Z-RECOGNITION

License Plate Recognition

AUTHORS: PATRICK ZIAJSKI AND ZARAK KHATTAK

SUBJECT: TECHNOLOGY PROJECT – TPJ655

PROGRAM: COMPUTER ENGINEERING TECHNOLOGY

INSTRUCTOR: B. SHEFLER

DATE: NOVEMBER 28, 2019

# EXECUTIVE SUMMARY

Z-Recognition is a License Plate Recognition system which provides clients with a new and intuitive way to implement and enforce parking systems. Our goal is to allow for the successful implementation of both hardware and software elements to allow for a working project. The hardware aspect of the project involves the implementation of a microcontroller which holds all the pre-required software to run the system. A camera that is used to capture images of License plates. A speaker and a Servo Motor. The software aspect of the project entails the creation of a python script which brings together all the hardware elements and have them operate with each other. The script captures an image which is processed as a JavaScript Object Notation (JSON) using Microsoft Azure Computer Vision to process the image. The captured data is then cross referenced against a MongoDB database for validity and the Servo Motor operates in response to valid data being found, whereas, the speaker operates if no matching data is found.

TABLE OF CONTENTS

[EXECUTIVE SUMMARY 1](#_Toc25861071)

[TABLE OF FIGURES 3](#_Toc25861072)

[TABLE OF TABLES 3](#_Toc25861073)

[INTRODUCTION 4](#_Toc25861074)

[FUNCTIONAL FEATURES 5](#_Toc25861075)

[SYSTEM SPECIFICATIONS 7](#_Toc25861076)

[OPERATING INSTRUCTIONS 8](#_Toc25861077)

[PRODUCT DESIGN, IMPLEMENTATION, AND OPERATION OF THE SYSTEM 9](#_Toc25861078)

[SYSTEM BLOCK-DIAGRAM 9](#_Toc25861079)

[SOFTWARE FLOW-CHART 9](#_Toc25861080)

[COMPONENT DESCRIPTION 10](#_Toc25861081)

[THEORY OF OPERATION 11](#_Toc25861082)

[HARDWARE 11](#_Toc25861083)

[SOFTWARE 12](#_Toc25861084)

[REQUIREMENTS FOR OPERATION 14](#_Toc25861085)

[MAINTENANCE REQUIREMENTS 15](#_Toc25861086)

[CONCLUSION 16](#_Toc25861087)

[FURTHER DEVELOPMENTS 17](#_Toc25861088)

[APPENDIX 18](#_Toc25861089)

[PARTS LIST / BILL OF MATERIALS 18](#_Toc25861090)

[CREDENTIALS 18](#_Toc25861091)

[BIBLIOGRAPHY, REFERENCES AND CITATIONS 19](#_Toc25861092)

[CONTACT INFORMATION 20](#_Toc25861093)

# TABLE OF FIGURES

[Figure 1: System Block Diagram 9](file:///C:\Users\patri\Documents\TPJ_FINAL_REPORT.docx#_Toc25861094)

[Figure 2: Software Flow Chart 9](file:///C:\Users\patri\Documents\TPJ_FINAL_REPORT.docx#_Toc25861095)

[Figure 3: Z-Recognition Graphical User Interface (GUI) 12](file:///C:\Users\patri\Documents\TPJ_FINAL_REPORT.docx#_Toc25861096)

# TABLE OF TABLES

[Table 1: Bill of Materials 18](#_Toc25861097)

# INTRODUCTION

Security is an important feature for a company providing a service to their customer. The company must ensure that its service can only be used by authorized personnel. This is where Z-Recognition comes into play. Z-Recognition is a license plate recognition system that ensures only authorized personnel have access to a designated area. Z-Recognition accomplishes this by taking an image of the approaching vehicle and processing it. If the vehicle is authorized, meaning the license plate is recognized, the vehicle is given access. Our group has decided on a license plate recognition system due to its uses in modern day society. Today, one can still find areas that are monitored by a single employee, or by a ticket-based entry system. These systems could be found inferior because of their reliance on periodic human interaction or supervision. Z-Recognition is designed to run autonomously with minimal costs, and only requires an active internet connection. This license plate recognition system also implements a core feature in the future of computer technology, machine learning. With machine learning, a prediction model can learn and improve without being explicitly programmed. This means that automated systems become more secure and reliable, without the need of constant supervision. Finally, this project requires both hardware and software implementation to be fully functional. It requires us to work with and learn both software development and hardware assembly which we, as Computer Engineering and Technology students would prefer.

# FUNCTIONAL FEATURES

* **Image Processing** – Using Microsoft’s Cloud based Computer Vision Service, called Azure, a captured image will be uploaded to Azure’s Service. The service performs Optical Character Recognition (OCR) on the image, and it will return a JavaScript Object Notation (JSON) object with all the recognized characters. This JSON object is then cross checked against a MongoDB database which contains a list of authorized vehicular license plates. To improve the accuracy on the returned Azure OCR, Image Binarization is implemented.
* **Motor Control** – The Raspberry Pi 3 Model B, a Microcontroller Unit (MCU), will be responsible for controlling a servo motor. This motor will imitate a traditional gate arm, which is used to allow entrance into a secured area generally used for parking.
* **Sound Execution** – The Raspberry Pi will emit a sound from a speaker in order to audibly notify the individual operating the system. The sound produced will be a 440Hz wave.
* **Scheduled Script Execution** – The Pi is compiled with a python script “ZRecognition.py”, this script is responsible for operating the whole system. Once executed, this script will run consistently until it is terminated by the user. This script is responsible for launching a Graphical User Interface to allow the user to capture images.
* **Python Tkinter Graphical User Interface (GUI)** – The presence of a GUI makes it much easier to see what is happening behind the scenes. The GUI has a select few, easy to use options described below:
  + Manual Capture – Grants users the ability to manually capture images. Selecting this option will require the user to click and select this option whenever they desire to capture an image
  + Timed Capture – Once this option is selected, the system will start to capture images at a timed rate that can be set by the user.
  + Stop Timed Capture – This option stops the timed interval image capture process
  + Timer Drop Down – Provides the user with the ability to select the time (in seconds) interval between each image capture. The drop-down menu ranges from 3 to 7 seconds.
  + The Image capture is displayed in two separate image boxes. The left most image is captured in real time and leaves it unmodified. The right most image capture is binarized, making it easier for the OCR model to recognize text, and also display regions around the text that has been recognized.

# SYSTEM SPECIFICATIONS

**Camera:**

* 720p / 30 FPS
* Fixed Focus
* Field of View: 60°

**Raspberry Pi 3 Model B**

* Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
* 1GB RAM
* BCM43438 wireless LAN
* 40-pin extended GPIO
* 4 Pole Stereo output
* Micro SD Memory Card

**Servo Motor:**

* Operating Voltage: 3.0V ~ 7.2V
* Operating Speed: 0.12second / 60°
* Temperature Range: -30 to +60

**Microsoft Azure:**

* Active student subscription
* Cognitive Services resources

**Mongo Database:**

* Access to MongoDB JSON database (active service required for operation)

# OPERATING INSTRUCTIONS

Please follow the listed steps to ensure that the system is operating as intended:

* Ensure that the Raspberry Pi Microprocessor is powered on
  + Plug the power cord into the Raspberry Pi and connect to a power source
* Connect the Raspberry Pi to an Internet source
  + Using either an Ethernet cable or a Wireless Fidelity (Wi-Fi) access point
* Ensure that the Camera is connected properly to the Raspberry Pi via Universal Serial Bus (USB)
* Execute the Main Python Script named “ZRecognition.py” responsible for running the functionalities of the program
  + Once the script is activated, the program should be up and running as intended
* Produce a legible license plate in front of the image capturing camera
  + Ensure that the license plate produced is legible and consists of alphanumeric characters
  + Hold the license plate still, centered at the front of the capturing camera field of view

Once the image is captured, processed and authenticated, the servo motor will rotate as intended to simulate a gate arm. If no data is found, the speaker will output a 440Hz wave to indicate access denial.

# PRODUCT DESIGN, IMPLEMENTATION, AND OPERATION OF THE SYSTEM

## SYSTEM BLOCK-DIAGRAM

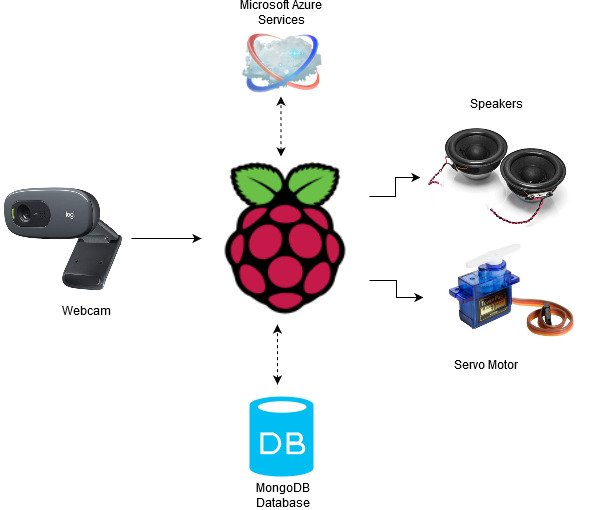


Figure : System Block Diagram

## SOFTWARE FLOW-CHART

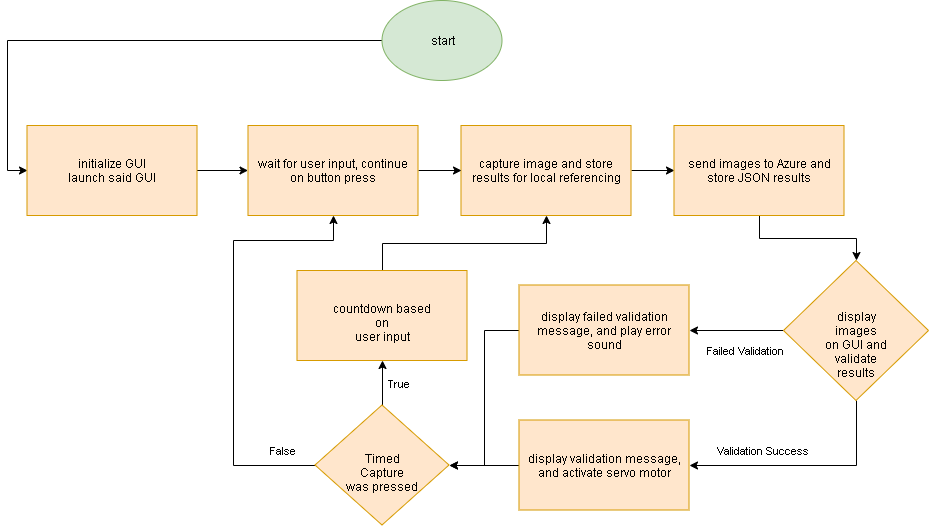


Figure : Software Flow Chart

## COMPONENT DESCRIPTION

The system utilized in this project consists of a few major components. Each component is listed and briefly described below:

* **Microprocessor – Raspberry Pi 3 Model B**
  + The Pi 3 model B is the sole microprocessor for this project. It contains a python script that is executed to run the program. The Pi makes use of a 1.2GHz Quad Core 64bit CPU with built in 1GB RAM. The Pi has been equipped with a 4GB Micro-SD memory card containing the operating system
* **Camera – Logitech C270 HD Webcam**
  + The Logitech C270 Webcam is installed to ensure that the images captured are of high quality making it easier than ever to interpret the images. These Images are taken in 720p with a fixed focus and a field view of 60°. The Images are then Interpreted by the Microsoft Azure Cognitive services and returned as a JSON object.
* **Servo Motor**
  + The RioRand SG90 Micro 9G Servo Motor is utilized in this project to simulate and resemble a gate arm opening and closing. Once the Image data has been authenticated by checking a database of existing and valid license plates, the gate arm will open and close if found. The Motor operates utilizing a voltage range of 3.0V to 7.2V

## THEORY OF OPERATION

The project can be broken down into two main components, Hardware, and the other being software.

### HARDWARE

* The hardware component of the project consists of a Raspberry Pi 3 Microprocessor, a camera, a speaker and a servo motor. These devices are bound together by a Python script.
* The servo motor will act as and simulate a gate arm found in abundance at most parking structures. It will have the ability to rotate 90° to visibly display the gate arm opening and closing.
* The speaker will be used to notify the clientele that access has been denied to them and the that access was not granted by generating a 440Hz wave.
* The Raspberry Pi 3 is responsible for hosting the operating system which will be responsible for running the python script which will operate the project.

### SOFTWARE

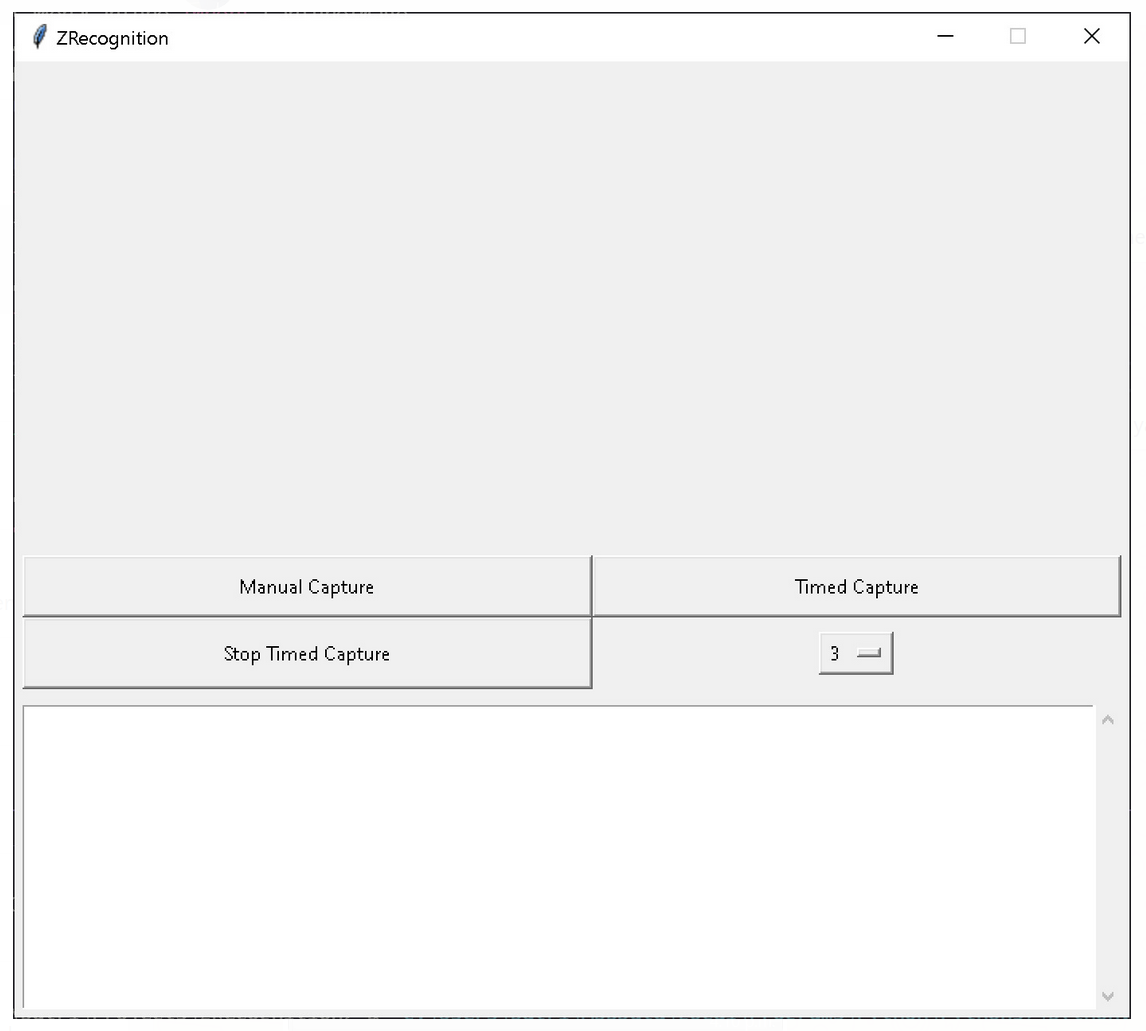
The Raspberry Pi contains within it the python project solution called “ZReconigtion.py”, which, when executed, launches a Graphical User Interface (GUI) pictured below. The GUI allows the user to either manually capture images or to take timed snapshots at a maximum interval of 7 seconds. A third button allows the user to stop the timed capture. 

Figure : Z-Recognition Graphical User Interface (GUI)

For both capture routines, manual and timed, a “take image” method is called. The difference is that for a timed capture routine, the method is called continuously separated by a user inputted interval and the manual capture routine is executed once. The “take image” method captures an image using the camera, saves the image locally, and makes a grayscale copy. A grayscale image is used because in order to binarize an image, which will greatly improve accuracy and consistent text recognition, there can only be color values between white, gray, black and any shade in between. The binarized image is then sent to the Computer Vision Service using an Application Programming Interface (API), which holds all necessary credentials and the binarized image’s bytes. The API, once the service completes its process, returns a JSON object storing all text extracted. The resultant data is then compared to all license plate values in the MongoDB database, which is a cloud-based Binary JSON (BSON) database with both paid and free tiers of subscriptions. A ratio is created based on the likeness of each stored database entry and the extracted text. Based on the ratio, if it meets or surpasses a software set threshold, a validation response will be executed. The validated response is set to output to the GUI log that the plate has been validated, and the servo motor will be rotated from 0° to 90° and then back down to 0°. For a failure to validate response, the servo motor is not activated and instead the script will create a soundwave at 440Hz, which is played through the speaker to alert the vehicle personnel that the license plate could not be validated.

## REQUIREMENTS FOR OPERATION

There are no major user required hardware components in order to operate the system, all hardware is provided along with the software portion, however, users will be required to sign up on MongoDB in order to have access to and utilize their database.

# MAINTENANCE REQUIREMENTS

Routine maintenance is a good way to avoid any unwanted interruptions of the services being provided. It is recommended that proactive maintenance be done on the system at a frequency of one week (7 days) to ensure product longevity, or anytime that the system fails to operate.

You can carry out the following steps to perform maintenance:

* Reset the Raspberry Pi
  + If possible, shut down the operating system on the Pi
  + Unplug the power cable from the Pi
  + Plug the power cable back into the Pi to power it on
* Relaunch the Python script
  + Once the Pi has been powered on, simply launch the “ZRecognition.py” script to start up the program

It is recommended that you test the service before deeming the maintenance completed. This will ensure that all systems are up and running.

Users can always get in touch with the contact staff should the user require assistance or clarification regarding the maintenance tasks or any other operational inquiries.

# CONCLUSION

Z-Recognition provides a more modern, efficient, scalable and simplistic approach of a security system. The design and implementation of this system required different aspects from many of the courses taken in Seneca College’s Computer Engineering Technology program. Skills and knowledge learned in ETD555, LNX155, NSP655, PRG469 and PRG655. However, the scope of this project required research above the foundations learned in these courses. For example, setting up and using the Azure Computer Vision resource, which required knowledge of JSON, Machine Learning, Image processing and more in-depth knowledge of Hypertext Transfer Protocol (HTTP). This project was successfully completed well within the initial projected timeline previously defined.

# FURTHER DEVELOPMENTS

Z-Recognition is a highly versatile system, making it easier than ever to increase or decrease its scalability with minimal workaround required. With the slightest modifications to the program, it can accommodate a wide range of situations and fulfil clientele needs. It could be deployed for single use garages, to an airport parking lot with multiple entrances and exits, multileveled parking buildings or even highway toll plazas.

Future extensions to this project may include but are not limited to:

* Motion detection for triggering the image capture and processing
  + Motion detection can help with power saving, activating the script and camera only when motion is detected
* Low Light detection / Night vision for improved Image capture
  + Allow for a better experience by capturing improved images in poorly illuminated areas
* Local computer vision service for a more reliable and independent system
  + A more controlled environment with less reliance on outside sources

# APPENDIX

## PARTS LIST / BILL OF MATERIALS

Table : Bill of Materials

|  |  |  |
| --- | --- | --- |
| BILL OF MATERIALS | | |
| Component Name | **Quantity** | **Price (CAD)** |
| Raspberry Pi 3 Model B | 1 | $35.00 |
| Logitech C270 HD Webcam | 1 | $32.00 |
| RioRand 5PCS x SG90 Micro 9d Servo | 1 | $16.99 |
| Gikfun 2” 8 Ohm 2W Audio Speaker | 1 | $16.88 |
| Total Cost w/ Tax | | **$113.98** |

**\*NOTE\*** *All prices listed above are in Canadian Dollars (CAD)*

## CREDENTIALS

Excluding any accounts or passwords required for accessing the cloud services used, which will need to be setup by the client, there are no log in credentials required to operate any aspect of this system.

## BIBLIOGRAPHY, REFERENCES AND CITATIONS

“Buy a Raspberry Pi 3 Model B – Raspberry Pi.” *Buy a Raspberry Pi 3 Model B – Raspberry Pi*, <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>.

“Logitech C270 HD Webcam, 720p Widescreen Video Calling and Recording (960-000694): Amazon.ca: Computers & Tablets.” *Logitech C270 HD Webcam, 720p Widescreen Video Calling and Recording (960-000694): Amazon.ca: Computers & Tablets*, <https://www.amazon.ca/gp/product/B004FHO5Y6/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&psc=1>.

“RioRand® 5PCS x SG90 Micro 9g Servo For RC Airplane Car Boat Genuine: Amazon.ca: Electronics.” *RioRand® 5PCS x SG90 Micro 9g Servo For RC Airplane Car Boat Genuine: Amazon.ca: Electronics*, <https://www.amazon.ca/gp/product/B00M8RGGFG/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&psc=1>.

“The Most Popular Database for Modern Apps.” *MongoDB*, <https://www.mongodb.com/>.

## CONTACT INFORMATION

**Patrick Ziajski**

Phone: (647) 339 2847

Email: pziajski@myseneca.ca

**Zarak Khattak**

Phone: (587) 226 3196

Email: zkhattak@myseneca.ca