm should respond as quickly as 0.01- 0.1 second per picture. Thus an estimation of total response time should be 0.1 - 1 second.

At last the MCU sends unlock signal to motor driver, which we tested: it takes less than one second to unlock.

Consider propagation delay, MCU operation time, a close estimation of one operation cycle (unlock attempt) is:

$$t \ = \ 1.73 \ + \ (1 - 0.1)\,/\,2 + 1 + 0.5 = 3.68 \pm 0.45 \ seconds$$

It does not take much longer than finding the key in your pocket or bag, take it out and unlock the door.

2.6 Software

2.6.1 Face Detection

One important process before the face identification would be the image pre-processing. For image inputs that have too much noise, unclear or contains no actual face, we do not want them to be considered during the test process. To achieve that, we will need face detection on every raw image.

We will be using Adaboost with image integral to detect and select valid face in the images. The boosting algorithm will help us on strengthening our neural network classifiers by adjusting weights in the image. The detection process could be done using image integral that helps to find face features, or Haar-like features.

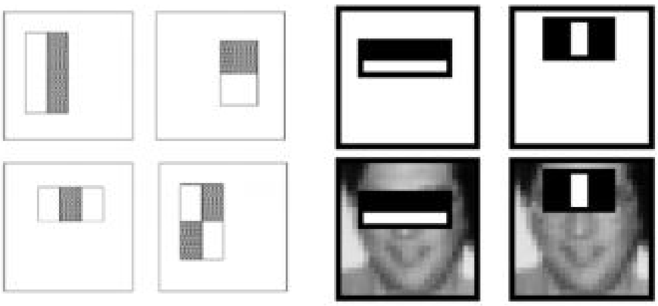


Figure 6. Rectangular Filters [11]

The feature output, as shown below, is the difference between two adjacent regions. One way of implementing fast computation of such difference is the image integral, so that every pixel is the sum of its neighbors to the upper left. Adaboost is used to choose the right and robust filter among various rectangle filters.

$$value = \sum(pixels\ in\ white) - \sum(pixels\ in\ black)$$

Formula 1. Feature Output Value[11]

Once this process is done all of the qualified face image data will be sent to the recognition algorithm for face identification.

2.6.2 Face Recognition

The software in our project is essential to accomplish our goal of face identification. We use a Local Binary Pattern (LBP) algorithm to approach the problem.

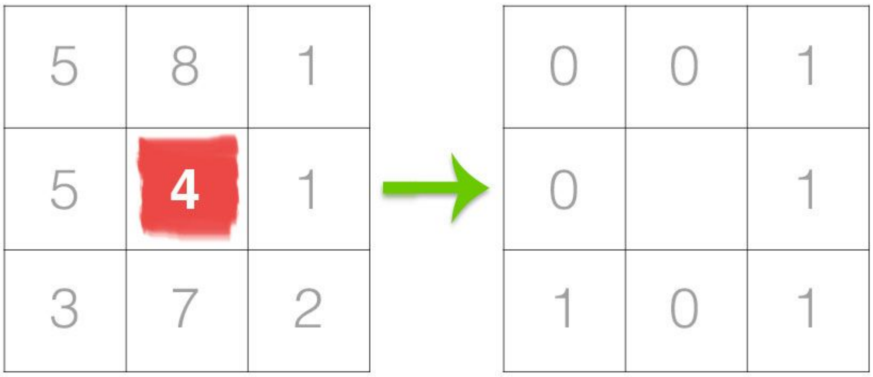


Figure 7. Comparing Local Weights of a Pixel with its Neighbors [10]

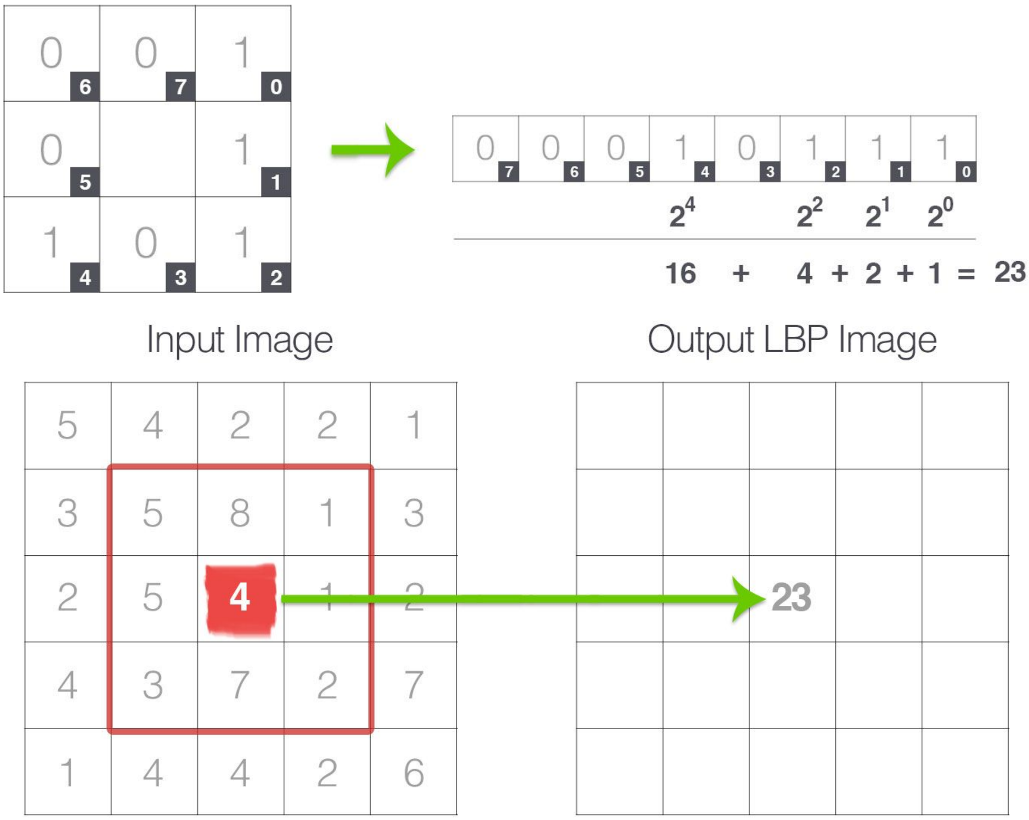


Figure 8. Computing a Binary Representation of a Pixel [10]

As shown in figure 7 and 8, LBP computes a binary value of an input image's local weights to represent a pixel in output LBP image. And the output LBP image can be put into consideration of LBP uniformity, which is the number of 0 to 1 and 1 to 0 transitions in the binary string[10].

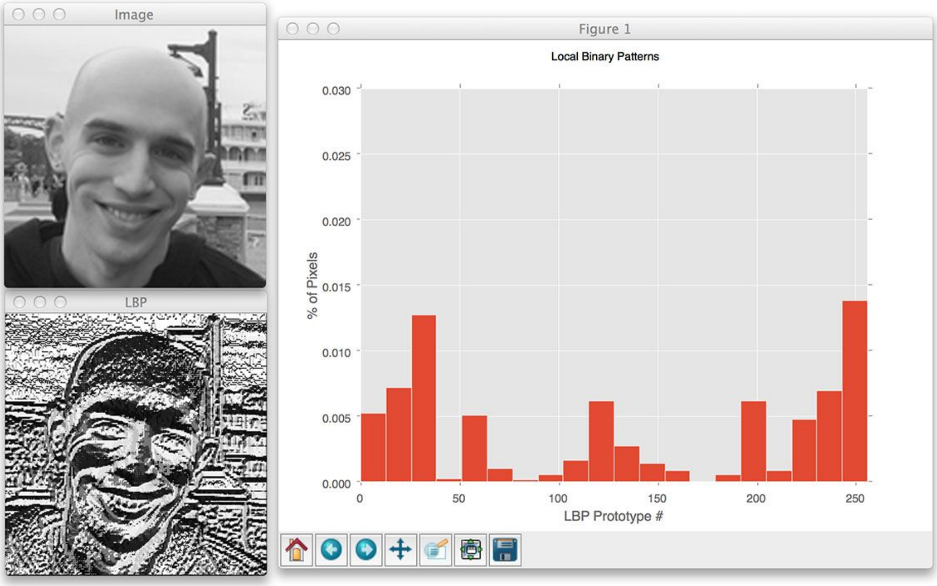


Figure 10. LBP on a Sample Face Image[10]

Figure 10 shows the histogram of local binary patterns for a face image. The trained model should have a classifier that best describes the local binary pattern of the user's face and thus be able to distinguish different faces by computing the percent difference in its binary pattern histogram.

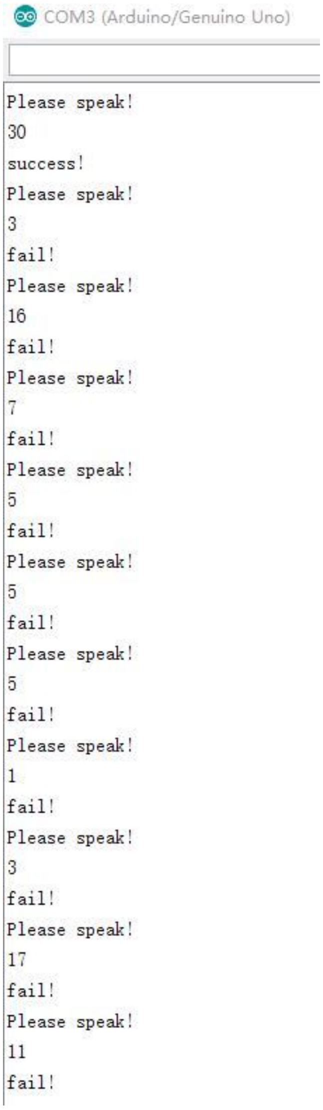
2.6.3 OpenCV

We installed OpenCV with its add-on face library on PYNQ board, and used CascadeClassifier and LBPHClassifier functions to help us implement face detection and recognition algorithms.

3 Verifications

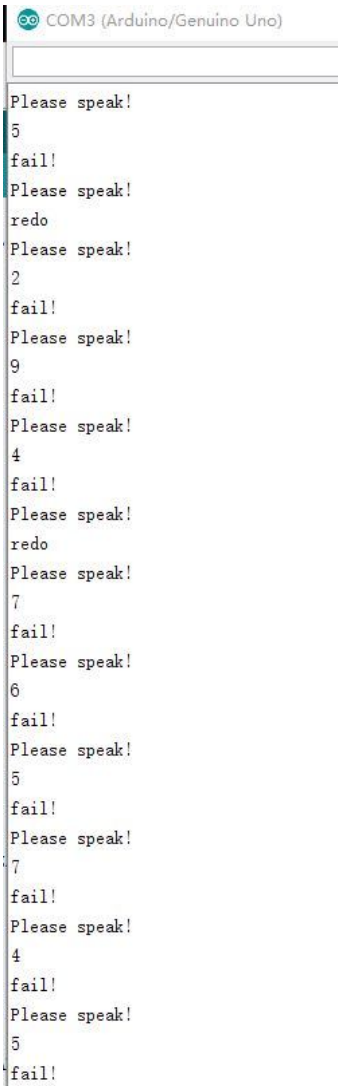
3.1 Sound detection

For sound detection, after we trained Zan's voice, we let him to test it 20 times with his voice of "hey" on Arduino. It success 17 times out of 20, which gives true positive rate of 85%. Also, we have tried other sounds for 20 times. The false positive rate is 1/20=5%



COM3 (Arduino/Genuino Uno)

```
Please speak!
30
success!
Please speak!
3
fail!
Please speak!
16
fail!
Please speak!
7
fail!
Please speak!
5
fail!
Please speak!
5
fail!
Please speak!
5
fail!
Please speak!
1
fail!
Please speak!
3
fail!
Please speak!
17
fail!
Please speak!
11
fail!
```



COM3 (Arduino/Genuino Uno)

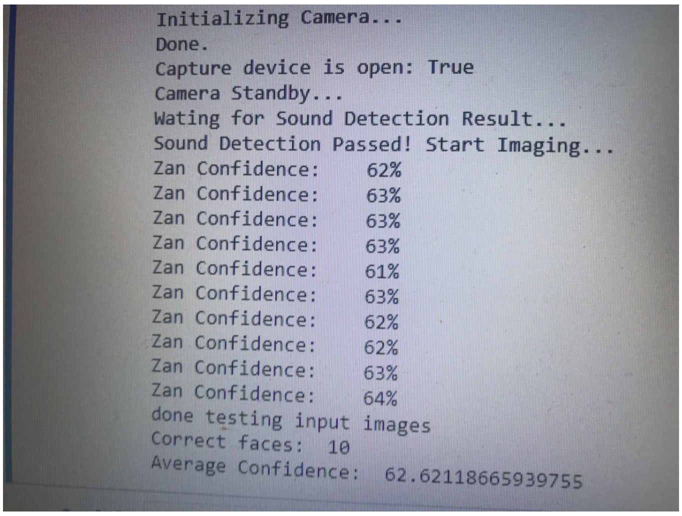
```
Please speak!
5
fail!
Please speak!
redo
Please speak!
2
fail!
Please speak!
9
fail!
Please speak!
4
fail!
Please speak!
redo
Please speak!
7
fail!
Please speak!
6
fail!
Please speak!
5
fail!
Please speak!
7
fail!
Please speak!
4
fail!
Please speak!
5
fail!
```

Figure 11. Testing Using Zan’s Voice (left)

Figure 12. Testing Using Zekun’s and Kaiji’s Voice (right)

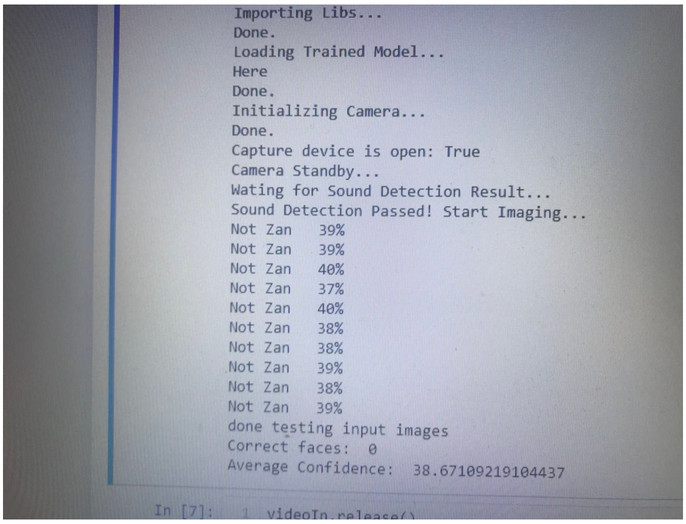
3.2 Face detection

For face detection, we trained Zan’s face with 250 pictures. We tested Zan’s face for 200 times, only 1 failed, which gives true positive of 99.5%.We tested Zekun’s and Kaiji’s faces each for 200 times with random little change of facial expressions. Only 3 of them passed in total, which gives false positive of 0.75%



```
Initializing Camera...
Done.
Capture device is open: True
Camera Standby...
Wating for Sound Detection Result...
Sound Detection Passed! Start Imaging...
Zan Confidence:    62%
Zan Confidence:    63%
Zan Confidence:    63%
Zan Confidence:    63%
Zan Confidence:    61%
Zan Confidence:    63%
Zan Confidence:    62%
Zan Confidence:    62%
Zan Confidence:    63%
Zan Confidence:    64%
done testing input images
Correct faces:  10
Average Confidence:  62.62118665939755
```

Figure 13. Test Case on Zan’s Faces



```
Importing Libs...
Done.
Loading Trained Model...
Here
Done.
Initializing Camera...
Done.
Capture device is open: True
Camera Standby...
Wating for Sound Detection Result...
Sound Detection Passed! Start Imaging...
Not Zan   39%
Not Zan   39%
Not Zan   40%
Not Zan   37%
Not Zan   40%
Not Zan   38%
Not Zan   38%
Not Zan   39%
Not Zan   38%
Not Zan   39%
done testing input images
Correct faces:  0
Average Confidence:  38.67109219104437

In [7]:   1  videoIn.release()
```

Figure 14. Test Case on Zekun’s Faces

4 Costs

4.1 Labor:

Our estimation of hourly wage is estimated based on majors.

Name	Estimated Hour	Estimated Labor Cost

Kaiji Lu	90	$3600
Zan Chen	90	$3600
Zekun Hu	90	$3600
Total	270	$10800

Table 1. Labor Cost

4.2 Parts:

Part Number	**Part Name**	**Manufacturer**	**Description**	**Quantity**	**Cost**
C920	USB Camera	Logitech	USB Camera	1	$28.00
CEM-C9745JAD462P2.54R	Microphone	SparkFun	Electret Microphone	1	$11.95
SEN13959	HC SR04 Ultrasonic Sensor	SparkFun	Proximity sensor on PCB	1	$3.95
ATMEGA328P-AU	ATMEGA328P-AU	MICROCHIP	Microprocessor on PCB	1	$13.00
ROB-09065(Sub-macro size)	Servo	SparkFun	Lock system	1	$8.95
PC6	DC Power Converter	Schumacher	Power unit	1	$18.39
ZYNQ XC7Z020-1CLG400C	PYNQ-Z1	Digilent	Recognition unit	1	$229.00
16G UHS-1 U3 Class10 TF	Micro SDHC U3 Card	Amplim	RAM for ZYNQ	1	$13.92

FD-5WSRGB-A	10mm Diffused RGB LED	FEDY	LED in lock system	2	$4.00
LM317MBSTT3	LM317	TI	Power Regulation Unit	2	$9.40
ECS-160-20-3X-EN-TR	Crystal Oscillator	ECS	CLK for PCB	1	$0.46
KS-01Q-01	Switch	E-Switch	Reset system	1	$0.53
Total Cost	$341.55				

Table 2. Parts Cost

4.3 Grand Total:

Grand Total = Labor Cost + Parts Cost = $11141.55

5 Ethics and Safety

There are some ethical concerns with our project. As our project includes a facial identification, we inevitably need to process people’s photos. We here promise that all the information through our technology is confidential as we should not go against the IEEE code of Ethics. As #1 and #9 mentions, we have to “disclose promptly factors that might endanger the public or the environment ” and “avoid injury other’s property” [5]. We not only need to make sure we do not misuse the data but also we have to carefully protect our data.

Since our project relates to a door lock , we have to try our best to improve the accuracy of our algorithm. As following the IEEE code of Ethics #7 , we humbly accept helpful advices or criticisms from all sources and we ensure that we credit these contributions properly. Besides, according to the IEEE code of Ethics #3, even if our final project does not come to be as accurate as we expect initially, we promise to report the true result and give justified claims. [5] We will truly report the outcome and statistics of our final project and compare them with our requirements.

The safety of our physical design is relatively high. We may have potential overheat for several chips and a motor which controls the lock. However, we do need to pay attention to our data safety when we use convolutional neural network to accomplish the goal of face identification. We have to watch out whether the data are going somewhere else when we did not intend. That’s a big safety concern and an ethical issue as we talked before.   All cloud services based on the US, the UK, France and other jurisdictions known to be tolerant of NSA-style snooping, which means the data we store on the cloud services all have potential danger of being accessed by others [6]. Thus, it is important for us to protect these identifiable information that we collect during our trials. One potential way to keep our data safe is to encryption and avoid storing too much data on the cloud services.

Finally, we aim to create good, safe and convenient intelligence system for others. To meet ACM Code of Ethics and Professional Conduct #1.1 , we dedicate to create a “safe social environment” as well as a “safe natural environment” [7]. That’s why our product aims to be lower cost than a surveillance camera but also high-accuracy promised intelligence system.

References

[1], Y.An *CNNs for Face Detection and Recognition*, 2017 [Online]. Available http://cs231n.stanford.edu/reports/2017/pdfs/222.pdf [Accessed: 07-Feb-2018]

[2]prnewswire.com(2018), “Global Home Security Products and Solutions Market to Reach US$ 12 Bn by 2025”. [Online]. Available: ”https://www.prnewswire.com/news-releases/global-home-security-products-and-solutions-market-to-reach-us-12-bn-by-2025-640292763.html)
[Accessed: 08-Feb-2018]

[3] H.Htlar Lwin , “Automatic Door Access System Using Face Recognition” “http://www.ijstr.org/final-print/june2015/Automatic-Door-Access-System-Using-Face-Recognition.pdf”) [Accessed: 07-Feb-2018]

[4] R. Zhao, W. Song, W. Zhang, T. Xing, J. Lin, M. Srivastava, R. Gupta, Z. Zhang, “Accelerating Binarized Convolutional Neural Networks with Software-Programmable FPGAs”[Online] Available “http://www.csl.cornell.edu/~zhiruz/pdfs/bnn-fpga2017.pdf” [Accessed: 08-Feb-2018]

[5] ”IEEE Code of Ethics”, ieee.org, 2017. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 07-Feb-2018]

[6] Acm.org. (1992). ACM Code of Ethics and Professional Conduct. [online] Available at: https://www.acm.org/about-acm/acm-code-of-ethics-and-professional-conduct [Accessed 07-Feb-2018].

[7] J.Naughton (2013), “Internet security: 10 ways to keep your personal data safe from online snoopers” [Online]. Available: https://www.theguardian.com/technology/2013/sep/16/10-ways-keep-personal-data-safe
[Accessed: 07-Feb-2018]

[8]Byron Jacquot(2014), “sound-detector” [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Sound/sound-detector.pdf
[Accessed: 17-Feb-2018]

[9]”Schematic Prints” [Online]. Available:
http://www.haoyuelectronics.com/Attachment/OV7725 + AL422B(FIFO) Camera Module(V1.0)   [Accessed: 21-Feb-2018]

[10]A. Rosebrock ,“Local Binary Patterns with Python and OpenCV” [Online]. Available:
https://www.pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/
[Accessed: 21-Feb-2018]

[11] L. Shapiro, “Recognition Part II: Face Detection via Adaboost” [Online]. Available:
https://courses.cs.washington.edu/courses/cse455/16wi/notes/15_FaceDetection.pdf
[Accessed: 21-Feb-2018]

RV Table:

Name	Requirements	Verification
Power Regulation (5 pt)	1.Able to outputs a voltage 4.5V~5.5V and a voltage 3V~3.6V separately. (5 pt)	1. Supply regulator with 9V DC and measures if it is able to produce two different voltages as desired.
Microcontroller (7 pt)	1.Able to receive analog/digital input and deliver analog/digital output. (5 pt)	A. Load a test program via Arduino Uno. B. Run program with Serial Monitor open. Attach function generator to pin 5. Attach oscilloscope to pin 14. C. Inspect whether it can receive signal from pin 5 and output signal to pin 14.
	2.Correctly complete FFT transform for input wave. (2 pt)	A.Load a test program via Arduino Uno. B. Run program with Serial Monitor open. Feed a signal from the function generator and run the fft algorithm. C. Check if the output on the monitor matches the input from function generator. (Inspect for its major peak)
Distance detection on microcontroller (3 pt)	1.Microphone only activates when the ultrasonic sensor detects an object within 30~35cm. (3 pt)	A. Load the activation program via Arduino Uno. B. Run program with Serial Monitor open. Check if it output matches the distance between the object and the sensor.
Sound Recognition on microcontroller (12 pt)	1.Able to detect a certain individual sound [ə] (“ah”) with an accuracy of 65%~75% at one trial. If success, a green led will blink and a PWM signal will be generated from pin 14. (6 pt)	A. Plug power to our project. Put an object close to the ultrasonic sensor so it always passes the detection. B. Make individual sound [ə] 20 times and see if at least 13 of them are successfully detected.

	2.Able to recognize a different sound other than [ə] as a failure with false positive rate under 10%~15%. If no sound exceeds the threshold within five seconds, it also considered as a failure. If failure, a red led will blink (5 pt) 3.If it has successfully detected the sound [ə] , then it will activates the camera and begins face identification. Otherwise, if it has detected two failures, the system will have 10 seconds cooldown and loop back to ultrasonic sensor. (1 pt)	A. Plug power to our project. Put an object close to the ultrasonic sensor so it always passes the detection. B. Make other individual sound like clapping or knocking the desk 20 times and check if no more than 3 of them are detected as [ə]. C. Also make no sound for five seconds and see if it also considered as a failure. A. Plug power to our project. Put an object close to the ultrasonic sensor so it always passes the detection. B. Make random individual sounds to have two failures. And check if the system will have 10s delay and loop back to the ultrasonic sensor part.
Lock Operation on microcontroller (3 pt)	1.The servo is able to rotate 360 degrees and after five seconds it will rotate backwards. (3 pt)	Manually feed an activate signal to the lock and see if it acts as intended.
Camera (3 pt)	1. Able to catch meaningful pictures of the user in a given time period. (2 pt) 2. Able to produce at least 30 of 640*480 images per second. (1 pt)	For both requirements, we can manually feed an activation signal to the camera and check if its output images meet the requirements.
Face Identification on PYNQ (17 pt)	1.Able to receive a PWM signal from microcontroller. (1 pt) 2.Able to identify a specific individual's face with an accuracy of…. If succeeds, the green led will	A. Use a function generator and generates a PWM signal to the PYNQ board. B. Check on the computer to see if the PYNQ board successfully receives the signal. A.Plug power to our project and manually feed an activation signal to the

	blink and the lock module will start to rotate. (8 pt)	camera. Let the person who trained the data standing in front of the camera. B. After running the algorithm check if the face identification process is successful. C. Repeat A-B 20 times and check if the accuracy meets the requirements.
	3. Able to recognize other people’s faces as a failure with a false positive rate below than … If fails, the red led will blink. (7 pt)	A.Plug power to our project and manually feed an activation signal to the camera. Let other people stand in front of the camera. B. After running the algorithm check if the face identification process fails. C. Repeat A-B 20 times and check if the false positive rate meets the requirements.
	4. After experiencing three failures, a red led should be constantly high and the whole system will be locked. (User can only use a door key at this point.) (1 pt)	When testing requirement 3, also check if the whole system terminates after three failures and the red led is always HIGH.

Appendix A:

MCU Indications:

1. If proximity sensor detected an object within a short distance, then starts sound detection and Green light blinks once
2. If sound detection fails one time: Red light blinks once
3. If sound detection successes: Green light on and transmit the signal to PNYQ board (to activate camera and begin face detection)
4. If sound detection fails twice: Red light blinks for ten seconds and loop back to the beginning
5. If face detection successes: Green light and Red light both on, lock starts to rotate, 3 seconds later, rotates back and both lights off
6. If face detection fails one time: Red light blinks twice, then red light off , green light on , start another trial
7. If face detection fails more than two times: Red light constantly on, and the whole system is locked.