Technological University of the Philippines
College of Engineering
ELECTRONICS ENGINEERING DEPARTMENT

DEVELOPMENT OF LICENSE PLATE RECOGNITION (LPR) SYSTEM FOR AUTOMATIC CAR BARRIER USING IMAGE PROCESSING AND ARTIFICIAL NEURAL NETWORK

by

Alacapa, Maria Angelica P.

Calangian, Adriane Roy F.

De Guzman, Romel A.

Merin, Emmanuel Joshua C.

Minguez, Grason B.

March 2018

Approval Sheet

Acknowledgement

The proponents of this study are beyond thankful to all those who have helped and supported the team to accomplish this project;

First of all to Almighty God for His unparalleled gift, the source of the members' wisdom, strength, courage, perseverance and to whom belong all the gratitude;

To their families for unending support and love. For their patience and understanding, as well as providing the moral and financial support the team needed to complete this project especially to Minguez residence for accommodating the members all throughout the making of the project;

To Mr. Reginald Taar, the University's head security for imparting his knowledge to the members especially in assisting them in the deployment of the project, for giving the proponents a chance to extend their research to University;

To Engr. Timothy M. Amado, their mentor and adviser, to the panel members and to ECE faculty, for giving guidance and suggestions for further improvement of the project. Their expertise greatly help in refining the researchers thinking and the project itself ;

This project was made possible through the joint effort of its proponents and is dedicated to all those who contributed and inspired the team to reach its success.

Abstract

Right level of security is necessary to ensure safe environment for everyone within the vicinity and monitoring the entrance and exit of vehicles is a primary task in security management. License Plate Recognition (LPR) system using image processing is a technology specially design for identifying vehicles and it became significant due to difficulty of human to identify vehicles as its population increases. LPR system is helpful in various applications such as traffic control, parking system and toll gates and integrated in this study with automatic car barrier. This system involves image acquisition, image processing, character recognition and database management. Image acquisition is where the image of license plate was taken with the use of an IP cam CCTV while image processing includes pre-processing, plate detection and character segmentation. Character recognition is the stage that uses Artificial Neural Network (ANN) and database management is where the license plate character matching takes place. The car barrier can operate automatically through database matching, remote-controlled if the plate is not recognized or not yet on the database and manual scheme in case of power interruption. The recognition accuracy is 88.89% and it is dependent on both license plate extraction and character segmentation performances.

Table of Contents

List of Figures

List of Tables

Table No. **Title** **Page No.**

List of Equations

Chapter 1

THE PROBLEM AND ITS SETTING

1.1 Introduction

Safety and security is essential in any company or organization to guarantee that they are free from any threat of danger, intruders, theft and other related crimes. Right level of security is necessary to ensure safe environment for everyone within the vicinity. LPR system became significant due to difficulty of human to identify vehicles as its population increases. There are numerous technology available for security purposes, including surveillance cameras, access control system, alarms which most establishments invest in. These technologies facilitate in monitoring the entrance and exit of vehicles which is a primary task in security management. In line with this, automatic vehicle identification system arises.

License plate recognition system using image processing is a technology specially design for identifying vehicles. License plate is the vital feature to be identified which is compose of characters; combination of letters and numbers and is unique in every vehicle. Among all the studies that have taken to consideration, the system main steps in algorithm comprises of the following: (i) license plate detection, (ii) license plate character segmentation; and (iii) license plate character recognition. The latter step is where the Artificial Neural Network is applied. It is a supervised learning algorithm which is important for it acts as a biological brain and responsible for training the system to learn recognize the license plate. LPR camera is widely use in various application for its built in software however it is usually costly. Studies that have been already developed to identify

Philippine License plate focused on the old plates. In this study, the new format of country's plate are trained for future application and is tested under possible actual set-up.

License plate recognition in this study is through image processing integrated by Artificial Neural Network (ANN). Image processing is for the analysis of digitized images while ANN is a computing system based on neural structure of brain working similarly by learning through trained samples. The chosen application is a security scheme that includes car barrier and database matching to provide additional and advance security. Identifying vehicles entering and leaving the facility which will intensify security system and in due course help in protection of lives and possession of all the individuals inside the facility is the center of this study.

1.2 Background of the Study

Car license plate is the plate attached to a vehicle that serves as an identification of the car owner. It is composed of characters; a combination of letters and numbers, which is unique on every vehicle. There are different kinds of vehicular license plates in the Philippines specifically for private vehicles, public utility vehicles (PUV), government vehicles, diplomatic vehicles, electric vehicles and motorcycle vehicles.

License plate recognition has many applications. Some of these are traffic related applications such as searching of stolen vehicles, road traffic monitoring, airport gate monitoring, speed monitoring and automatic parking lots access control (Sharma et al., 2014). LPR can also provide high security and there is no need for human interaction. In this system only authorized cars are allowed to enter. (Judith Sen et al., 2014).

Capturing an image of the car with the license plate may be taken by CCTV cameras or smartphones. Some of these cameras capture videos instead of images, in which the video is manually dissected into frames and undergo image processing. This process yields license plate detection, license plate character segmentation and license plate character recognition (Caner et al., 2008).

In license plate detection, either edge detection or threshold is involved (Yohimori et al., 2004; Hongliang et al., 2004). Some of which tried to use the color, shape and texture of the license plate (Ron et al., 2002). In license plate character segmentation, connected component labeling (CCL) analysis is used in segmentation of the characters (Wang et al., 2003). The last step is the license plate character recognition. In this step, Caner, Gecim, and Alkar used self-organizing map (SOM) which involves artificial neural network that can be used in wide range of applications including optical character recognition (OCR) (Caner et al., 2008).

1.3 Statement of the Problem

License Plate Recognition (LPR) has been the focus of many studies since 1990s for the advancement of traffic control and surveillance systems. Subsequently, many methods have been devised to address problems such as processing rate, computational capability and most importantly the recognition rate of the software. Since all these methods have different approach, it seems improper to explicitly proclaim which system is best and holds high performance and accuracy.

Also, several areas provides no special security requirement that causes fraudulent practices. With this system, only registered license plates is granted for a quick entrance.

Thus, this study is established to design a License Plate Recognition system through Artificial Neural Network most suitable to the application of automatic car barrier system for security of the recipient.

1.4 Objectives

1.4.1 General Objective

This study aims to develop a neural network-based license plate recognition (LPR) system using image processing.

1.4.2 Specific Objectives

1. To design and control an automatic car barrier using microcontroller.
2. To develop a program for extraction of license plate region and character recognition through back-propagation neural network.
3. To provide an automated vehicle logging system for the university.
4. To implement the system for facilitating in the safety and security of Technological University of the Philippines – Manila in relation to vehicles.

1.5 Significance of the Study

The study will provide development to the idea of producing an equipment for the recipient's security system.

Technological University of the Philippines-Manila (TUP-M) will benefit in this study for the convenience and improvement of its security. They will be able to easily identify the owner's information from database through license plate recognition. Thus it will lessen the time and human interference between the driver and personnel. Since there is a need of database, high security is also provided in this study. With a mismatched license plate from database, vehicle is obliged to undergo inspection by the personnel before passing the automatic car barrier. Consequently, the study will provide a suitable overall performance for the said application.

1.6 Scope and Limitation

The study is proposed to design an automatic License Plate Recognition system through LPR camera in conjunction with Artificial Neural Network (ANN). This study will have evident impact in traffic surveillance, vehicle tracking and security.

The study deals with detection and recognition of vehicle's plate number through image processing of still images obtained by the CCTV camera. Recognition of the license plate will be done for the incoming and outgoing vehicles in the area. The system can identify the new Philippine license plate formats during daytime and nighttime however in nighttime, plate extraction must be done in places with enough brightness or fixed illumination. The study is only limited to cars, bus, trucks and does not cover those with illegible and damaged plates due to wear and tear since it may not represent the actual information of the plate number. The system can only permit those vehicles included in the database's whitelist, otherwise it will require human intervention for inspection. Identification of the driver is not part of the study and the program can only reflect names that are registered in the database. In addition, weather conditions that can greatly affect the accuracy of the recognition like heavy rains are considered out of scope.

1.7 Definition of Terms

License Plate – a rectangular metal plate consists of numbers and letters attached to vehicles' bumper as an identification for registered vehicles.

License Plate Recognition (LPR) – is a technology that uses image processing to identify vehicle's license plate.

Closed-Circuit Television (CCTV) –a video surveillance that uses video cameras to record motions and to transmit a signal to a specific place, on a limited set of monitors. It will be used to capture the license plate to be processed by the LPR system.

(https://en.m.wikipedia.org/wiki/Closed-circuit_television)

Artificial Neural Network (ANN) – it is a computing system based on a biological brain composed of highly interconnected processing elements, which process information by series of samples that will be fed to the system.

(https://www.tutorialspoint.com/artificial_intelligence/ artificial_intelligence_neural_networks.htm)

Automatic Car Barrier – A boom barrier or boom gate, is a bar used to allow or block vehicle access through a controlled point. It usually rises up 90 degrees vertically.

(https://en.wikipedia.org/wiki/Boom_barrier)

Chapter 2

REVIEW OF RELATED LITERATURES AND STUDIES

This chapter presents the review of related literatures, studies and patents significant to the project.

2.1 Related Studies

2.1.1 Parking System

There are numerous ways of implementing parking system. In the study of *A Real-Time License Plate Detection System for Parking Access* by Fajar Astuti Hermawati et al. in 2010, it addresses the problem on how to recognize vehicles that enter the parking lot and how to recognize various types of license plates in different light conditions automatically and precisely in just a second. The system is used as an access control entry of vehicles into the parking area. The study proposes the use of detection system for the recognition of vehicles and License Plate Recognition (LPR). The detection system uses a detector distance circuit using distance sensor PING. Microcontroller Basic Stamp will count the time needed to receive the ultrasonic waves and determine the distance to the car. License Plate Recognition (LPR) employs Fourier transform to detect the location of license plate and Hidden Markov model method for the algorithm character segmentation. The proposed system is successfully implemented with various types of Indonesia license plates; the private vehicles, the government vehicle and public transportation under different light condition and camera position. The result of the experiment shows that the proposed method is reliably precise, despite of some images that failed to be identified with an efficiency of 84.38 % for plate recognition in the real-time experiment and the average execution time for all recognition process is 5.834 second. This study suggests further research and focus on algorithm that will improve the result for low resolution images.

In *Research of Automatic Parking system based On License Plate Recognition* by Qing Tian et al. in 2013 discussed a parking system capable of recognizing plate numbers automatically integrated by VC++ and QT programming. Due to urbanization and rapid vehicle ownership, the need of intelligent traffic system in which License Plate Recognition plays a vital role. The automatic parking system is consist of hardware architecture (automatic barriers, cameras, and computer) and software implementation. Automatic barriers are used to allow or permits a certain vehicle in the entrance and exit of the parking area. Cameras are used for frame by frame video streaming that are transferred to computer for detection and recognition purposes of the license plate. The plate detection is done by linear detection and thresholding while segmentation through combined method of vertical and horizontal projection. The system includes a database that can provide for data query and a friendly interface for managing vehicles information. This database will record vehicle details such as license plate and entrance time. The system's result shows the average processing time between 30ms to 100ms for it to recognize and store the information in the database. The overall accuracy for recognition is 72% however implementing it school entrance with proper position (front only) and angle leads to 97% accuracy.

A study, *Advanced License Plate Recognition System for Car Parking* by Judith Sen E et al. in 2014 discussed that an efficient and secured parking facilities needs a robust car parking system. The structure of the system contains two main sections, Radio Frequency Identification (RFID) and Open Computer Vision. RFID is contactless and secured identifier. It identifies vehicles through detecting and accessing the data in the RFID tag, and only authorized vehicles have it and are permitted to enter the parking area. The number plate recognition is done using Open Computer Vision. Obtained images through web camera is used as input which undergoes pre-processing before actual plate number recognition occurs. Open CV is used due to its capability to enhanced computational efficiency in real-time applications and the fact that it is free and promotes simple operations. This system offers time-saving and easy-parking for vehicle owners than manual system.

Table 1. Summary of Parking System using License Plate Recognition

Author	Year	Title	Remarks
Fajar Astuti Hermawati et al.	2010	A Real-Time License Plate Detection System for Parking Access	• The proposed system is successfully implemented with various types of Indonesia license plates. • It uses distance sensor PING, Hidden Markov and Fourier Transform. • average execution time for all recognition process is 5.834 second
Qing Tian et al.	2013	Research of Automatic Parking system based on License Plate Recognition	• The system has hardware architecture—automatic barriers, cameras, and computer and software implementation. • Uses Database for Data query and matching • average processing time between 30ms to 100ms
Judith Sen E et al.	2014	Advanced License Plate Recognition System for Car Parking	• Uses Radio Frequency Identification (RFID) based reserved parking system and Open Computer Vision for plate recognition.

2.1.2 Existing License Plate Recognition System

Numerous studies have been involved in creating their own approach in License Plate Recognition System that shows noteworthy and promising results.

License Plate Recognition (LPR) System discussed by Hui Wu et al. in 2011, is a computer-based LPR system with high accuracy at night. Currently, commercially-available LPR are capable of recognizing plate numbers of vehicles with speed up to 1200 km/h and has 95% to 98% reliability. Consequently, the study's main purpose is to improve the accuracy of the technology in their country—China. Their system includes selecting optimal frame, extracting license plate, isolating and identifying characters. License Plate is detected through video frames taken by the system by taking account of the plate's characteristics. Once done, the characters are isolated and segmented by Matlab. Last, Classification of standard Chinese license plate through template matching and neural net method. Based on statistical results, the system has accuracy of 97% (daytime) and 96% (night time). Thus, it claims that it works efficient under low-light conditions.

Another study is Smart License Plate Recognition system by V. Koval et al. in 2003 that can be used in tollbooth for recognizing vehicle's plate numbers and could be integrated to control payment in parking areas, bridges and etc. Their algorithm consists of capturing car images, deblurring frames, extracting and identifying license plate. It is similar to study stated above, however, their system includes image fusion—an enhancement through combination of different images before doing the plate extraction and recognition. It aims to address the problem present to the commercially available LPR where it cannot be readily used when vehicle image is at different style or angle. The results showed that the system can correctly identify license plate using artificial neural network with 95% accuracy.

The research study entitled License Plate Detection using Harris Corner and Character Segmentation by Integrated Approach from an Image by T. Panchal et al. in 2016 described a LPR system that can manage distinctive light conditions, distortion and obstructions. This study proposed a system with integrated segmentation approach which is suitable for moving vehicles and lighting conditions. The image obtained through camera undergo pre-processing technique. After that, Harris Corner algorithm is applied to detect the corners of the plate and segmenting the characters from the detected plate number. Several controlling parameters like Height, AR and LP of the plate region are considered to avoid false detection. Experiments shows that it has an overall accuracy of 93.84%

There are related studies but very few that is done here in the Philippines. An LPR system has already been developed in De La Salle University using fuzzy techniques and statistical method by Cue et al. in 1998 with approximately 87% accuracy. The system has significant flaw and inaccuracies due to blurred images, images outside the proximity range and those plates not having the Philippine License Plate Format. Also, it has no pre-processing techniques that can adapt to environment or lighting conditions.

Similarly, Automated License Plate Identifier (ALDEN) was developed to recognize images of Philippine license plates. The approach of this study includes pre-processing techniques to enhance the raw images and Perspective Distortion Correction. It captures vehicle images at a sufficient distance which undergo to image enhancement before segmenting distinct characters of the plate. The system has 95.33% accuracy for older plate format while 91.11% for the newer one. However the system fails to recognize the plate number in several conditions like extreme darkness, blurred images and partial occlusions. (Villanueva, 2010)

Also, Plate Recognition System for beating the red light by Rikki Robles in 2010 was proposed to expand the Metro Manila Development Authority's manual apprehension system. It can electronically identify violators of the said traffic policy. It also ensures minimal contact between traffic enforcers and motorist. The system includes three major step –image acquisition, segmentation and identification through different neural networks. Distinctively, this study incorporates 10 plate number candidates before localizing the correct plate number. The system has proven to recognize plate numbers, however the images must be a large-scaled one and has good lighting to have an accurate recognition.

Table 2. Summary of Related Literature on Existing License Plate Recognition System

Author	Year	Title	Algorithm	Remarks
Hui Wu et al.	2011	License Plate Recognition (LPR) System	1. selecting optimal frame 2. extracting license plate 3. isolating and; 4. identifying characters	This study focused on improving the accuracy and reliability of existing LPR in China
V. Koval et al.	2003	Smart License Plate Recognition system	1. to capture the car's images 2. to deblur of image frames 3. to extract image of license plate 4. to extract characters from license plate image 5. To recognize license plate characters and identify the vehicle.	Their system includes image fusion to enhance the quality of the images and increase the accuracy of detection.
Tejendra Panchal et. al	2016	License Plate Detection using Harris Corner and Character Segmentation by Integrated Approach from an Image	1. Image acquisition 2. Extraction Using Harris Corner 3. Segmentation 4. Identification of Plate Number	The study includes pre-processing technique and an integrated plate extraction. It can avoid false detection of non-character regions.

Table 3. Continuation on Summary of Related Literature on Existing License Plate Recognition System

Author	Year	Title	Algorithm	Remarks
Cue et. Al	1998	LPR system	1. fuzzy techniques in thresholding and; 2. Statistical methods for the character recognition.	Inaccuracies due to blurred images, outside the set distance and obstructions
Myra Josephine S. Villanueva		Automated License Plate Recognition of Philippine License Plates	1. Image Acquisition 2. License Plate Detection 3. Perspective Distortion Correction 4. Character Segmentation 5. Character Identification	It has pre-processing technique and perspective distortion correction for images taken in different angle.
Rikki Ruperto N. Robles	2010	Plate Number Recognition System For Beating The Red Light Violators	1. Plate Number Localization 2. Character Segmentation 3. Plate Number Identification	This LPR is capable of identifying violators electronically. It uses 10 images for candidates before detecting plate number.

2.1.3 Image Processing

Image processing is the manipulation of digitized images, to improve its quality. Image processing can also be used to detect a car license plate. The License Plate Recognition (LPR) involves three stages: plate detection, character segmentation and character recognition. These stages are composed of image processing techniques except the character recognition stage which involves artificial neural network. Some studies used different combination of image processing techniques to every stage.

The preprocessing stage is where the image is manipulated for preparation in the plate detection. Most of the studies including the study, Automated Number Plate Recognition System Using Machine learning algorithms, by Bhardwaj and Gujral in 2014, uses Grayscale Conversion and Noise Removal with the use of Median filter in the preprocessing stage. In the study of Dalida et al. in 2016, entitled Development of Intelligent Transportation System for Philippine License Plate Recognition, removed the shadow with the use of improved Bernsen algorithm.

1.1.3.1 Plate Detection

For the first stage, these LPR studies used different kind of image processing techniques that can be used to detect a car's license plate.

The study, Vehicle License Plate Recognition Using Morphology and Neural Network, by Patel in 2013, used opening and closing mathematical morphology, and edge detection with the use of Sobel operator to make the car license plate stand out. This allows the projection of binary difference image horizontally and vertically, smooth these two projection curves, and search their peaks to find the accurate rectangle that includes the plate.

The article, Number Plate Recognition Using an Improved Segmentation by Sutar and Shah in 2014, proposed two methods of license plate extraction. This is either by estimation of its location or by image segmentation by using image binarization method. The study also mentioned that the researchers used skew correction and detection for the proper positioning of the license plate.

In the study entitled Automatic License Plate Recognition System using SURF Features and RBF Neural Network by Chaturvedi and Sethi in 2013, mentioned the use of SURF for plate detection. SURF method is used for plate extraction based on the determinant of the hessian matrix.

The study of Lukic, Milan and Jovanovic in 2016, entitled Two-stage Algorithm for License Plate Extraction, proposed two-stage algorithm for plate extraction. The first stage involves dynamically adjustable thresholding for determining potential license plate areas. The second stage uses contrast information, entropy and vertical edge density for license plate extraction from other candidate regions.

In the article, Segmentation-and-Annotation-Free License Plate Recognition with Deep Localization and Failure Identification proposed by Bulan, Kozitsky, Ramesh and Shreve in 2017, the researchers used a two-stage approach that first extracts a set of candidate regions using a weak sparse network of winnows classifier and then filters them using a strong convolutional neural network (CNN) classifier in the second stage.

A study of Dalida et al. in 2016, Development of Intelligent Transportation System for Philippine License Plate Recognition, used Connected Component Analysis for labelling regions and preserving characters by removing unwanted information.

2.1.3.2 Character Segmentation

The next stage is about the partitioning of a digital image into multiple sets of pixels. Numerous studies used different kinds of methods in character segmentation.

In the study, Vehicle License Plate Recognition Using Morphology and Neural Network, by Patel in 2013, the common method used for character segmentation is the edge detection. In which the edge of each character will be the beginning and end of the cut.

The article, Number Plate Recognition Using an Improved Segmentation by Sutar and Shah in 2014, the researchers first get the output of the extracted number plate using labelling components, and then each and every character will be separated in the number plate image by using split and also find the length of the number plate, after that find the database and correlation if both the value is same means it will generate the value 0-9 and A - Z, and display it.

In the study made by Chaturvedi and Sethi in 2013, entitled Automatic License Plate Recognition System using SURF Features and

RBF Neural Network, morphological operations are used such as open, close and dilation. Then the components that has a lower threshold value are removed and the extracted characters will be fed on to the ANN.

The study, Block-Based Neural Network for Automatic Number Plate Recognition of Sagar and Dutta in 2014, the character segmentation is done by extracting each components in the binary plate region. In this study character segmentation involves Connected Component Analysis, Center Line Rule and Blob Extraction.

In the article proposed by Bulan, Kozitsky, Ramesh and Shreve in 2017, entitled Segmentation-and-Annotation-Free License Plate Recognition with Deep Localization and Failure Identification, segmentation and optical character recognition (OCR) are done by the use of probabilistic interference method which is based on hidden Markov models (HMMs) where the Veterbi algorithm is applied to determine the most likely code.

A study of Dalida et al. in 2016, Development of Intelligent Transportation System for Philippine License Plate Recognition, Hough transform is used in character segmentation for the tilt correction of the plate. From this, horizontal and vertical projections are used to determine the boundaries of the character that will be used for segmentation.

There are many different methods that can be used in License Plate Recognition. The Table 2.3.1 shows the summary of different image processing techniques used in the studies mentioned above.

Table 4. Summary of Image Processing Techniques

Title	Author	Image Preprocessing	Plate Detection / Extraction	Character Segmentation
Automated Number Plate Recognition System Using Machine learning algorithms	Bhardwaj and Gujral	Grayscale Conversion and Noise Removal	Edge Detection (Sobel)	
Vehicle License Plate Recognition Using Morphology and Neural Network	Patel		opening and closing mathematical morphology	Edge Detection
Number Plate Recognition Using an Improved Segmentation	Sutar and Shah		estimation of its location or image binarization method with skew correction	Filter and Crop(Edge Detection)
Automatic License Plate Recognition System using SURF Features and RBF Neural Network	Chaturvedi and Sethi		SURF	Open, close and dilation morphology

Table 4. Continuation on Summary of Image Processing Techniques

Title	Author	Image Preprocessing	Plate Detection / Extraction	Character Segmentation
Two-stage Algorithm for License Plate Extraction	Lukic, Milan and Jovanovic		1^{st} stage: adjustable thresholding 2^{nd} stage: contrast information, entropy and vertical edge density	-
Segmentation-and-Annotation-Free License Plate Recognition with Deep Localization and Failure Identification	Bulant et al.		winnows classifier with CNN	OCR
Development of Intelligent Transportation System for Philippine License Plate Recognition	Dalida et al.	Grayscale Conversion and Noise Removal with Skew Correction	Connected Component Analysis	Hough transform and horizontal and vertical projections

2.1.4 Artificial Neural Network

Artificial neural network or ANN is a mathematical paradigm based on the anatomical and functional abstraction of biological neural network. It is composed of reticulated group of nodes called artificial neurons or neurons that processes information applying network-based complex mathematical computations. A typical ANN structure consists of different parts: input, hidden, and output layers. In most applications, ANN alters its composition during its learning stage based on the input, target, and output information.

Artificial neural network can be used in character recognition. Any ANN algorithms can be applied to the system in order to achieve accurate and precise results. Related studies have shown significant outcomes after applying different ANN algorithms in their systems.

A study, An Approach to Recognize Characters using Neural Network in LPR System, by Singh et al. in 2014, focused on the use of Feed-Forward Neural Network for identifying and recognizing Indian license plate characters. Log-Sigmoid transfer function was used in the neural network. The system has inputs to the neural network, which consist of individual character images from the segmentation process. The output of the neural network is a computer-readable form of the image. The study compared different results using different number of nodes in the hidden layer: 200, 150, 100, 75, and 50 nodes. High recognition accuracy of 93% was achieved after using 200 nodes, while 72% was attained after utilizing 50 nodes. The study concluded that the use of feed-forward neural network, with increased number of hidden nodes, is sufficient to achieve high recognition accuracy.

Another study, Vehicle License Plate Recognition Using Artificial Neural Networks of Öz et al. in 2013, used Back Propagation algorithm, which is a feed-forward neural network capable of adjusting weights and biases based on the back-propagated error values related to the target values. The system was designed for identifying Turkish license plate characters consisting of 4 letters and 4 numbers, respectively. A training data of 36 characters was used. The neural network consists of 209 nodes in the input layer, 36 nodes in the hidden layer, and 36 nodes in the output layer. The training process of the neural network took 20000 iterations, with a learning rate and momentum rate of 20% and 80%, respectively. An error of 7.63% was obtained after the learning process. A significant recognition accuracy of 100% was achieved.

A New License Plate Recognition System Based on Probabilistic Neural Networks by Öztürk et al. in 2012 focused on the utilization of Probabilistic Neural Network (PNN) for recognizing Turkish license plate characters. PNN is a feed-forward neural network derived from the Bayesian network, which consists of 4 layers: input, hidden, pattern or summation, and output layers, capable of generating accurate predicted target probability scores. Extracted characters from image processing are transferred to the input layer of the PNN. A recognition accuracy of 96.5% was achieved after testing 1914 individual characters.

Number Plate Detection with a Multi-Convolutional Neural Network Approach with Optical Character Recognition for Mobile Devices by Gerber et al. in 2016 utilized Convolutional Neural Network (CNN) for identifying vehicle and for detecting plate. CNN, with the help of Optical Character Recognition (OCR), was also used for recognizing license plate characters. CNN is a neural network explicitly applied to images, wherein the layers have neurons arranged in three dimensions: height, depth, and width. The system obtained a maximum success rate of 96% with maximum false positives of 84.

Another study, Two Neural Networks for License Number Plate Recognition by Akoum et al., compared Multi-Layer Perceptron (MLP) and Hopfield neural networks performances in the recognition of Lebanese license plate characters. Approximately 350 sample license plates were used with a total of 1130 symbols. Different Hopfield (HOP112, HOP552, and HOP1008) and MLP (MLP112, MLP252, and MLP1008) architectures were considered. The highest recognition performance of 87% was obtained by HOP1008, which consists of 1008 neurons, and the lowest performance of 64% was acquired by MLP1008, also consists of 1008 neurons. A maximum performance of 80% for MLP was obtained by MLP252, while a minimum performance of 69% for HOP was acquired by HOP112.

A study about LPR in the Philippines also applied ANN algorithm for recognizing license plate characters. Vehicle Parking Inventory System Utilizing Image Recognition through Artificial Neural Networks by Bartolome et al. used back-propagation algorithm consisting of 50 input neurons. The network is trained using 5860 data sets resulting in a 0.0001645724% error. A high character recognition accuracy of 99.98% was achieved from 200 samples of license plates.

Different neural network algorithms were used in recognizing license plate characters. Table 2.4.1 shows the summary of the performances of different ANN algorithms for identifying license plate characters from different related studies. It clearly shows that the back-propagation neural network algorithm has the highest character recognition performance. Back-propagation algorithm best suits in certain applications like license plate recognition, where continuous unsupervised learning is required. Multi-layer perceptron algorithm acquired the lowest performance score. These performances are still dependent on the results of image processing.

Table 5. Summary of Performances of Different ANN Algorithms

For Character Recognition

Title	**Author**	**Artificial Neural Network Algorithm**	**Character Recognition Performance**
An Approach to Recognize Characters using Neural Network in LPR System	Singh et al.	Feed-Forward	93%
Vehicle License Plate Recognition Using Artificial Neural Networks	Öz et al.	Back-Propagation	100%
A New License Plate Recognition System Based on Probabilistic Neural Networks	Öztürk et al.	Probabilistic	96.5%
Number Plate Detection with a Multi-Convolutional Neural Network Approach with Optical Character Recognition for Mobile Devices	Gerber et al.	Convolutional	96%
Two Neural Networks for License Number Plate Recognition	Akoum et al.	Hopfield	87% (Maximum)
		Multi-Layer Perceptron	80% (Maximum)
Vehicle Parking Inventory System Utilizing Image Recognition through Artificial Neural Networks	Bartolome et al.	Back-Propagation	99.98%

2.2 Related Patents

License Plate Recognition is widely used in several application and system particularly for law enforcement and parking control system. The following are related patents that implement an LPR technology:

A patent entitled *Portable Covert License Plate Reader* by Mark Edward Windover, Bernard D. Howe in 2009 with a publication number of US7504965 B1 is a surveillance system for monitoring vehicle license plates. This system can identify and read license plate of numerous moving vehicles that cross the field view of the camera. It comprises two units: the license plate reader which uses (1)infrared camera, (2) image acquisition and processing device, (3) an illuminator, (4)lighting device and (5)a wireless transmitter; and mobile surveillance unit which uses (6)wireless receiver, (7)monitor, and (8)on board processing unit.

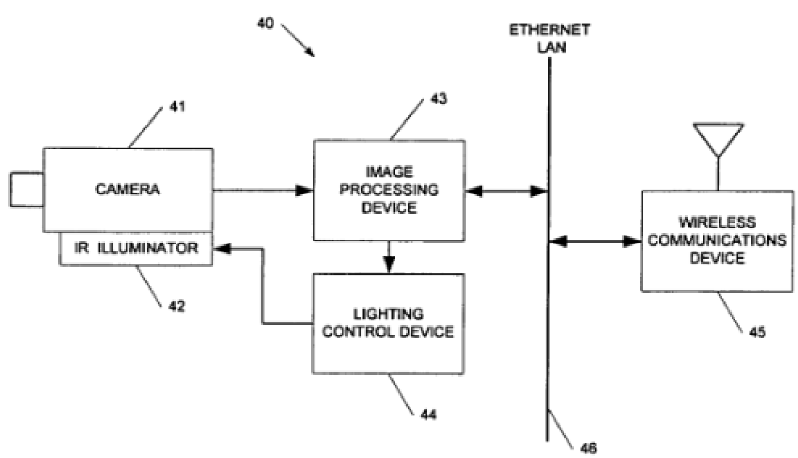


Fig. 2.1 System Architecture of the Covert Portable License Plate Reader

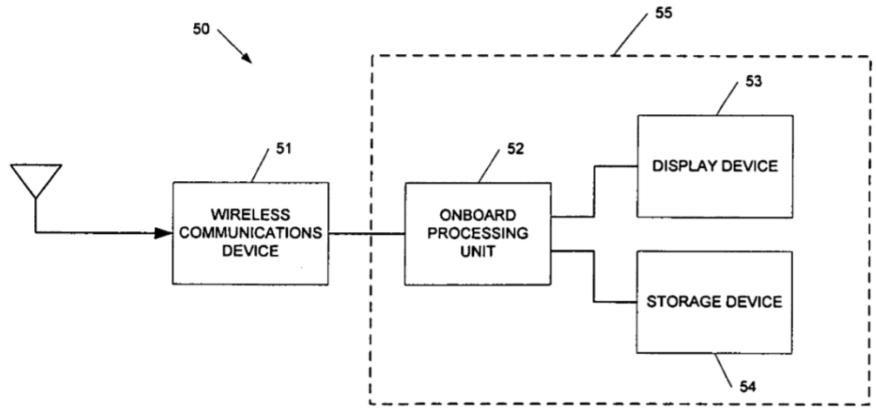


Fig. 2.2 System Architecture of the Mobile Surveillance System

Automated License Plate Locator and Reader Including Perspective Distortion is a US patent with publication number of US005651075A in 1997 by James F. Frazier, Craig C. Reinhart and James F. Alves. This patent focuses more on methods relating to (1) the correction of perspective distortion through warping the images of license plate, (2) the location of license plate which contains edge enhancement and shadow reduction by executing Laplacian operator to the input image, (3) the identification of license plate which contains a modified binary correlation of image with reference characters to provide a necessary output, and (4) the increase rate of its overall performance for producing an output. The processes disclosed herein will provide an additional development and modification on its certain application.

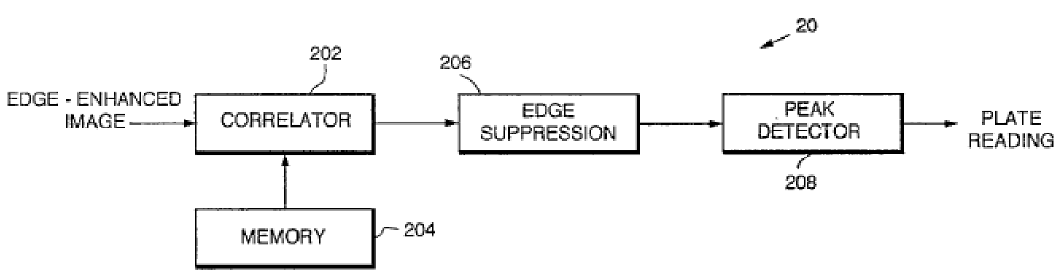


Fig. 2. 3 Block Diagram of the system

The inventors, Claus Neubauer and Jenn-Kwai Tyan, have their copyright to *License Plate Recognition with an Intelligent Camera* in 2003 with the publication number of US006553131B1. This patent discloses a camera that captures and recognizes license plate which contains program device such as detection of orientation, position, illumination condition and blurring of an image. The system also includes segmenting program for identification of character's position and statistical classifier for recognition of each character. Moreover, installation of memory is essential for its database that will store license plates, records, programmed codes and accuracy of its recognition.

A US patent entitled *Automated License Plate Recognition System for Use in Law Enforcement Vehicles* in 2006 by William D. Rau and Nathan M. Rau with a publication number of US6982654B2 is essentially designed for stolen cars. This system is directly installed to a law enforcement vehicle. This includes: (1) A camera which can be mounted on various parts of the enforcement vehicle for identifying license plate. This can work either the vehicle is in motion or in stillness. Then, it produces a signal as a digital image of the license plate. (2) A processor now receives the signal from the camera and it is where an LPR takes place. The processor interprets and recognize the characters of the license plate which will be compared to the database of stolen cars stored in a (3) hard drive. Other feature of the said scheme is the identification of the car's model, type or color. Afterwards, a signal will be delivered to an (4) output device that matches the target plate and stolen plate from the database. This displays information concerning the recognized license plate. Also, it indicates if the plate corresponds to any information of database.

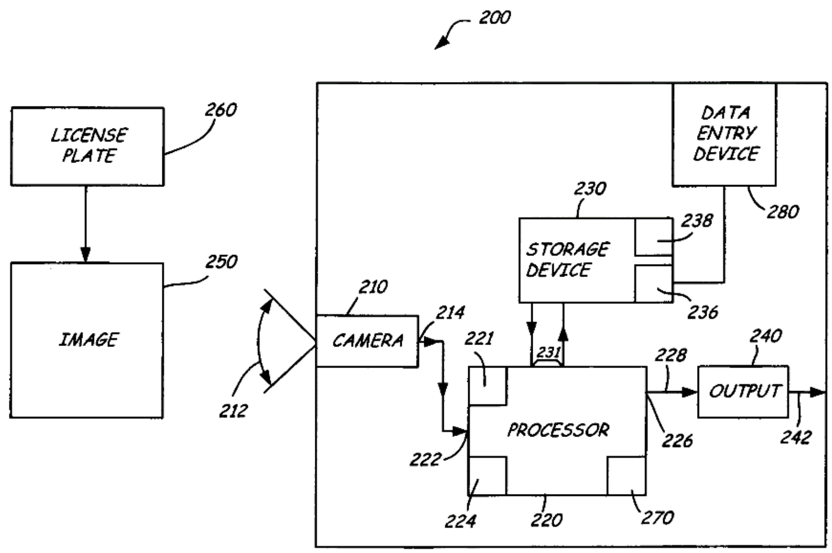


Fig. 2. 4 Block Diagram of the Components

From China, a patent entitled *Automobile License Plate Automatic Recognition Method and Implementing Device* by Xu Huiying, Zhu Xinzhong, Hu Chengyi, Zhao Jianmin with a publication number of CN101398894A is appropriate for automatic identifying method and realizing device for identifying and dividing the licenses which do not affect by the shoot traffic video image quality under a complex background with good identifying effects to the characters. The following are the steps included in the identifying technique: (1) tracing a movement multi-target body area; (2) capturing candidate license area image; (3) capturing a real license edge point; (4) capturing the smallest license area; (5) dividing characters and removing non-character images; (6) normalizing the character images and extracting the original coarse gridding characteristics of the character images; (7) using a trained BP neutral network for realizing the accurate identification of the license characters.

Table 6. Comparison of Different Patents

	Patent 1	**Patent 2**	**Patent 3**	**Patent 4**	**Patent 5**
Use of:					
• **Neural Network**					
• **Illumination**					
• **Database**					
For Day and Night					
For Moving vehicles					
Resolves Distortion					

Table 2.5 shows the comparison of all the related patents regarding the use of neural network, illumination and database. In the table shown above, Patent 1 entitled as *Portable Covert License Plate Reader* is the only patent that can operate both day and night. Also, this table displays the better patent for moving vehicles which is the patent 1 and 4. Additionally, patent 2 entitled *Automated License Plate Locator and Reader Including Perspective Distortion* focuses for resolving distortion.

Chapter 3

METHODOLOGY

This section of the study contains a detailed explanation of the methodology. It's consist of project development, software and hardware design, project components—materials and equipment used, implementation and testing procedure.

3.1 Research Development

3.1.1 Conceptual Framework

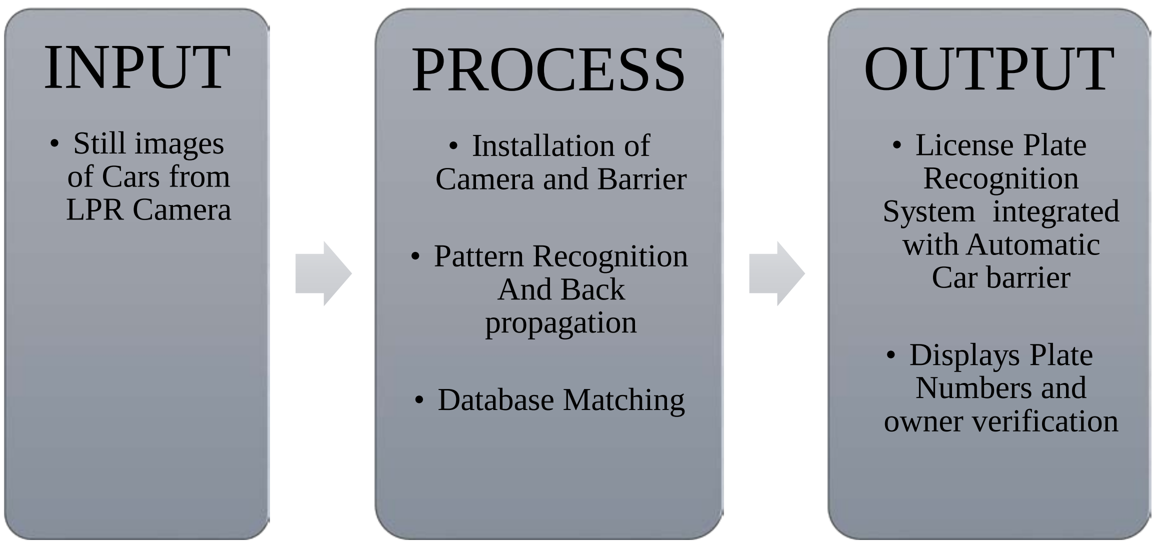


Figure 1. Conceptual Framework

The Conceptual framework shown in Figure 1 includes the important parameters—a.) Input b.) Process and c.) Output of the entire methodology. The main idea of the study is to develop a system which allows vehicles, specifically cars to enter the university through license plate recognition in conjunction with the automatic car barrier. It is to impose added security for the campus.

The input of the project is the captured images of the vehicles which will be subjected to the different algorithms such as pattern recognition and back propagation. The system will verify whether the vehicles are authorized or not before allowing it to enter the university. The final output is a system that are capable of plate number recognition and owner verification.

3.1.2 Research Design

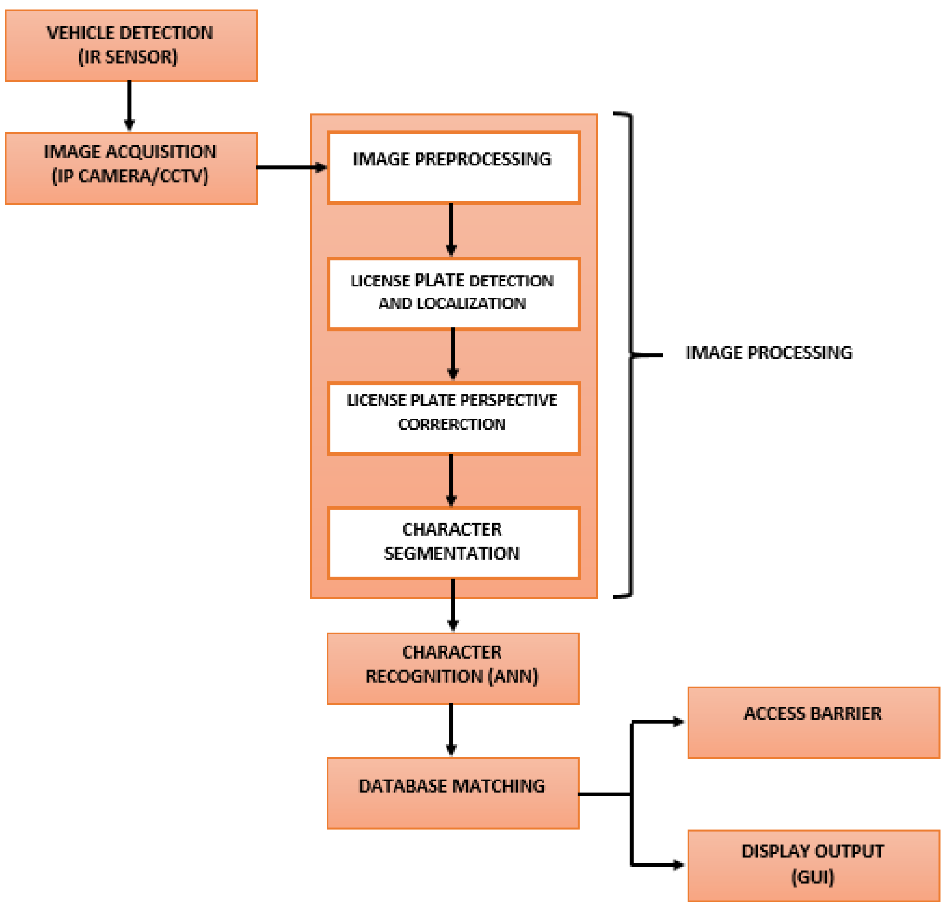


Figure 2. System's Block Diagram

The block diagram of the study is shown in Figure 2. The LPR camera is triggered to grab a frame in the video through its built-in motion detector. The captured image undergoes a pre-processing technique to enhance its overall quality before localization of plate region occurs. This region is then segmented into different sub-regions that contains characters. Pattern Recognition and Back-propagation are the Artificial Neural Network techniques used to distinguish the characters of the actual license plate. Once done, it is verified in the database's whitelist before allowing the vehicle to pass beyond the car barrier. Moreover, recognized license plate is also displayed in the monitor.

3.1.3 Research Process Flow

The flowchart presented below is followed for the attainment and achievement of the objectives.

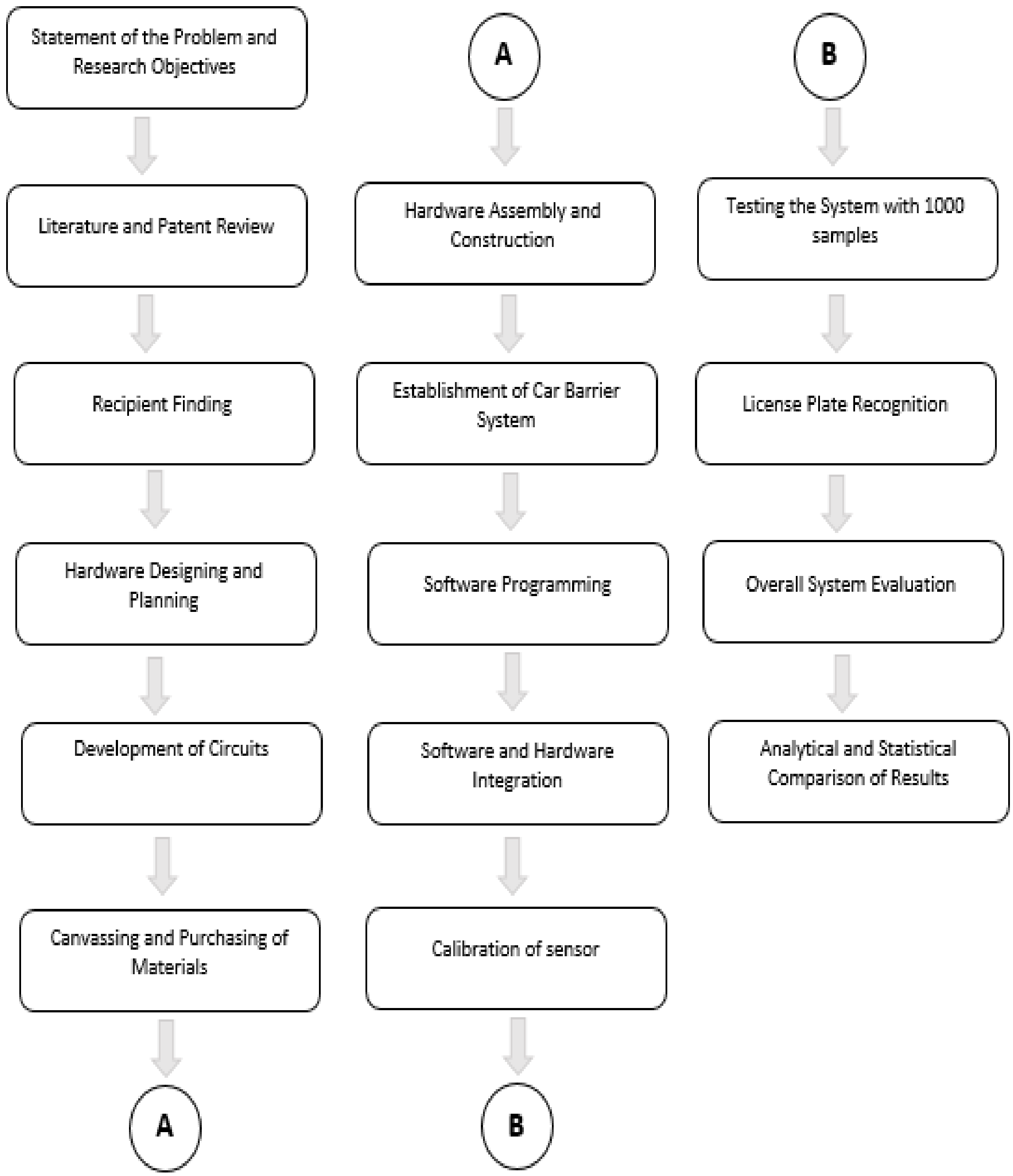


Figure 3. Research Flow Process

3.2 Hardware Design of the License Plate Recognition System

This section describes the physical components of the project. The system's wiring, security barrier used, as well as LPR module—camera, monitor and memory storage are taken into account in the development of License Plate Recognition System.

3.2.1 Project Components

The following or other similar products will be utilized for the system's design and structure:

Camera

This study uses a weather proof 4MP Full HD WDR Network Small IR Bullet Camera which is continuously operating day and night. The used of CCTV with IR illumination is to compensate following issues such as moving vehicles, headlights and low-light conditions.

Figure 4. Dahua DH-IPC-HFW4421SN IP Camera
(https://www.bhphotovideo.com/c/product/1189777REG/dahua_technology_dh_ipc_hfw4421sn_ipc_hfw4421sn_4mp_full_hd.html)

Power over Ethernet (PoE) Switch

PoE switch is a device that allows network cables to transmit electrical power and data at the same time. The project uses TEF1105P-4-63W, a 5-port 10/100Mbps where the 4 ports are PoE. This can automatically detect and supply power to all compatible powered devices. The benefits of using a PoE switch are; cost effective, time saving, flexible and reliable. IP cameras are preferably PoE supported for fast deployment and easy positioning.

Figure 5. TendaTEF1105P-4 63W PoE Switch

(https://homesecurity1st.co.za/product/tenda-teg1009p-9-port-gigabit-desktop-switch-with-4-port-poe/)

Arduino Uno

The Arduino Uno is the main controller of the car barrier system. It is powered through its USB Port (5V). A program is written into it to enable data acquisition, processing and data transfer via serial communication. It receives and analyze the data input of sensors for prompting the camera as well as the input of database which will activate the barrier to function.

Figure 6. Arduino Uno

(https://www.sparkfun.com/products/11021)

Monitor

The monitor will be used to display the information in the system's GUI such as the process of recognizing the plate, plate number, authorized person based on the database's record. It is also serves as the storage for database and recorded feeds of CCTV.

Figure 7. Computer System

(https://tweakers.net/pricewatch/231709/acer-emachines-el1200/specificaties/)

Car Barrier

A customized boom barrier or fence to block an incoming vehicle for inspection before allowing through the facilities. It will have a typical design that rises in vertical arc manner once authorization was given.

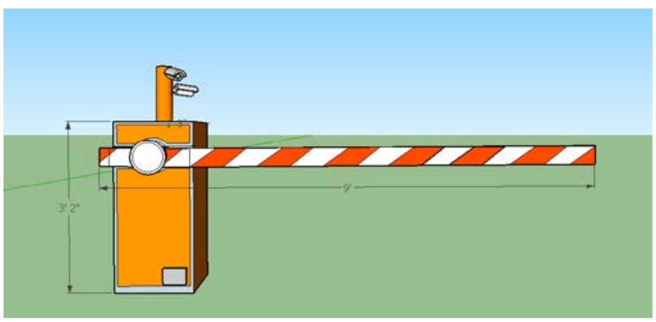


Figure 8. Customized Car Barrier

Infrared (IR) Sensor

Infrared sensors can detection motions or sense its surroundings through infrared radiation. This study utilizes these sensors for prompting the camera to capture the vehicles' license plate and for identifying the direction of the vehicles.

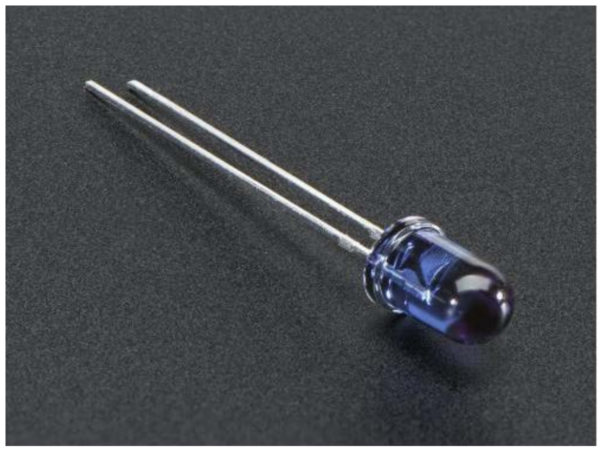


Figure 9. IR Led Super-Bright 5mm 940nm

(https://circuit.rocks/ir-led-super-bright-5mm-940nm.html)

3.3 Software Development

3.3.1 Programming Language

The programming language used in this study is MATLAB, which is a multi-paradigm high level language specifically designed for mathematical and engineering applications. It perfectly fits for license plate detection and character recognition that utilizes image processing and artificial neural network.

3.3.2 Software Structure

3.3.2.1 Image Acquisition

This is the first stage of an LPR system where the image to be processed was taken with the use of a LPR. The frame was extracted from the video by the use of Image Acquisition Tool in MATLAB.

3.3.2.2 Image Pre-processing

In this stage, the image undergoes different filter operations for faster and easier evaluation. The pre-processing stage is composed of different image processing techniques that are essential in image analysis. The first technique is RGB to Grayscale conversion. In this technique the pixels of the image will be converted to different shades of gray. Then the image intensity should be adjusted to adjust the amount of light in each pixel.

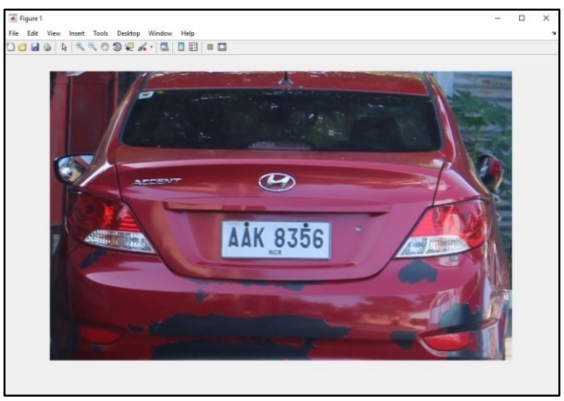


Figure 10. Original Image

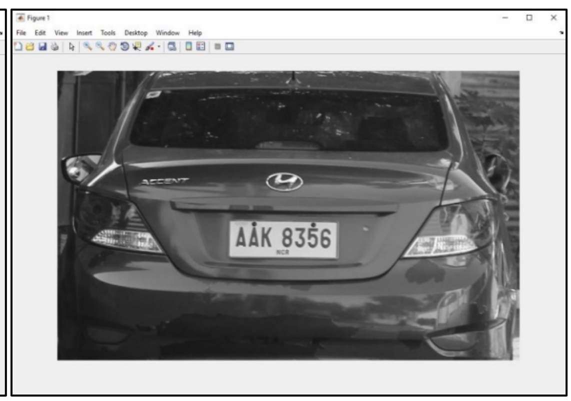


Figure 11. Image converted to Grayscale

3.3.2.3 License Plate Detection

There are different methods of locating the license plate region but the most effective of them all is the study of Farhad Faradji, Amir Hossein Rezaie and Majid Ziaratban entitled *A Morphological-Based License Plate Location* in 2007. The study indicates that the detection rate is 83.50%, where 400 sample images were used and were taken from different scenes like diverse angles and different lighting conditions. Just like what the study indicated, the image will undergo different stages for the detection of the region of the license plate. The first step is vertical edge detection, then histogram analysis is applied, after that is dilation both in horizontal and vertical directions, next is to find the common regions from the horizontal and vertical dilation, then erosion is applied, after that is to find the biggest blob, and lastly smoothen and enlarge the blob. It doesn't stop there, since the purpose of the previous processes is to reduce the area where the license plate is located.

Now that the region of the license plate is narrowed down, edge detection, vertical and horizontal dilation, finding the common region and image filling will be applied to the image to get the blob of the license plate itself. Then the largest blob will be selected, so that all unnecessary white pixels will be removed. After this, the blob will be used to crop the area of the license plate from the captured image.

After the extraction of the license plate, the license plate will undergo the process of perspective correction so that it will be easier for the ANN to recognize the characters.

3.3.2.4 Character Segmentation

In this stage, individual characters are detected in the license plate. The characters are divided into two, letters and numbers. Using horizontal and vertical projections, the unwanted areas are removed, the largest space in between the letters and numbers is detected and the characters are segmented. When the area of the black pixels fall within a specified range, it will be automatically stored in a variable that will be used for the next stage. Figure 12 shows the segmented characters from the license plate image.

Figure 12. Segmented Characters

3.3.2.5 Character Recognition

In this phase, the segmented character images are recognized. Alphabetical (A-Z), specifically uppercase letters, and numerical (0-9) characters are identified. Different techniques can be implemented in order to characterize and recognize the segmented images. In this study, artificial neural network (ANN) is utilized. In order to employ ANN, specific inputs and outputs must be obtained. That is, for the input of ANN, feature vectors of the segmented images are required, which are acquired through image feature extraction. For the output, logical vectors are utilized in order to classify each character.

3.2.2.5.1 Feature Extraction

Feature extraction is the phase in which the attributes of the character image are obtained, that can be used to distinguish individual character's identity, which then is utilized in ANN. For this study, the two-dimensional Gabor filter is used to extract the features.

The 2D Gabor filter is a band-pass linear filter used to analyze the spatial frequency of objects in the image through specified directions and orientations. It is widely used for texture analysis, which is convenient for character recognition applications. It can be represented mathematically by obtaining its impulse response function, which is a Gaussian-enveloped sinusoidal wave:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = e^{-\frac{x'^2+\gamma y'^2}{2\sigma^2}} e^{i\left(2\pi\frac{x'}{\lambda}+\psi\right)} \qquad \textbf{Equation 1}$$

Where:

$$x' = x\cos\theta + y\sin\theta \qquad \textbf{Equation 2}$$

$$y' = -x\sin\theta + y\ \cos\theta \qquad \textbf{Equation 3}$$

In the impulse response function, λ is the sinusoidal wavelength, θ is the orientation of the Gabor function stripes, ψ is the phase offset, σ is the standard deviation of the Gaussian envelope, and γ is the spatial aspect ratio. The impulse response function of 2D Gabor filter is a complex function that has an even-symmetric real component and an odd-symmetric imaginary component. In this study, only the even-symmetric real component is used. It is mathematically represented as:

$$g(x,y;\lambda,\theta,\psi,\sigma,\gamma) = e^{-\frac{x'^2+\gamma y'^2}{2\sigma^2}} \cos\left(2\pi \frac{x'}{\lambda} + \psi\right) \quad \textbf{Equation 4}$$

Through this equation, a Gabor filter bank can be obtained that will be used to extract the Gabor features. A specified Gabor filter bank of five scales and eight orientations is applied to obtain forty 39x39 matrices. Figure 13 illustrates the Gabor filter bank used in the system.

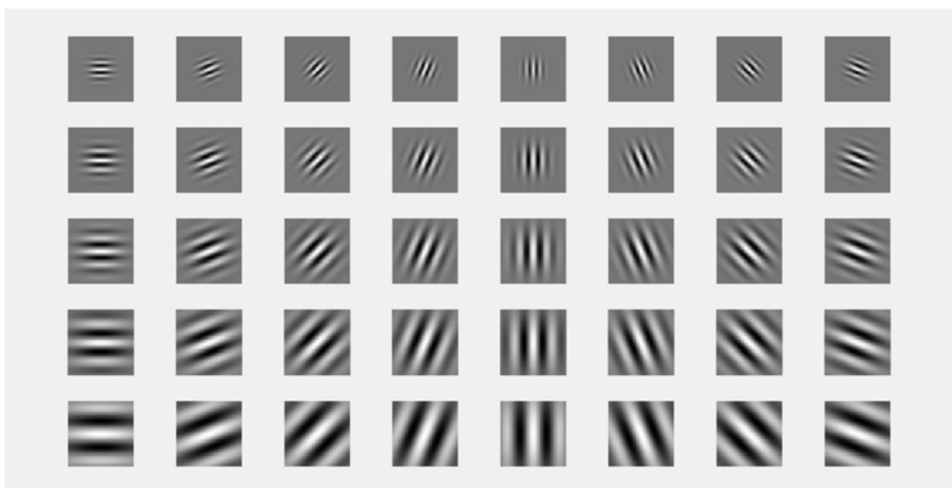


Figure 13. Gabor Filter Bank (five scales and eight orientations)

Each segmented binary image is complemented. Then it is resized to 50x50 pixels and passed through the Gabor filters to obtain the filtered images. Figure 3.14 shows the complemented segmented image of character “A” resized in 50x50 pixels. Figure 15 illustrates the filtered image of character “A” in Figure 14. It clearly shows that the characteristics of the

Gabor filter is relevant to the filtered images of the segmented character while maintaining the original features of the character “A”.

Figure 14. Complemented binary image of segmented character “A”.

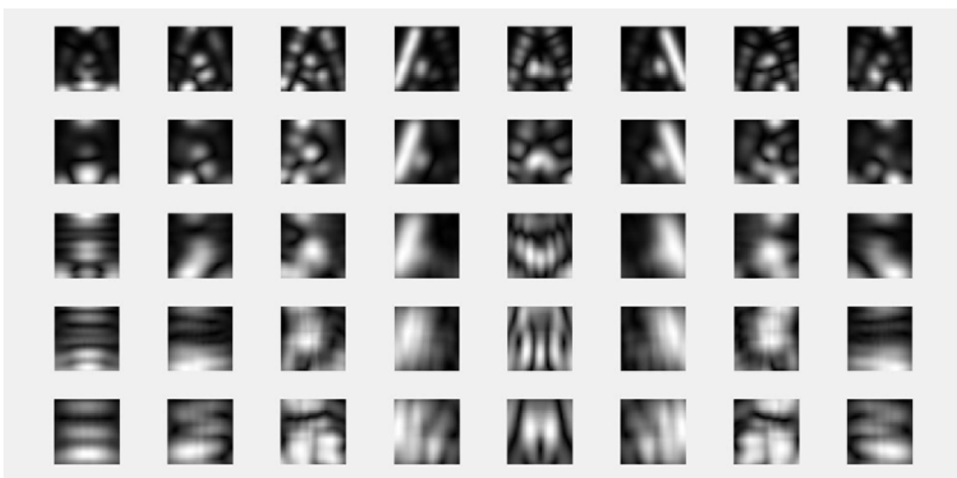


Figure 15. Gabor filtered images of segmented character “A”.

The characteristics and values of the filtered images are the bases for extracting the feature vector. The vector was down sampled in the order of five to minimize the data while still maintaining its features. Figure 16 shows the plotted feature vector of segmented character “A”.

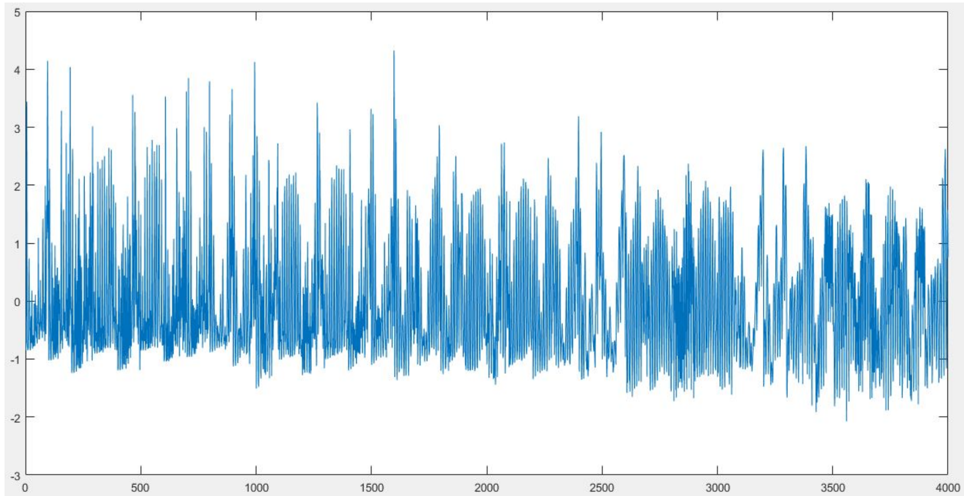


Figure 16. Feature vector of segmented character “A”.

The feature vector has a length of 4,000, that is, each segmented character image will have 4,000 inputs in ANN.

3.2.2.5.2 Artificial Neural Network

The Artificial Neural Network (ANN) is a computing system through mathematical connectionism, inspired by biological neural networks, that comprises connected and interrelated nodes called artificial neurons. These neurons are situated on three separate layers: input layer, hidden layer, and output layer. The input and output layers have specific number of artificial neurons based on the number of inputs and outputs of the ANN. The hidden layer on the other hand can be expanded to multiple layers and can have extensive number of artificial neurons. However, the amount of neurons in the hidden layer and the number of hidden layer itself significantly affect the performance of the ANN. Figure 17 shows the basic construction of ANN.

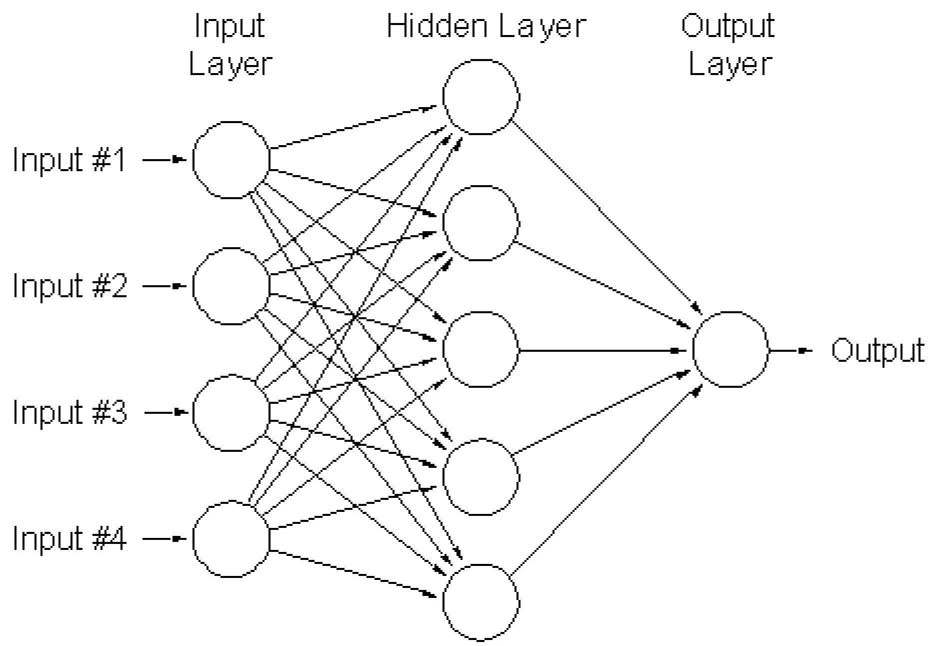


Figure 17. Basic Artificial Neural Network Structure

For this system, a two-layer feed-forward neural network is utilized, with a hidden layer comprising an activation function of hyperbolic tangent sigmoid, and an output layer composing of soft max transfer function. This network construction is appropriate for pattern recognition, which is applicable for system utilizing logical vector outputs.

Two separate networks are constructed for alphabetical and numerical recognition. The number of neurons in the hidden layer for each network were chosen that yield optimum performance in training and generalization. Logical vector outputs for alphabetical and numerical recognition are also separated. Figure 18 shows the logical vector outputs for each numerical recognition. Same pattern applies to vector outputs for alphabetical recognition.

0	[1	0	0	0	0	0	0	0	0	0]
1	[0	1	0	0	0	0	0	0	0	0]
2	[0	0	1	0	0	0	0	0	0	0]
3	[0	0	0	1	0	0	0	0	0	0]
4	[0	0	0	0	1	0	0	0	0	0]
5	[0	0	0	0	0	1	0	0	0	0]
6	[0	0	0	0	0	0	1	0	0	0]
7	[0	0	0	0	0	0	0	1	0	0]
8	[0	0	0	0	0	0	0	0	1	0]
9	[0	0	0	0	0	0	0	0	0	1]

Figure 18. Logical vector outputs for numerical recognition.

The neural network for alphabetical recognition has an input layer with 4000 neurons, generated due to 4,000 Gabor features, a hidden layer with 10 neurons, and an output layer with 26 neurons. Figure 19 shows the construction for alphabetical recognition neural network.

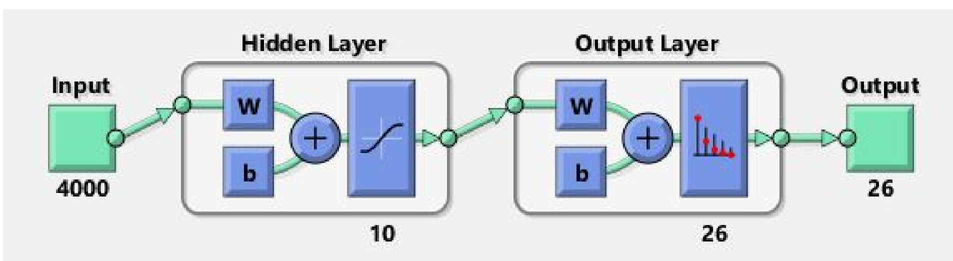


Figure 19. Neural Network Structure for Alphabetical Recognition

The network for numerical recognition on the other hand has an input layer with 4000 neurons, a hidden layer with 5 neurons, and an output layer with 10 neurons. Figure 20 shows the structure for numerical recognition neural network.

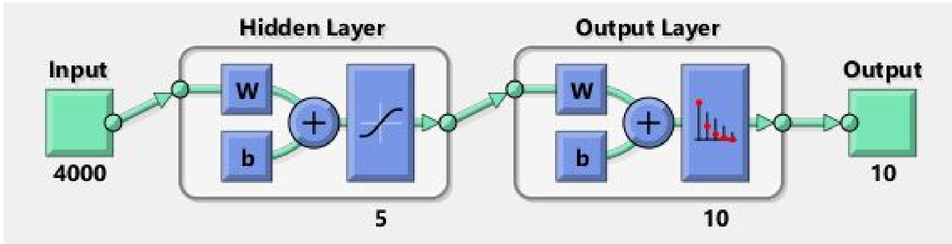


Figure 20. Neural Network Structure for Numerical Recognition

3.2.2.5.3 Neural Network Training

The neural networks were both trained using scaled conjugate gradient back-propagation algorithm, a supervised learning algorithm that updates the weights and biases of the network through scaled conjugate gradient method. This training method can be utilized if the weights, inputs, and activation functions have derivative functions. It is essential for training pattern recognition networks that use logical vectors as outputs.

The training initiates by randomly allocating the sample inputs to the network into 70-15-15 ratio, for training, validation, and testing respectively. The training samples are used for training the neural network having a specific target output for each sample. It is the main requirement for training the network that requires the highest partition among the input samples. The validation samples are utilized for regulating the training process by evaluating the cross-entropy error used for generalization. If the cross-entropy error increases in successive iterations, the training stops, giving the neural network a good generalization capability. The testing samples are used for testing the neural network after training, wherein, test errors are evaluated to measure the network performance.

The alphabetical recognition neural network was trained based on 520 sample images of letters used in the license plate, that is, 20 samples per letter each having 4,000 Gabor feature inputs to the network. The training phase of the network divides the sample images into 70-15-15 partitions, where 364, 78, and 78 samples were allocated randomly for training, validation, and testing, respectively. The numerical recognition neural network on the other hand was trained using 200 sample images of numbers, 20 samples per number each having 4,000 Gabor feature inputs. The samples were divided into 70-15-15 ratio, resulting to 140, 30, and 30 samples utilized for training, validation, and testing, respectively. Both neural networks obtained zero error in training, validation, and testing during the training phase.

3.3.2.6 Database

In this stage, the information from the character recognition stage will be collected and stored.

3.3.3 Flowchart

Figure 21 shows the software flowchart of the LPR system implemented in Matrix Laboratory (MatLab). The program will decide whether the image captured contains License plate. Once image satisfies the detection of plate, it will undergo character segmentation and recognition. This data will be displayed and checked against the pre-defined database. The 'matched' vehicles will be allowed to enter; otherwise it will be subjected to manual inspection.

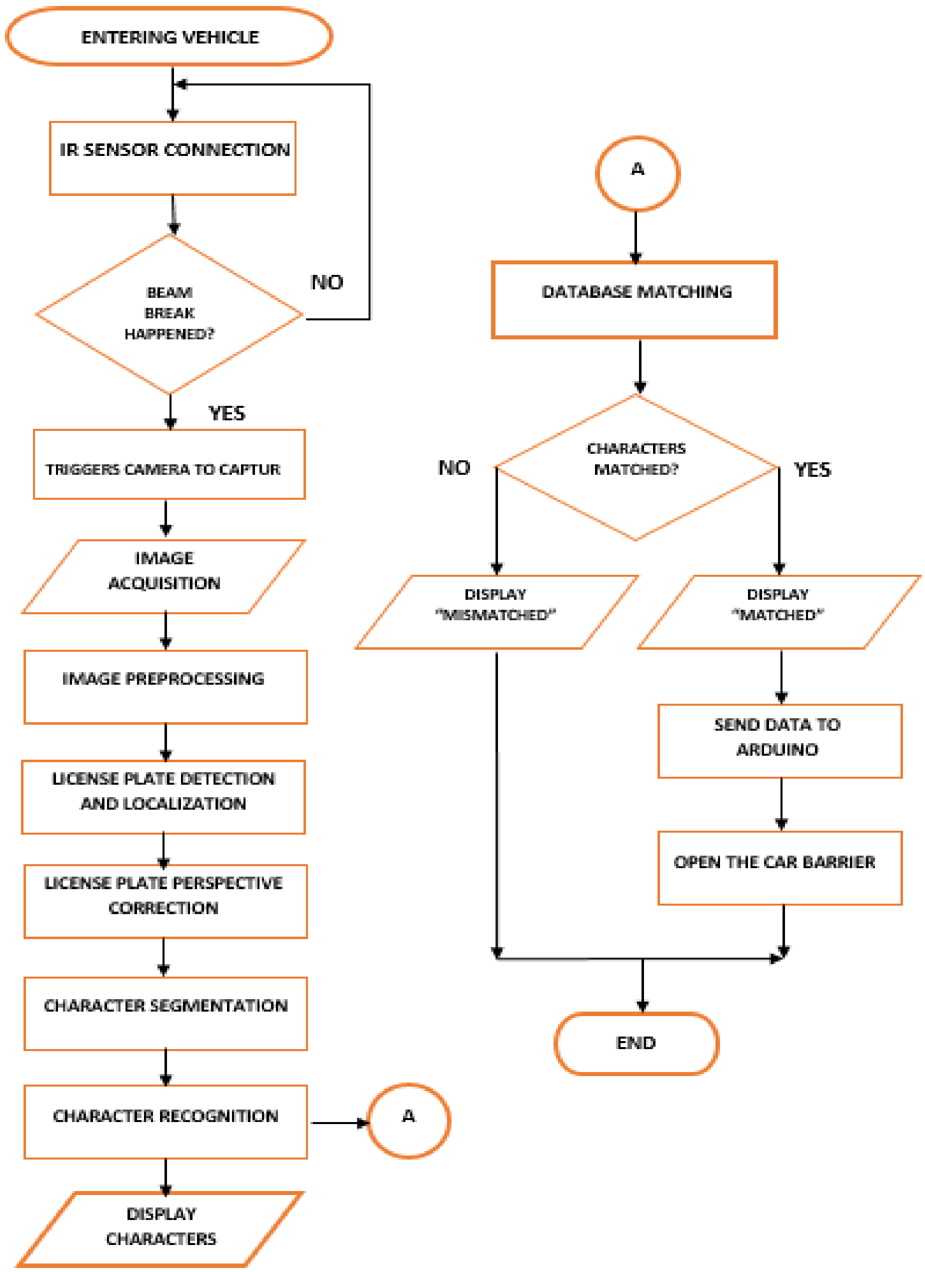


Figure 21. Software Flowchart

3.4 Operational and Testing Procedure

The accuracy of the software is the main factor to consider whether the system is competent enough to the commercially available License Plate Recognition System. The researcher's utilized basic neural network techniques while minimizing errors through optimal set up of system.

The proponents will be coordinating to the Management of Technological University of the Philippines – Manila (TUP-M) about the deployment of the project. The researchers are planning to test the recognition with 100 samples of plate numbers and acquire ninety percent (90%) confidence through License Plate Recognition (LPR) camera. Image acquisition through the camera's built in motion detection. The system will be evaluated during daytime and night-time to cover different lighting conditions. An additional lighting will be activated at night to lessen the inaccuracies. Also, testing will only be done under light to normal weather conditions to exclude the great effect heavy rain to the capability of the camera to capture high quality images. The result will be evaluated statistically under these aforementioned conditions.

3.5 Bill of Materials

I. Hardware

ITEMS	Quantity	Unit Price (Php)	Total Price (Php)
Acrylic Prototype	5	111.00	555.00
Acrylic Prototype	5	146.00	730.00
Tubular (for sensors)	1	790.00	790.00
		TOTAL	**2,075.00**

II. IR Transmitter

ITEMS	Quantity	Unit Price (Php)	Total Price (Php)
IC (LM 555)	8	30.00	240.00
Super Bright IR	4	85.00	340.00
IR (Generic)	4	15.00	60.00
Resistors	28	1.00	28.00
Transistor (BC 107)	4	10.00	40.00
Capacitor	16	6.00	96.00
Acrylic Board	1	120.00	120.00
PCB	1	210.00	210.00`
PCB	1	150.00	150.00
Ferric Chloride	1	55.00	55.00
Pin Header	1	40.00	40.00
Wires		207.00	207.00
Male-Male Wires		52.00	52.00
		TOTAL	**1638.00**

III. IR Receiver

ITEMS	Quantity	Unit Price (Php)	Total Price (Php)
Infrared Receiver	5	60.00	300.00
Infrared Receiver	5	5.00	25.00
LED	5	2.00	10.00
TSOP	3	65.00	195.00
VS1838	2	150.00	300.00
Resistor	10	1.00	10.00
Capacitor	5	1.00	5.00
		TOTAL	**520.00**

IV. Barrier Design and Controller

ITEMS	Quantity	Unit Price (Php)	Total Price (Php)
Power Supply (24 V DC)	1	1,000.00	1,000.00
Relay	1	420.00	420.00
Arduino UNO	1	1,300.00	1,300.00
Motor and Microswitch		2,600.00	2,600.00
Mild Steel (Plate)		900.00	900.00
Steel (Body)		1,400.00	1,400.00
Pulley	2	462.50	925.00
Shaft and Pillow Block		1,827.00	1,827.00
Nuts and Bolts		699.00	699.00
Springs		130.00	130.00
Belt		120.00	120.00
Paints		1,120.00	1,120.00
Aluminum		1,000.00	1,000.00
OTHERS		420.00	420.00
Labor			15,000
		TOTAL	**26,141.00**

V. Camera

ITEMS	Quantity	Unit Price (Php)	Total Price (Php)
CCTV	1	4,230.00	4,230.00
CCTV	1	2,250.00	3,200.00
DC Power Supply	1	230.00	230.00
POE-Switch	1	1,950.00	1,950.00
		TOTAL	**9,610.00**

VI. Installation

ITEMS	Quantity	Unit Price (Php)	Total Price (Php)
Plastic Molding ½”	5	35.00	175.00
Plastic Molding 1”	2	90.00	180.00
Plastic Molding 1”	2	149.75	299.50
1-meter Royal Cord #14	8	75.00	600.00
1-meter Speaker wire	34	9.00	306.00
Dyna bolt	8	20.00	160.00
Masonry bit (13mm)	1	139.75	139.75

Masonry bit (4mm)	3	69.75	209.25
Switch Adaptor	1	250.00	250.00
Nuts & bolts	4	34.75	139.00
Labor			950.00
		TOTAL	3408.50

VII. Prototype

ITEMS	Quantity	Unit Price (Php)	Total Price (Php)
Tubular	1	365.00	365.00
Angle Bar	1	228.00	228.00
GI Sheet	1	380.00	380.00
Motor (DC)	1	2,300.00	2,300.00
Spray Paint	2	100.00	200.00
Nuts and Bolts	16	6.00	96.00
		TOTAL	**3,569.00**

VIII. Total Expenses

ITEMS	**Total Price (Php)**
Hardware	2,075.00
IR Transmitter	1,638.00
IR Receiver	520.00
Barrier Controller	26,141.00
Camera	9,610.00
Installation	3,408.50
Prototype	3,569.00
TOTAL	**46,961.50**

3.6 Gantt Chart

Table 7. Gantt chart

ACTIVITIES	2016		2017												2018	
	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sept	Oct	Nov	Dec	Jan	Feb
Searching for Project Titles																
Composing Chapter 1-2																
Title Defense																
Revision of Chapter 1-2																
Consultation with Project Adviser																
Meeting with University's Management																
Composing Chapter 3																
Topic Defense																
Development of Software																
Canvassing of Materials																
Hardware Design and Development																
Progress Report																
Operational and Testing Procedure																
Evaluation of Results																
Pre-Final Defense																
Further Testing and Evaluation of the project																
Implementation of the Project																
Finalization of Documents																
Final Defense																

The Gantt chart displays in Table 7 is used as a graphical illustration of the proponent's project timeline. This tool aids the proponent for planning, organizing, systematizing and establishing the development and advancement of the project presented.

CHAPTER 4

RESULT AND DISCUSSION

This chapter contains the project technical description, structural description, project capabilities and limitations and tabulation of results and interpretation.

4.1 Project Technical Description

This project is about developing a system which aims to recognize license plates and once matched with those in the database, will control the car barrier to automatically allow the vehicle to pass. The system's design is composed of Closed-Circuit Television, Arduino Uno, Mat Lab, Infrared (IR) sensors, and customized car barrier.

Figure 22 shows the proposed setup of the project. There are four pairs of IR sensors, two for each side of the barrier since the system will operate for both entrance and exit. The Arduino Uno is the one responsible for controlling the sensors. Arduino Uno and IR receivers are supplied by 5 DC volts' supply while the IR transmitter is supplied by 9 DC volts. Sensors' arrangement determines the direction of the vehicle, if the vehicle blocked two sensors, it will prompt the camera to capture. Once there is a data extracted from the image through the LPR system, it will undergo database matching. When it is a match, a signal will be transmitted to the microcontroller to operate the car barrier. Otherwise, the user will opt to remote control function.

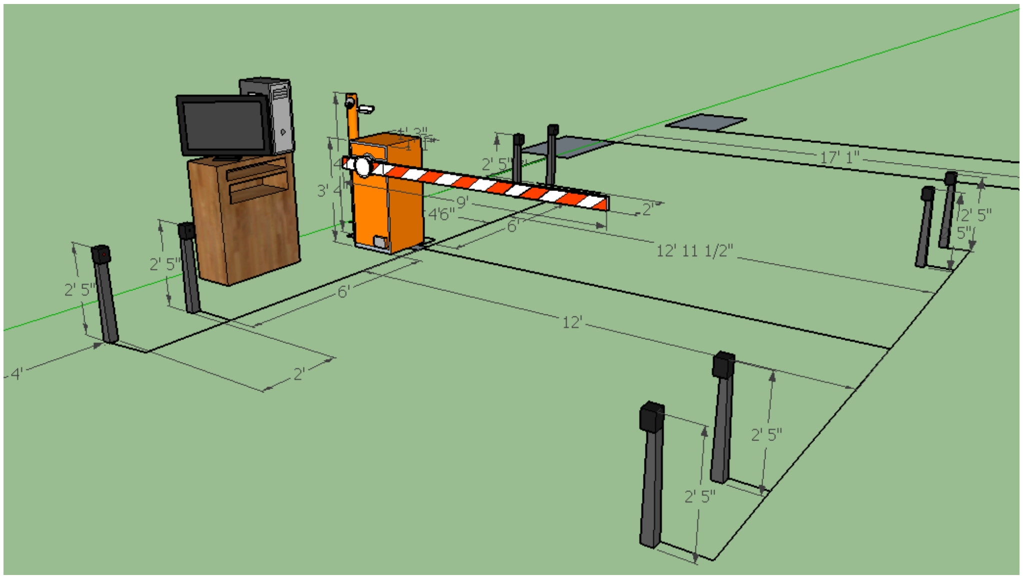


Figure 22. Working Drawing

4.2 Project Structural Organization

Infrared transmitter and receiver are both placed inside an acrylic container. The bigger printed circuit board is the transmitter and the smaller one is the receiver which can be seen in Figure 23.

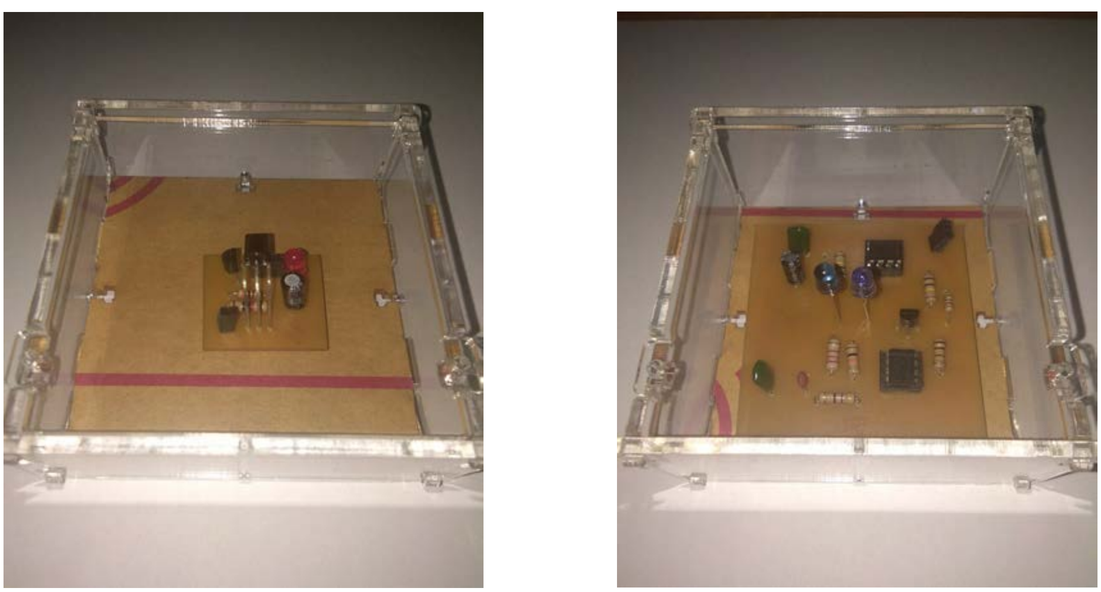


Figure 23. Infrared Sensor

The Graphical User Interface of the System is as shown in Figure 24. It contains the Camera's Live Feed on the left and several panels for the extraction, perspective correction, segmentation and recognition of license plate on the right. The GUI also indicates the vehicle's owner and the status of the database matching process.

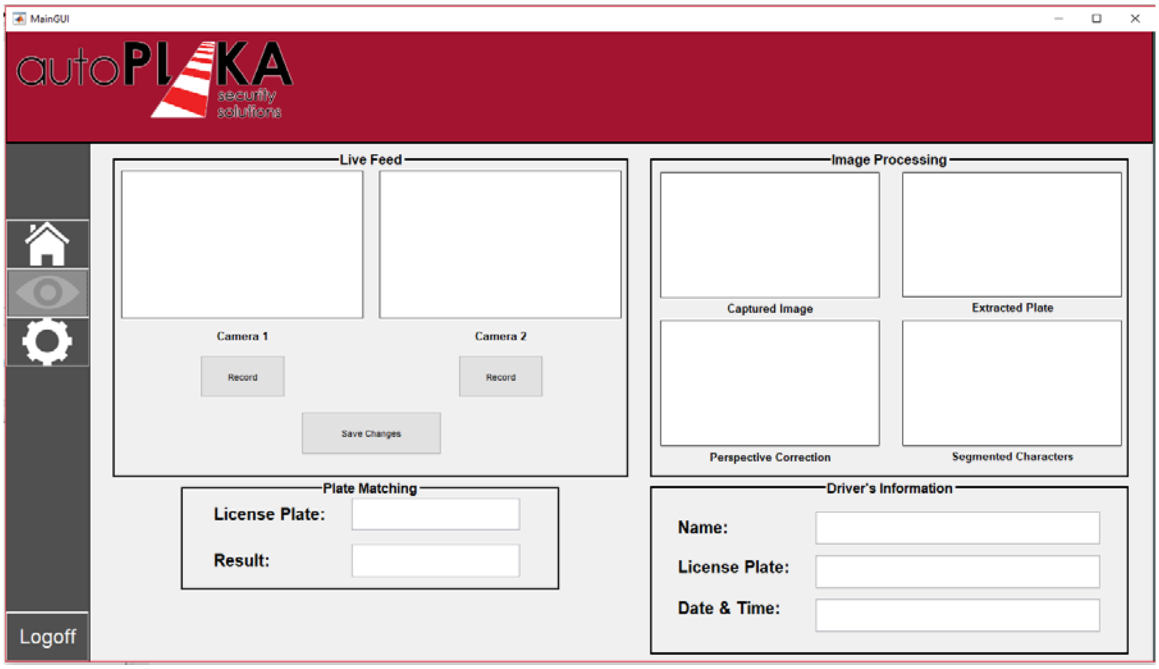


Figure 24. License Plate Recognition Graphical User Interface

The proposed design for the car barrier is shown in Figure 25. It is 1.016 meters in height, 0.356 meters in width, and 0.33 meters in length.

Figure 25. Customized Car Barrier Design

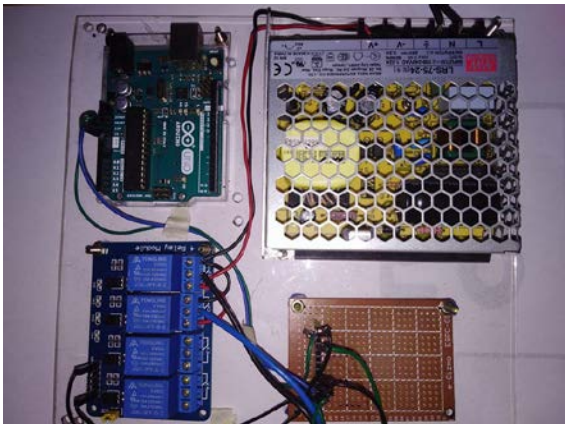


Figure 26. Car barrier's controller design

And lastly the main controller of the car barrier is shown in Figure 26. It is composed of power supply, relay and Arduino Uno. The power supply is rated 24 volts and delivers 3.2 amperes to the motor. The 4-channel relay is the one responsible for reversing the motor's rotation causing the arm of the barrier to swing from zero (0) to ninety (90) degrees and vice versa. . The Arduino Uno contains the pre-defined code for the car barrier's operations or functionality.

4.3 Project Limitations and Capabilities

Based on the testing conducted by the proponents, the system can read new license plate format as shown in Figure 27. The project can operate during daytime and nighttime. The project is effective on cars, trucks and buses but not in motorcycles due to its size and shape. The barrier will automatically open if the license plate is matched. Otherwise, it will be manually operated through remote control if the vehicle is not on the database. In the event of power interruption, the arm can be lifted up to 90 degrees through a crank. The program cannot recognize those plate numbers that are damaged and faded.

Figure 27. Philippines' new license plate format *(https://en.wikipedia.org/wiki/Vehicle_registration_plates_of_the_Philippines)*

4.4 Tabulation of Results and Interpretation

The researchers used 54 samples for the testing of the proposed method. The test is composed of 3 different types of Philippine license plates; these are government plates, public utility vehicle plates and private plates. The test was performed with assumed position of the camera and the plate, which is based on the measurements of the suggested layout for the automatic car barrier system.

Table 8. Results of the Test

LPR Stages	Success/Total Samples	Percentage Accuracy
License Plate Detection and Localization	54/54	100%
License Plate Perspective Correction	54/54	100%
Character Segmentation	50/54	90.91%
Character Recognition	48/54	88.89%

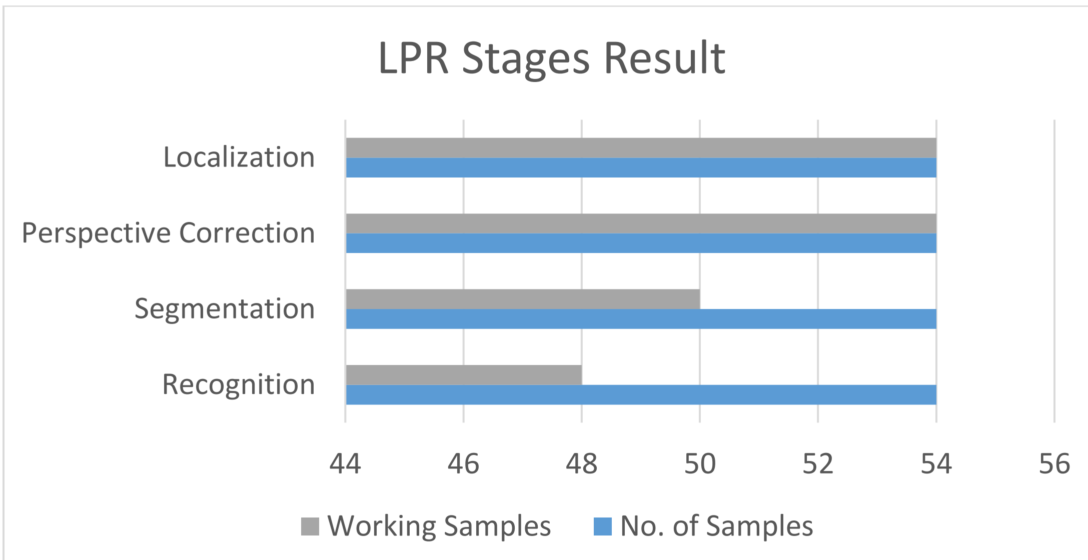


Figure 28. Graphical Data of LPR Stages

Shown in Table 8 are the results from 54 plates tested for recognition and Figure 28 displays its graphical representation. We obtained 100% in license plate localization and perspective correction. This shows that the method used for localization is appropriate for plates that are far from the camera. Meanwhile, the character segmentation still has errors because it detects the outline of the plate and considers it as a character. Lastly, the character recognition has standard accuracy because of the plates that doesn't pass the segmentation. Also, some of the errors of the character recognition is mistaken letters which can be corrected by further training of the ANN.

Overall, the results are acceptable but its efficiency and accuracy can still be developed by improving the program and resolving the errors, especially in the character segmentation and character recognition.

Table 9. LPR Stages Data and Database Matching

LICENSE PLATE RECOGNITION					
License Plate	**LPR Stages**				**Result**
	License Plate Detection and Localization	License Plate Perspective Correction	Character Segmentation	Character Recognition	
1. CLD 689	✓	✓	✓	✓	Matched
2. TOR 191	✓	✓	✓	✓	Matched
3. PUI 309	✓	✓	✓	✓	Matched
4. PIZ 923	✓	✓	✓	✓	Matched
5. CGU 755	✓	✓	✓	✓	Matched
6. XPK 774	✓	✓	✓	✓	Matched
7. WOF 762	✓	✓	✓	✓	Matched
8. CKX 980	✓	✓	✓	✓	Matched
9. XLN 848	✓	✓	✓	✓	Matched
10. AOA 5771	✓	✓	✓	✓	Not matched (not in the DB)
11. UON 407	✓	✓	✓	✓	Matched
12. AAA 6840	✓	✓	✓	✓	Matched
13. ELX 178	✓	✓	✓	✓	Matched
14. UYX 414	✓	✓	✓	✓	Matched
15. ACA 1974	✓	✓	✓	✓	Matched

Table 9. Continuation on LPR Stages Data and Database Matching

LICENSE PLATE	LPR Stages				Result
	License Plate Detection and Localization	License Plate Perspective Correction	License Plate Detection and Localization	Character Recognition	
16. 012345	✓	✓			Not matched
17. WCR 956	✓	✓	✓	✓	Matched
18. ATA 8504	✓	✓	✓	✓	Matched
19. AAK 2709	✓	✓			Not matched
20. UUS 538	✓	✓			Not matched
21. UDD 452	✓	✓	✓	✓	Matched
22. UQZ 488	✓	✓	✓	✓	Matched
23. CHH 418	✓	✓	✓	✓	Matched
24. WPE 775	✓	✓	✓		Not matched
25. SAA 3854	✓	✓	✓	✓	Not Matched (Not in DB)
26. UCQ 552	✓	✓	✓	✓	Matched
27. AEA 9707	✓	✓	✓	✓	Not matched (not in DB)
28. UUN 386	✓	✓			Not matched
29. AAP 8070	✓	✓	✓	✓	Not matched (not in the DB)
30. AEA 7892	✓	✓	✓	✓	Matched
31. NIH 477	✓	✓	✓	✓	Matched
32. CAP 547	✓	✓	✓	✓	Matched

Table 9. Continuation on LPR Stages Data and Database Matching

LICENSE PLATE	LPR Stages				Result
	License Plate Detection and Localization	License Plate Perspective Correction	License Plate Detection and Localization	Character Recognition	
33. NFI 198	✓	✓	✓		Not matched
34. UWH 592	✓	✓	✓	✓	Matched
35. XPJ 298	✓	✓	✓	✓	Matched
36. GAD 558	✓	✓	✓	✓	Matched
37. XMK 289	✓	✓	✓	✓	Matched
38. XNG 473	✓	✓	✓	✓	Matched
39. OME 245	✓	✓	✓	✓	Not matched
40. SAD 547	✓	✓	✓	✓	Matched
41. WDB 515	✓	✓	✓	✓	Matched
42. CLK 439	✓	✓	✓	✓	Matched
43. ZCH 634	✓	✓	✓	✓	Not matched
44. XGR 850	✓	✓	✓	✓	Matched
45. CNH 135	✓	✓	✓	✓	Matched
46. TIZ 922	✓	✓	✓	✓	Matched
47. UQE 664	✓	✓	✓	✓	Matched
48. SJR 900	✓	✓	✓	✓	Matched
49. WTF 943	✓	✓	✓	✓	Matched
50. PIT 105	✓	✓	✓	✓	Matched
51. TIJ 250	✓	✓	✓	✓	Matched
52. TQB 738	✓	✓	✓	✓	Matched
53. ZKG 959	✓	✓	✓	✓	Matched
54. WPI 256	✓	✓	✓	✓	Not matched

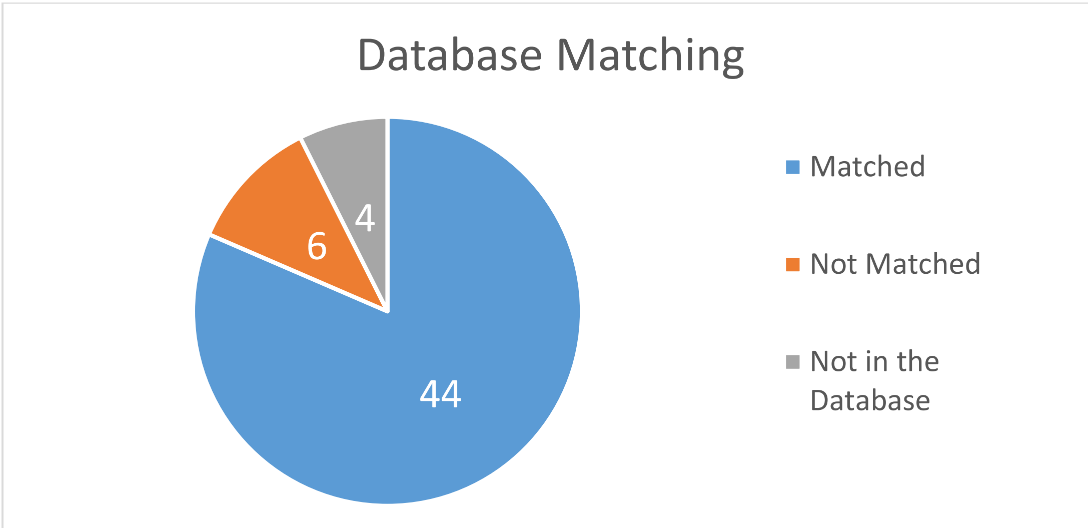


Figure 29. Graphical Data of Database Matching

Figure 29 shows the result of database matching and out of 50 plates in the database that were tested, 44 are recognized and were successfully matched. The 4 plates tested which are not included in the database but were recognized will display an output of “no match” same with plates that were unsuccessfully recognized.

CHAPTER 5

SUMMARY OF FINDINGS, CONCLUSIONS AND RECOMMENDATIONS

5.1 Summary of Findings

The project study entitled “Development of License Plate Recognition (LPR) System for Automatic Car Barrier Using Image Processing and Artificial Neural Network” aims to develop a local LPR system which can cater security to an establishment which includes vehicle tracking, traffic control, and surveillance. The project focused on the identification of the new Philippine license plate format of the vehicles that are passing in Gate one (1) of the Technological University of the Philippines – Manila.

From the pre-testing conducted by the proponents, it was found that the program is capable of recognizing the dummy license plate which resembles the implemented new format, and considering the proposed set-up of the whole system, the program was able to identify the plate numbers better in daylight than those with poor illumination. Nevertheless, the program can still recognize the characters during night time through the camera’s built-in infrared sensors. With the chosen algorithm for image processing and artificial neural network, it can be said that the license plate detection and its perspective correction from the captured image are very efficient. However, there are times that the system is unable to accurately segment each character in the license plate if the processed image is blurred and distorted resulting in an incorrect recognition. Moreover, the character recognition find it hard to recognize which is which, if the letters contains similar attributes or if it has lost some information from the previous stages of the algorithm, i.e. the system sometimes recognized letter ‘X’ as ‘R’, ‘F’ as ‘P’, letter ‘I’ as ‘T’ and so on.

Hence, the researchers aim to develop a more efficient system that is commercially competitive both in price and quality. The proponents will focus on improving the rate of the character segmentation and recognition to increase the current system's overall accuracy. Furthermore, the researchers will design the License Plate Recognition system wherein it can be easily implemented to any traffic management and surveillance industry.

5.2 Conclusions

1. The use of the Arduino Uno as a controller of the car barrier has been significant and valuable wherein the proponents were able to implement the two functionalities (in the microcontroller) below:
    a) Automatic Operation using Database matching The Infrared sensor will trigger the camera to capture a photo of the vehicle if there is an IR beam break happened. This photo will undergo image processing to recognize the license plate. The output will be compared to the database's information and once matched, the car barrier will automatically open.
    b) Manual Operation using Remote Control If the license plate number is not in the database, the guard-on-duty will be responsible for opening the car barrier via remote control.
2. The license plate recognition program effectively executes license plate region extraction and character segmentation and recognition. It performs efficiently with an accuracy of 100% for license plate region extraction, 90.91% for character segmentation, and 88.89% for character recognition based on 54 sample images. The recognition accuracy is dependent on both license plate extraction and character segmentation performances.

3. An automatic logging system will be available for vehicle which enters or exits the Gate 1 of Technological University of the Philippines. It will record all the time of entry and exit of the vehicles with their respective plate numbers. Surveillance of the traffic is also incorporated in this system that will assist the security of the University. This electronic vehicle logging system will ease up the whole process.
4. The proponents successfully installed the system at Gate 1 of the Technological University of the Philippines-Manila.

5.3 Recommendations

For further research and development of this study, the proponents highly suggest the following:

1. To implement more features to feed in the ANN to increase the accuracy of the license plate.
2. To use a more advanced type of sensors for prompting the CCTV camera to capture.
3. To improve the projects program which is its capability to adjust itself for recognition of plates based on different lighting. Moreover, to enhance and to extend the program’s capacity to identify all license plates’ format which includes old and new.

BIBLIOGRAPHY

Akoum, A., Daya, B., & Chauvet, P. (2010). Two Neural Networks For License Number Plates Recognition. *Journal of Theoretical and Applied Information Technology*, 25-32.

Bartolome, L. S., Bandala, A. A., Llorente, C., & Dadios, E. P. (2012). Vehicle Parking Inventory System Utilizing Image Recognition Through Artificial Neural Networks. *IEEE Region 10 Annual International Conference, Proceedings/TENCON*.

Chaturvedi, A., & Sethi, N. (2013). Automatic License Plate Recognition System using SURF Features and RBF Neural Network. *International Journal of Computer Applications*.

Fajar Astuti Hermawati, R. K. (2010). A Real-Time License Plate Detection System for Parking Access. *Telekomnika Vol. 8, No. 2*, 97 – 106.

Gerber, C., & Chung, M. (2016). Number Plate Detection With A Multi-Convolutional Neural Network Approach With Optical Character Recognition For Mobile Devices. *Journal of Information Processing Systems*, 100-108.

Koval, V., Turchenko, V., Kochan, V., Sachenko, A., & Markowsky, G. (2003). Smart License Plate Recognition System Based On Image Processing Using Neural Network. *Proceedings of the 2nd IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2003*, 123-127.

Lukic, M., Tuba, M., & Jovanovic, R. (2016). Two-Stage Algorithm For License Plate Extraction. *INTERNATIONAL JOURNAL OF SYSTEMS APPLICATIONS, ENGINEERING & DEVELOPMENT*, 335-339.

Myra Josephine S. Villanueva, J. P. (n.d.). Automated License Plate Recognition of Philippine License Plates.

Öztürk, F., & Özen, F. (2012). A New License Plate Recognition System Based on Probabilistic Neural Networks. *Procedia Technology*, 124-128.

Panchal, T., Patel, H., & Panchal, A. (2016). License Plate Detection Using Harris Corner and Character Segmentation by Integrated Approach from an Image. *Procedia Computer Science*, 419-425.

Qing Tian, T. G. (2013). Research of Automatic Parking System Based On License Plate Recognition. *Cyberspace Technology (CCT 2013)*.

Ruperto Robles, R. N., Baes, G. B., Solano, G. A., Marcelina Lirazan, M. B., & Imperial, R. H. (2010). Plate Number Recognition System for Beating the Red Light Violators.

Sagar, D., & Dutta, M. (2014). Block-Based Neural Network for Automatic Number Plate Recognition.

Sen, E. J., Deepa Merlin Dixon, K., Anto, A., Anumary, M. V., & Mieheal, D. (2014). Advanced license plate recognition system for car parking. *International Conference on Embedded Systems, ICES 2014*, 162-165.

Singh, S., & Kaur, M. (2014). An Approach to Recognize Characters using Neural Network in LPR System. *International Journal of Computing and Technology*, 1-4.

Sutar, G. T., & Shah, A. V. (2007). Number Plate Recognition Using an Improved Segmentation. *International Journal of Innovative Research in Science, Engineering and Technology (An ISO Certified Organization).*

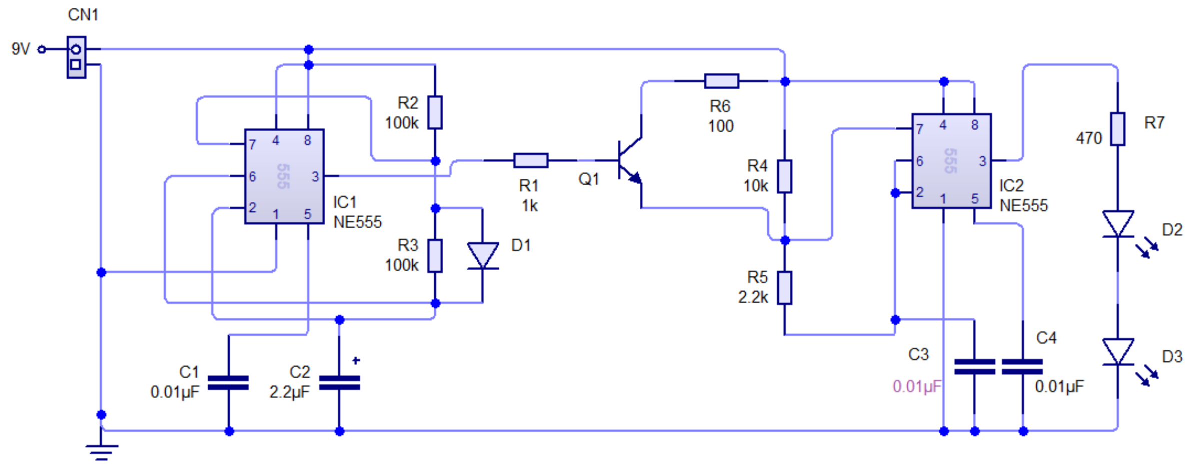
Tian, Q., Guo, T., Qiao, S., Wei, Y., & Fei, W.-w. (2014). Design of Intelligent Parking Management System Based on License Plate Recognition. *Journal of Multimedia*, 774-780.

r, M. J., Ilao, J. P., Cervania, C. M., Ku, V. S., Carvey, D., & Ty, T. R. (n.d.). Automated License Plate Recognition of Philippine License Plates 2 License Plate Detection Module.

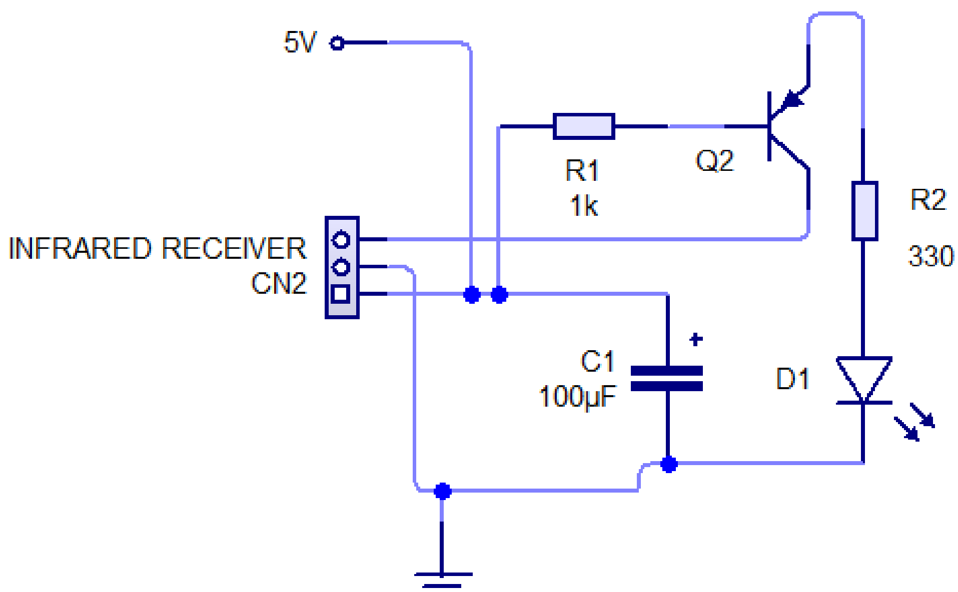
Wu, H., & Li, B. (2011). License plate recognition system. *2011 International Conference on Multimedia Technology*, *ICMT 2011*, 5425-5427.

APPENDIX A

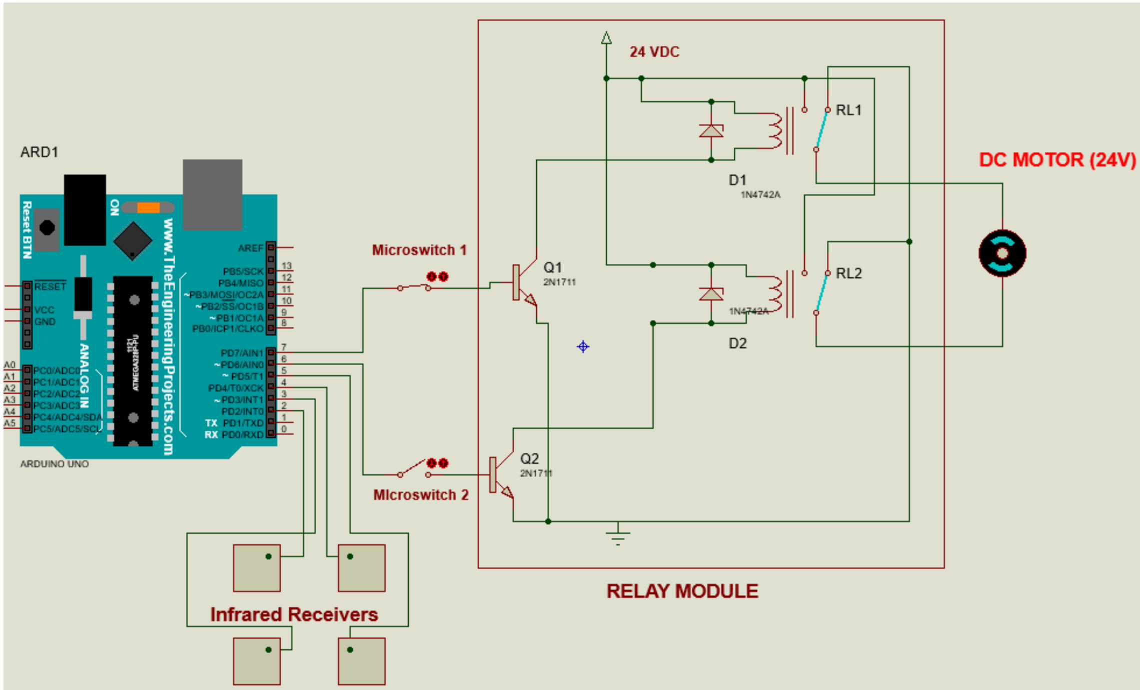
Schematic Diagram



Infrared Transmitter



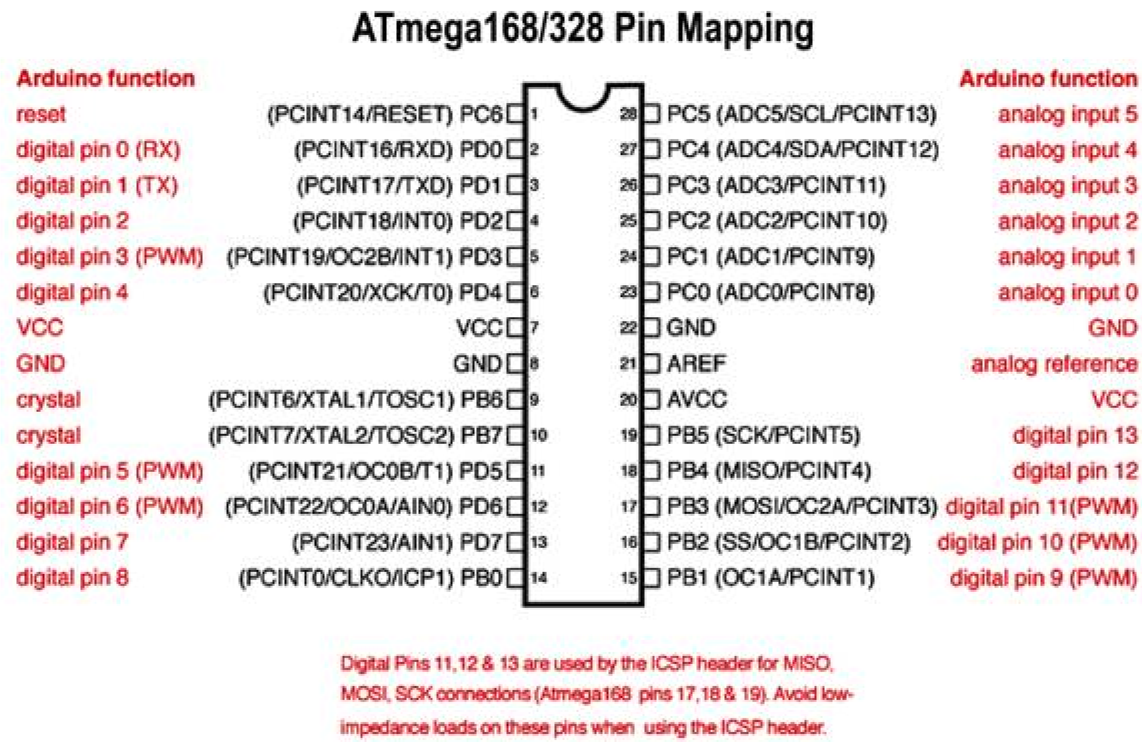
Infrared Receiver



Controller Circuit for Car Barrier

APPENDIX B

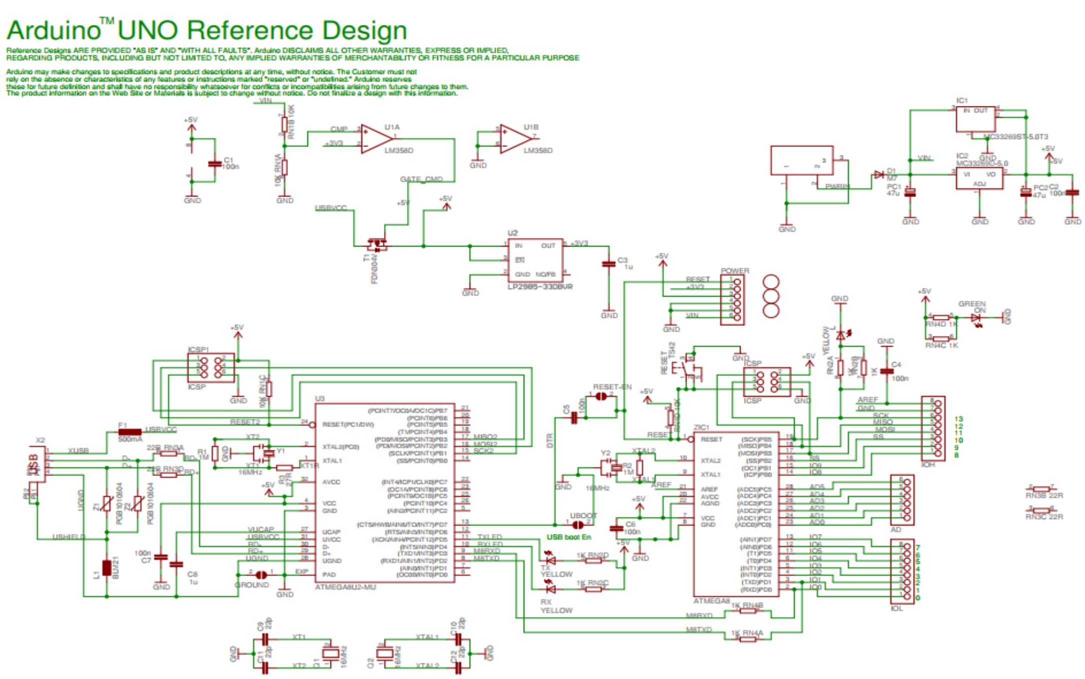
Data Sheets

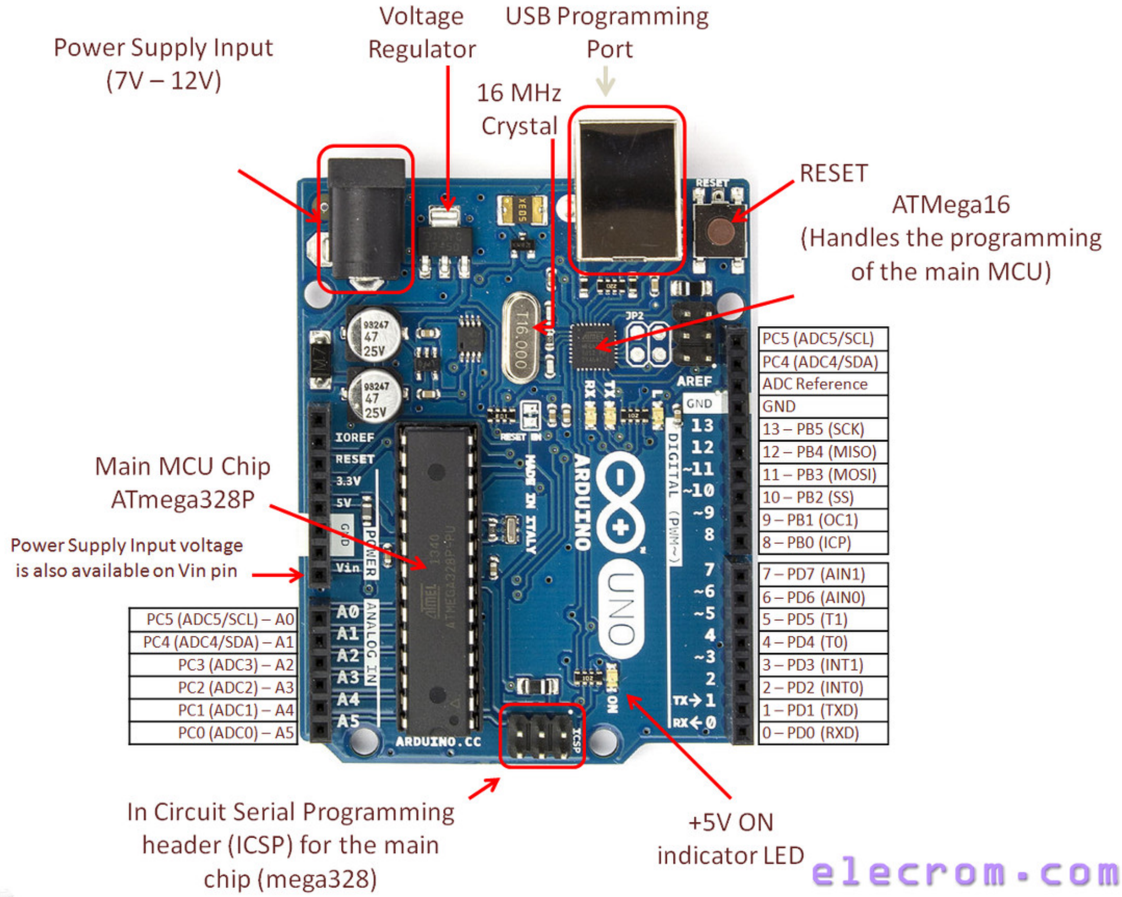


ARDUINO UNO PIN MAPPING TABLE

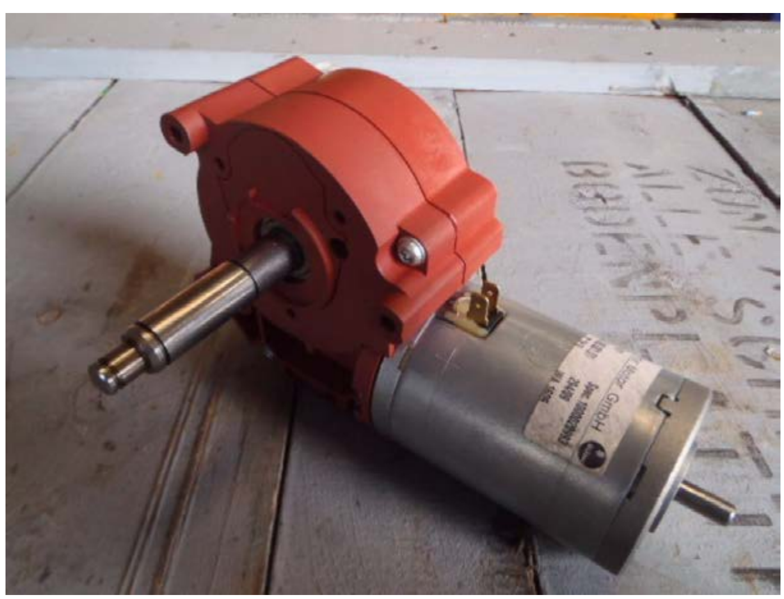
Pin Number	Pin Name	Mapped Pin Name
1	PC6 (PCINT14/RESET)	Reset
2	PD0 (PCINT16/RXD)	Digital pin 0 (RX)
3	PD1 (PCINT17/TXD)	Digital pin 1 (TX)
4	PD2 (PCINT18/INT0)	Digital pin 2
5	PD3 (PCINT19/OC2B/INT1)	Digital pin 3 (PWM)
6	PD4 (PCINT20/XCK/T0)	Digital pin 4
7	VCC	VCC
8	GND	GND
9	PB6 (PCINT6/XTAL1/OSC1)	crystal
10	PB7 (PCINT7/XTAL2/OSC2)	crystal
11	PD5 (PCINT21/OS0B/T1)	Digital pin 5 (PWM)
12	PD6 (PCINT22/OS0A/AIN0)	Digital pin 6 (PWM)

13	PD7 (PCINT23/AIN1)	Digital pin 7
14	PB0 (PCINT7/CLKO/ICP1)	Digital pin 8
15	PB1 (OC1A/PCINT1)	Digital pin 9 (PWM)
16	PB2 (SS/OC1B/PCINT2)	Digital pin 10 (PWM)
17	PB3 (MOS/OC2A/PCINT3)	Digital pin 11 (PWM)
18	PB4 (MISO/PCINT4)	Digital pin 12
19	PB5 (SCK/PCINT5)	Digital pin 13
20	AVCC	VCC
21	AREF	Analog reference
22	GND	GND
23	PC0 (ADC0/PCINT8)	Analog input 0
24	PC1 (ADC1/PCINT9)	Analog input 1
25	PC2 (ADC2/PCINT10)	Analog input 2
26	PC3 (ADC3/PCINT11)	Analog input 3
27	PC4 (ADC4/SDA/PCINT12)	Analog input 4
28	PC5 (ADC4/SCL/PCINT13)	Analog input 5





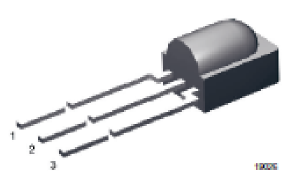
Power Supply Input
(7V – 12V)
Voltage
Regulator
USB Programming
Port
16 MHz
Crystal
RESET
ATMega16
(Handles the programming
of the main MCU)
PC5 (ADC5/SCL)
PC4 (ADC4/SDA)
ADC Reference
GND
13 – PB5 (SCK)
12 – PB4 (MISO)
11 – PB3 (MOSI)
10 – PB2 (SS)
9 – PB1 (OC1)
8 – PB0 (ICP)
7 – PD7 (AIN1)
6 – PD6 (AIN0)
5 – PD5 (T1)
4 – PD4 (T0)
3 – PD3 (INT1)
2 – PD2 (INT0)
1 – PD1 (TXD)
0 – PD0 (RXD)
Main MCU Chip
ATmega328P
Power Supply Input voltage
is also available on Vin pin
PC5 (ADC5/SCL) – A0
PC4 (ADC4/SDA) – A1
PC3 (ADC3) – A2
PC2 (ADC2) – A3
PC1 (ADC1) – A4
PC0 (ADC0) – A5
In Circuit Serial Programming
header (ICSP) for the main
chip (mega328)
+5V ON
indicator LED
elecrom.com



SPECIFICATION OF THE MOTOR

BRAND	Buhler
TYPE	7191972
VOLTAGE (VOLTS)	12 – 24 VDC
SPEED (RPM)	40 RPM at 12 V and 80 RPM at 24 V
AXLE SIZE (MM)	12mm and 8 mm
LENGTH (CM)	20 cm
WIDTH (CM)	10 cm
HEIGHT (CM)	10 cm

IR Receiver Modules for Remote Control Systems



FEATURES

- Very low supply current
- Photo detector and preamplifier in one package
- Internal filter for PCM frequency
- Improved shielding against EMI
- Supply voltage: 2.5 V to 5.5 V
- Improved immunity against ambient light
- Insensitive to supply voltage ripple and noise
- Material categorization: For definitions of compliance please see www.vishay.com/doc?99912

RoHS COMPLIANT

GREEN

MECHANICAL DATA

Pinning for TSOP381.., TSOP383.., TSOP385..:

1 = OUT, 2 = GND, 3 = V_S

Pinning for TSOP391.., TSOP393.., TSOP395..:

1 = OUT, 2 = V_S, 3 = GND

Please see the document "Product Transition Schedule" at www.vishay.com/ir-receiver-modules/ for up-to-date info, when this product will be released.

DESCRIPTION

These products are miniaturized receivers for infrared remote control systems. A PIN diode and a preamplifier are assembled on a lead frame, the epoxy package acts as an IR filter.

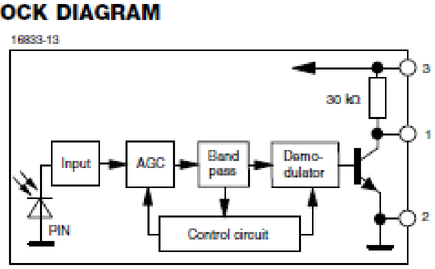
The demodulated output signal can be directly decoded by a microprocessor. The TSOP381.., TSOP391.. are compatible with all common IR remote control data formats. The TSOP383.., TSOP393.. are optimized to better suppress spurious pulses from energy saving fluorescent lamps. The TSOP385.., TSOP395.. have an excellent noise suppression. It is immune to dimmed LCD backlighting and any fluorescent lamps. AGC3 and AGC5 may also suppress some data signals in case of continuous transmission.

This component has not been qualified according to automotive specifications.

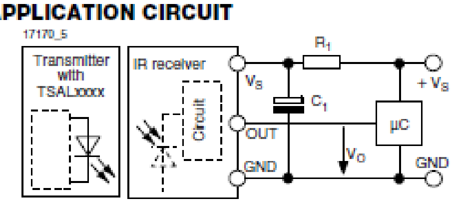
PARTS TABLE

CARRIER FREQUENCY	SHORT BURST AND HIGH DATA RATE (AGC1)		NOISY ENVIROMENTS AND SHORT BURST (AGC3)		VERY NOISY ENVIROMENTS AND SHORT BURSTS (AGC5)	
	PINNING					
	1 = OUT, 2 = GND, 3 = V_S	1 = OUT, 2 = V_S, 3 = GND	1 = OUT, 2 = GND, 3 = V_S	1 = OUT, 2 = V_S, 3 = GND	1 = OUT, 2 = GND, 3 = V_S	1 = OUT, 2 = V_S, 3 = GND
30 kHz	TSOP38130	TSOP39130	TSOP38330	TSOP39330	TSOP38530	TSOP39530
33 kHz	TSOP38133	TSOP39133	TSOP38333	TSOP39333	TSOP38533	TSOP39533
36 kHz	TSOP38136	TSOP39136	TSOP38336	TSOP39336	TSOP38536	TSOP39536
38 kHz	TSOP38138	TSOP39138	TSOP38338	TSOP39338	TSOP38538	TSOP39538
40 kHz	TSOP38140	TSOP39140	TSOP38340	TSOP39340	TSOP38540	TSOP39540
56 kHz	TSOP38156	TSOP39156	TSOP38356	TSOP39356	TSOP38556	TSOP39556

BLOCK DIAGRAM



APPLICATION CIRCUIT



R_1 and C_1 are recommended for protection against EOS. Components should be in the range of 33 Ω < R_1 < 1 kΩ, C_1 > 0.1 µF.

ABSOLUTE MAXIMUM RATINGS				
PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
Supply voltage		V_S	- 0.3 to + 6	V
Supply current		I_S	3	mA
Output voltage		V_O	- 0.3 to (V_S + 0.3)	V
Output current		I_O	5	mA
Junction temperature		T_j	100	°C
Storage temperature range		T_{stg}	- 25 to + 85	°C
Operating temperature range		T_{amb}	- 25 to + 85	°C
Power consumption	$T_{amb} \leq 85$ °C	P_{tot}	10	mW
Soldering temperature	t ≤ 10 s, 1 mm from case	T_{sd}	260	°C

Note

- Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect the device reliability.

ELECTRICAL AND OPTICAL CHARACTERISTICS (T_{amb} = 25 °C, unless otherwise specified)						
PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
Supply current	$E_v = 0$, $V_S = 3.3$ V	I_{SD}	0.27	0.35	0.45	mA
	E_v = 40 klx, sunlight	I_{SH}		0.45		mA
Supply voltage		V_S	2.5		5.5	V
Transmission distance	$E_v = 0$, test signal see fig. 1, IR diode TSAL6200, I_F = 200 mA	d		45		m
Output voltage low	I_{OSL} = 0.5 mA, E_e = 0.7 mW/m², test signal see fig. 1	V_{OSL}			100	mV
Minimum irradiance	Pulse width tolerance: $t_{pi} - 5/f_o < t_{po} < t_{pi} + 6/f_o$, test signal see fig. 1	$E_{e\,min.}$		0.12	0.25	mW/m²
Maximum irradiance	$t_{pi} - 5/f_o < t_{po} < t_{pi} + 6/f_o$, test signal see fig. 1	$E_{e\,max.}$	30			W/m²
Directivity	Angle of half transmission distance	$\varphi_{1/2}$		± 45		deg

TYPICAL CHARACTERISTICS (T_{amb} = 25 °C, unless otherwise specified)

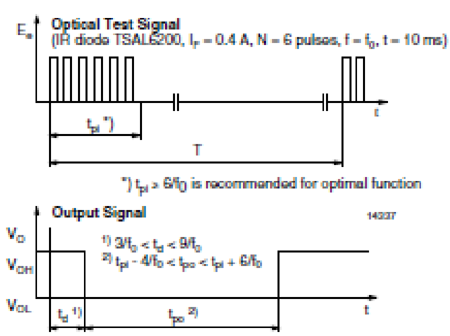


Fig. 1 - Output Active Low

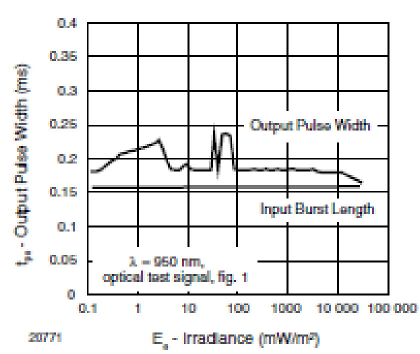


Fig. 2 - Pulse Length and Sensitivity in Dark Ambient

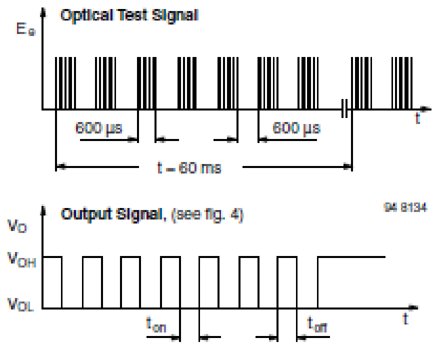


Fig. 3 - Output Function

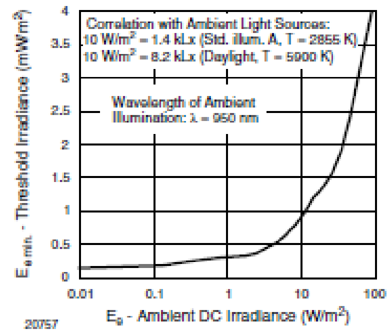


Fig. 6 - Sensitivity in Bright Ambient

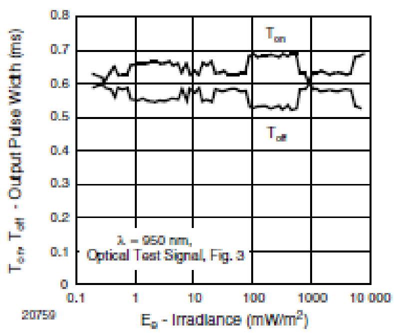


Fig. 4 - Output Pulse Diagram

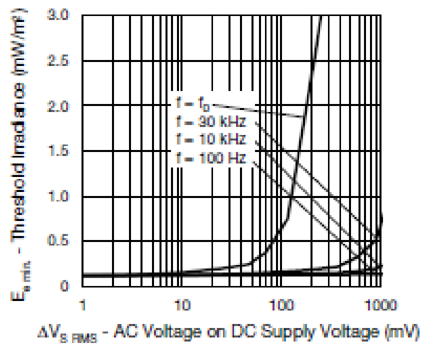


Fig. 7 - Sensitivity vs. Supply Voltage Disturbances

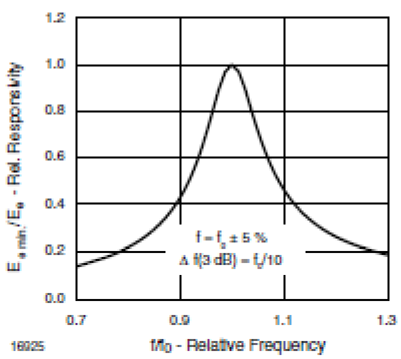


Fig. 5 - Frequency Dependence of Responsivity

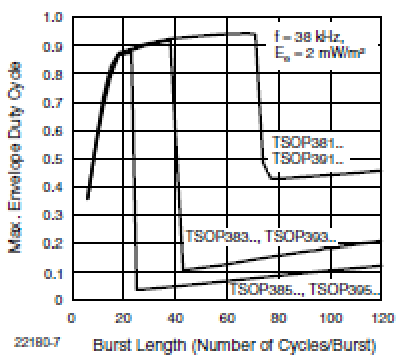


Fig. 8 - Max. Envelope Duty Cycle vs. Burst Length

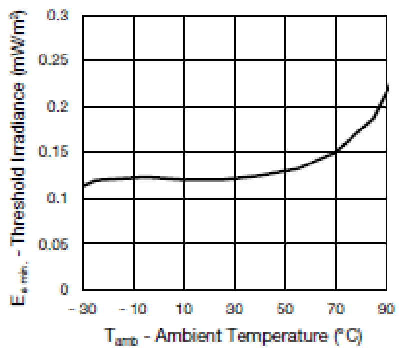


Fig. 9 - Sensitivity vs. Ambient Temperature

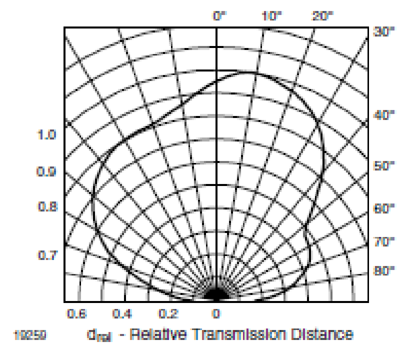


Fig. 12 - Vertical Directivity

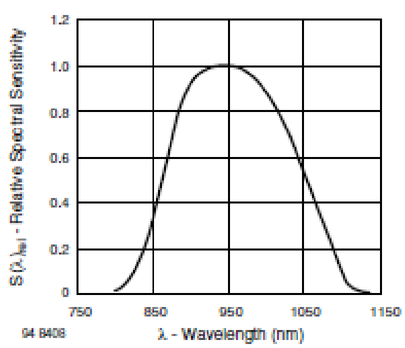


Fig. 10 - Relative Spectral Sensitivity vs. Wavelength

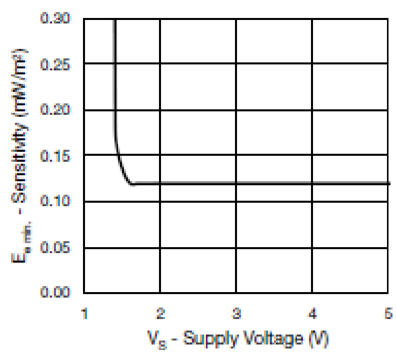


Fig. 13 - Sensitivity vs. Supply Voltage

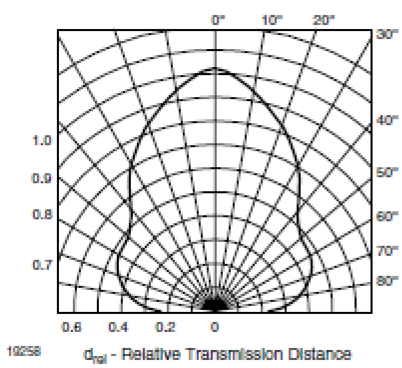


Fig. 11 - Horizontal Directivity

SUITABLE DATA FORMAT

These products are designed to suppress spurious output pulses due to noise or disturbance signals. Data and disturbance signals can be distinguished by the devices according to carrier frequency, burst length and envelope duty cycle. The data signal should be close to the band-pass center frequency (e.g. 38 kHz) and fulfill the conditions in the table below.

When a data signal is applied to the IR receiver in the presence of a disturbance signal, the sensitivity of the receiver is reduced to insure that no spurious pulses are present at the output. Some examples of disturbance signals which are suppressed are:

- DC light (e.g. from tungsten bulb or sunlight)
- Continuous signals at any frequency
- Modulated noise from fluorescent lamps with electronic ballasts (see figure 15 or figure 16)

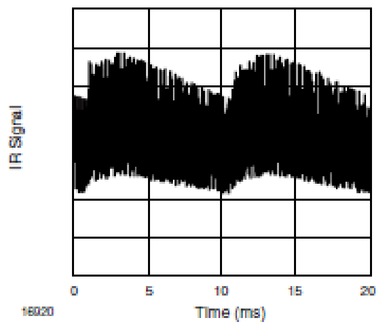


Fig. 14 - IR Signal from Fluorescent Lamp with Low Modulation

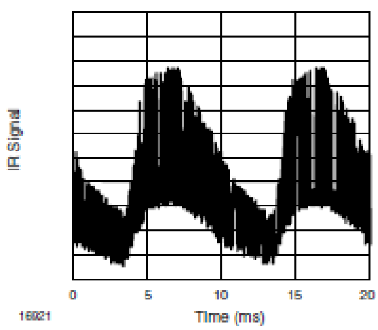


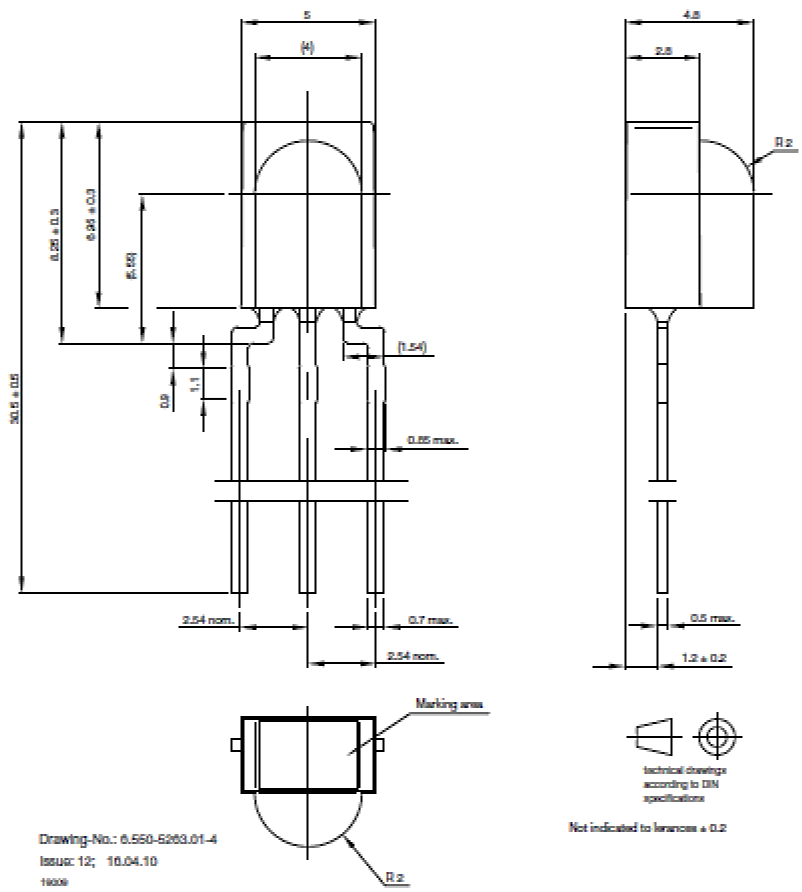
Fig. 15 - IR Signal from Fluorescent Lamp with High Modulation

	TSOP381.., TSOP391..	TSOP383.., TSOP393..	TSOP385.., TSOP395..
Minimum burst length	6 cycles/burst	6 cycles/burst	6 cycles/burst
After each burst of length a minimum gap time is required of	6 to 70 cycles ≥ 10 cycles	6 to 35 cycles ≥ 10 cycles	6 to 24 cycles ≥ 10 cycles
For bursts greater than a minimum gap time in the data stream is needed of	70 cycles > 1.2 x burst length	35 cycles > 6 x burst length	24 cycles > 25 ms
Maximum number of continuous short bursts/second	2000	2000	2000
Recommended for NEC code	yes	yes	yes
Recommended for RC5/RC6 code	yes	yes	yes
Recommended for Sony code	yes	no	no
Recommended for RCMM code	yes	yes	yes
Recommended for r-step code	yes	yes	yes
Recommended for XMP code	yes	yes	yes
Suppression of interference from fluorescent lamps	Common disturbance signals are supressed (example: signal pattern of fig. 15)	Even critical disturbance signals are suppressed (examples: signal pattern of fig. 15 and fig. 16)	Even critical disturbance signals are suppressed (examples: signal pattern of fig. 15 and fig. 16)

Note

- For data formats with long bursts (more than 10 carrier cycles) please see the datasheet for TSOP382.., TSOP384.., TSOP392.., TSOP394..

PACKAGE DIMENSIONS in millimeters



Disclaimer

ALL PRODUCT, PRODUCT SPECIFICATIONS AND DATA ARE SUBJECT TO CHANGE WITHOUT NOTICE TO IMPROVE RELIABILITY, FUNCTION OR DESIGN OR OTHERWISE.

Vishay Intertechnology, Inc., its affiliates, agents, and employees, and all persons acting on its or their behalf (collectively, "Vishay"), disclaim any and all liability for any errors, inaccuracies or incompleteness contained in any datasheet or in any other disclosure relating to any product.

Vishay makes no warranty, representation or guarantee regarding the suitability of the products for any particular purpose or the continuing production of any product. To the maximum extent permitted by applicable law, Vishay disclaims (i) any and all liability arising out of the application or use of any product, (ii) any and all liability, including without limitation special, consequential or incidental damages, and (iii) any and all implied warranties, including warranties of fitness for particular purpose, non-infringement and merchantability.

Statements regarding the suitability of products for certain types of applications are based on Vishay's knowledge of typical requirements that are often placed on Vishay products in generic applications. Such statements are not binding statements about the suitability of products for a particular application. It is the customer's responsibility to validate that a particular product with the properties described in the product specification is suitable for use in a particular application. Parameters provided in datasheets and/or specifications may vary in different applications and performance may vary over time. All operating parameters, including typical parameters, must be validated for each customer application by the customer's technical experts. Product specifications do not expand or otherwise modify Vishay's terms and conditions of purchase, including but not limited to the warranty expressed therein.

Except as expressly indicated in writing, Vishay products are not designed for use in medical, life-saving, or life-sustaining applications or for any other application in which the failure of the Vishay product could result in personal injury or death. Customers using or selling Vishay products not expressly indicated for use in such applications do so at their own risk. Please contact authorized Vishay personnel to obtain written terms and conditions regarding products designed for such applications.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document or by any conduct of Vishay. Product names and markings noted herein may be trademarks of their respective owners.

Material Category Policy

Vishay Intertechnology, Inc. hereby certifies that all its products that are identified as RoHS-Compliant fulfill the definitions and restrictions defined under Directive 2011/65/EU of The European Parliament and of the Council of June 8, 2011 on the restriction of the use of certain hazardous substances in electrical and electronic equipment (EEE) - recast, unless otherwise specified as non-compliant.

Please note that some Vishay documentation may still make reference to RoHS Directive 2002/95/EC. We confirm that all the products identified as being compliant to Directive 2002/95/EC conform to Directive 2011/65/EU.

Vishay Intertechnology, Inc. hereby certifies that all its products that are identified as Halogen-Free follow Halogen-Free requirements as per JEDEC JS709A standards. Please note that some Vishay documentation may still make reference to the IEC 61249-2-21 definition. We confirm that all the products identified as being compliant to IEC 61249-2-21 conform to JEDEC JS709A standards.

APPENDIX C

Source Codes

MATLAB CODE

MainGUI.m

```
function varargout = MainGUI(varargin)
% MAINGUI MATLAB code for MainGUI.fig
%      MAINGUI, by itself, creates a new MAINGUI or raises the existing
%      singleton*.r6
%
%      H = MAINGUI returns the handle to a new MAINGUI or the handle to
%      the existing singleton*.
%
%      MAINGUI('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in MAINGUI.M with the given input
arguments.
%
%      MAINGUI('Property','Value',...) creates a new MAINGUI or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before MainGUI_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to MainGUI_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only
one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help MainGUI

% Last Modified by GUIDE v2.5 07-Mar-2018 11:15:20

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
    'gui_Singleton',  gui_Singleton, ...
    'gui_OpeningFcn', @MainGUI_OpeningFcn, ...
    'gui_OutputFcn',  @MainGUI_OutputFcn, ...
    'gui_LayoutFcn',  [] , ...
    'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
```

```
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before MainGUI is made visible.
function MainGUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to MainGUI (see VARARGIN)
handles.output = hObject;

set(handles.edit8,'string','Username','ForegroundColor',[.502 .502 .502])
set(handles.edit9,'string','Password','ForegroundColor',[.502 .502 .502])
a1=imread('Logo\home.png');

q1=get(handles.pushbutton3, 'position');
g1 = imresize(a1,[q1(4) q1(3)]);
set(handles.pushbutton3,'CData',g1);

a=imread('Logo\eye.png');
q=get(handles.pushbutton4, 'position');
g = imresize(a,[q(4) q(3)]);
set(handles.pushbutton4,'CData',g);

a2=imread('Logo\settings.png');
q2=get(handles.pushbutton5, 'position');
g2 = imresize(a2,[q2(4) q2(3)]);
set(handles.pushbutton5,'CData',g2);

axes(handles.axes18)
imshow(imread('Logo\newlogo.png'));
drawnow;
delete *.jpg;
try
    handles.cam =  ipcam('http://192.168.1.3:80/axis-cgi/mjpg/video.cgi',
'admin', 'admin');
    handles.cam2 = webcam(1);
catch
    errorMessage = sprintf('Camera Not Connected');
    set(handles.edit20,'string',errorMessage);
end

movegui(gcf,'center')
```

```
guidata(hObject, handles);


% --- Outputs from this function are returned to the command line.
function varargout = MainGUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in togglebutton4.
function togglebutton4_Callback(hObject, eventdata, handles)
% hObject    handle to togglebutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of togglebutton4
h = get(hObject,'value');
if h
   set(handles.togglebutton4, 'ForegroundColor', [1 0 0]);
   set(handles.togglebutton4, 'String', 'Recording');
   set(handles.pushbutton9,'enable','on');
else

   set(handles.togglebutton4, 'ForegroundColor', [0 0 0]);
   set(handles.togglebutton4, 'String', 'Record');
   set(handles.pushbutton9,'enable','on');
end


% --- Executes on button press in togglebutton5.
function togglebutton5_Callback(hObject, eventdata, handles)
% hObject    handle to togglebutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
h2 = get(hObject,'value');
if h2
   set(handles.togglebutton5, 'ForegroundColor', [1 0 0]);
   set(handles.togglebutton5, 'String', 'Recording');
   set(handles.pushbutton9,'enable','on');
else
   set(handles.togglebutton5, 'ForegroundColor', [0 0 0]);
```

```
    set(handles.togglebutton5, 'String', 'Record');
    set(handles.pushbutton9,'enable','on');
end
% Hint: get(hObject,'Value') returns toggle state of togglebutton5

% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.pushbutton9,'enable','off');

axes(handles.axes1);
get(handles.axes1,'Position')
hImage = image(zeros(1296,2304));
preview(handles.cam,hImage)
drawnow;

axes(handles.axes6);
get(handles.axes6,'Position')
hImage2 = image(zeros(1296,2304));
preview(handles.cam2,hImage2)
drawnow;

t = datetime('now');
[y,m,d] = ymd(t);
[h,n,s] = hms(t);
ymdhns = sprintf('%g_%g_%g__%g_%g_%g',y,m,d,h,n,s);
vidWriter = VideoWriter(sprintf('frames %s',ymdhns),'MPEG-4');
vidWriter2 = VideoWriter(sprintf('frames2 %s',ymdhns),'MPEG-4');

if
(get(handles.togglebutton4,'value')==1)&&(get(handles.togglebutton5,'value')
==1)
    open(vidWriter);
    open(vidWriter2);
    inputImage = getappdata(handles.btnGo, 'capturedimage');
    Image = imcrop(inputImage,[0 300 2304 (1296-449)]);
    axes(handles.axes2);
    imshow(Image);

    try
        Image = imrotate(Image,10);
        Image = licPlateLocalization(Image);
        ImageGray = rgb2gray(Image);
        ImageGray = medfilt2(ImageGray,[5 5]);
```

```
        edgeImage = edge(ImageGray,'sobel');
        Dy    = strel('rectangle',[30,5]);      % Vertical Extension
        MBy   = imdilate(edgeImage,Dy);
        Dx    = strel('rectangle',[5,30]);      % Horizontal Extension
        MBx   = imdilate(edgeImage,Dx);
        morphMask = MBy.*MBx;
        binaryMask =
imbinarize(ImageGray,'adaptive','ForegroundPolarity','dark');
        binaryImage = morphMask.*binaryMask;
        binaryImage = imfill(binaryImage,'holes');
        [extractedImg,biggestBlob] = licPlateExtraction(Image,binaryImage);
    catch
        disp('Error Extraction')
    end

    try
        [persCorrImageBW,persCorrImageRGB] =
persCorrection(Image,biggestBlob);
    catch
        disp('Error Perspective')
    end

    try
        [recognizedChars,segmentedImagesCell] =
characterSegRec(persCorrImageBW);
    catch
        disp('Error Segment Recog')
    end

    axes(handles.axes4);
    imshow(extractedImg);

    axes(handles.axes3);
    imshow(persCorrImageRGB);

    axes(handles.axes5);
    dirOutput = dir(fullfile('*.jpg'));
    fileNames = {dirOutput.name}';
    montage(fileNames, 'Size', [1 length(segmentedImagesCell)]);

    set(handles.edit1,'string',recognizedChars)

    conn = database('databasefinal', 'root', 'naignalac', 'Vendor', 'MYSQL',
'Server', 'localhost', 'PortNumber', 3305);
    curs = exec(conn, ['SELECT  employee.idemployee'...
        ' , employee.name'...
```

```
    ' , employee.platenum'...
    ' FROM  databasefinal.employee ']);

curs = fetch(curs);
x = curs.Data
A = recognizedChars;
B = x(:,3)                    % List of Plate Numbers in Database
C = strcmp(A,B);
t=datestr(datetime('now'));
for D=find(C==1);
    if any(D)
        MATCHED = 1
    else
        MATCHED = 0
    end
    if MATCHED==1

        ResultName = x(D,2);
        ResultPlate = x(D,3);
        ResultTime = t;

        set(handles.edit3,'String', ResultName);
        set(handles.edit6,'String', ResultPlate);
        set(handles.edit7,'String', ResultTime);
        set(handles.edit2,'String', 'Matched');
    else
        ResultName = 'NA';
        ResultPlate = A;
        ResultTime = t;

        set(handles.edit3,'String', ResultName);
        set(handles.edit6,'String', ResultPlate);
        set(handles.edit7,'String', ResultTime);
        set(handles.edit2,'String', 'No match');
    end
    colnames={'Name','PlateNumber', 'Date_Time', 'Status'};
    tablename='Result';
    data=Result;
    data_table=cell2table(data, 'VariableNames', colnames)
    conn2=database('Result', 'root', 'naignalac');
    fastinsert(conn2, tablename, colnames, data_table)

    query2='SELECT * FROM Result ORDER BY qwert';
    curs2=exec(conn2, query2);
    curs2=fetch(curs2);
```

```
    end

    fprintf(s, '%s', MATCHED); % send answer variable content to arduino
    fclose(s);

elseif
(get(handles.togglebutton4,'value')==0)&&(get(handles.togglebutton5,'value')
==0)
    close(vidWriter);
    close(vidWriter2);
    inputImage = getappdata(handles.btnGo, 'capturedimage');
    Image = imcrop(inputImage,[0 300 2304 (1296-449)]);
    axes(handles.axes2);
    imshow(Image);

    try
        Image = imrotate(Image,10);
        Image = licPlateLocalization(Image);
        ImageGray = rgb2gray(Image);
        ImageGray = medfilt2(ImageGray,[5 5]);
        edgeImage = edge(ImageGray,'sobel');
        Dy    = strel('rectangle',[30,5]);      % Vertical Extension
        MBy   = imdilate(edgeImage,Dy);
        Dx    = strel('rectangle',[5,30]);      % Horizontal Extension
        MBx   = imdilate(edgeImage,Dx);
        morphMask = MBy.*MBx;
        binaryMask =
imbinarize(ImageGray,'adaptive','ForegroundPolarity','dark');
        binaryImage = morphMask.*binaryMask;
        binaryImage = imfill(binaryImage,'holes');
        [extractedImg,biggestBlob] = licPlateExtraction(Image,binaryImage);
    catch
        disp('Error Extraction')
    end

    try
        [persCorrImageBW,persCorrImageRGB] =
persCorrection(Image,biggestBlob);
    catch
        disp('Error Perspective')
    end

    try
        [recognizedChars,segmentedImagesCell] =
characterSegRec(persCorrImageBW);
    catch
```

```
        disp('Error Segment Recog')
    end

    axes(handles.axes4);
    imshow(extractedImg);

    axes(handles.axes3);
    imshow(persCorrImageRGB);

    axes(handles.axes5);
    dirOutput = dir(fullfile('*.jpg'));
    fileNames = {dirOutput.name}';
    montage(fileNames, 'Size', [1 length(segmentedImagesCell)]);

    set(handles.edit1,'string',recognizedChars)

    conn = database('databasefinal', 'root', 'naignalac', 'Vendor', 'MYSQL',
'Server', 'localhost', 'PortNumber', 3305);
    curs = exec(conn, ['SELECT  employee.idemployee'...
        ' , employee.name'...
        ' , employee.platenum'...
        ' FROM  databasefinal.employee ']);

    curs = fetch(curs);
    x = curs.Data
    A = recognizedChars;
    B = x(:,3)                  % List of Plate Numbers in Database
    C = strcmp(A,B);
    t=datestr(datetime('now'));
    for D=find(C==1);
        if any(D)
            MATCHED = 1
        else
            MATCHED = 0
        end
        if MATCHED==1

            ResultName = x(D,2);
            ResultPlate = x(D,3);
            ResultTime = t;

            set(handles.edit3,'String', ResultName);
            set(handles.edit6,'String', ResultPlate);
            set(handles.edit7,'String', ResultTime);
            set(handles.edit2,'String', 'Matched');
        else
```

```
        ResultName = 'NA';
        ResultPlate = A;
        ResultTime = t;

        set(handles.edit3,'String', ResultName);
        set(handles.edit6,'String', ResultPlate);
        set(handles.edit7,'String', ResultTime);
        set(handles.edit2,'String', 'No match');
      end
      colnames={'Name','PlateNumber', 'Date_Time', 'Status'};
      tablename='Result';
      data=Result;
      data_table=cell2table(data, 'VariableNames', colnames)
      conn2=database('Result', 'root', 'naignalac');
      fastinsert(conn2, tablename, colnames, data_table)

      query2='SELECT * FROM Result ORDER BY qwert';
      curs2=exec(conn2, query2);
      curs2=fetch(curs2);

    end

    fprintf(s, '%s', MATCHED); % send answer variable content to arduino
    fclose(s);

  elseif (get(handles.togglebutton4,'value')==1) &&
  (get(handles.togglebutton5,'value')==0)
    open(vidWriter);
    close(vidWriter2);

    inputImage = getappdata(handles.btnGo, 'capturedimage');
    Image = imcrop(inputImage,[0 300 2304 (1296-449)]);
    axes(handles.axes2);
    imshow(Image);

    try
      Image = imrotate(Image,10);
      Image = licPlateLocalization(Image);
      ImageGray = rgb2gray(Image);
      ImageGray = medfilt2(ImageGray,[5 5]);
      edgeImage = edge(ImageGray,'sobel');
      Dy    = strel('rectangle',[30,5]);      % Vertical Extension
      MBy   = imdilate(edgeImage,Dy);
      Dx    = strel('rectangle',[5,30]);      % Horizontal Extension
      MBx   = imdilate(edgeImage,Dx);
      morphMask = MBy.*MBx;
```

```
        binaryMask =
imbinarize(ImageGray,'adaptive','ForegroundPolarity','dark');
        binaryImage = morphMask.*binaryMask;
        binaryImage = imfill(binaryImage,'holes');
        [extractedImg,biggestBlob] = licPlateExtraction(Image,binaryImage);
    catch
        disp('Error Extraction')
    end

    try
        [persCorrImageBW,persCorrImageRGB] =
persCorrection(Image,biggestBlob);
    catch
        disp('Error Perspective')
    end

    try
        [recognizedChars,segmentedImagesCell] =
characterSegRec(persCorrImageBW);
    catch
        disp('Error Segment Recog')
    end

    axes(handles.axes4);
    imshow(extractedImg);

    axes(handles.axes3);
    imshow(persCorrImageRGB);

    axes(handles.axes5);
    dirOutput = dir(fullfile('*.jpg'));
    fileNames = {dirOutput.name}';
    montage(fileNames, 'Size', [1 length(segmentedImagesCell)]);

    set(handles.edit1,'string',recognizedChars)

    conn = database('databasefinal', 'root', 'naignalac', 'Vendor', 'MYSQL',
'Server', 'localhost', 'PortNumber', 3305);
    curs = exec(conn, ['SELECT  employee.idemployee'...
        ' , employee.name'...
        ' , employee.platenum'...
        ' FROM  databasefinal.employee ']);

    curs = fetch(curs);
    x = curs.Data
    A = recognizedChars;
```

```
B = x(:,3)                    % List of Plate Numbers in Database
C = strcmp(A,B);
t=datestr(datetime('now'));
for D=find(C==1);
   if any(D)
      MATCHED = 1
   else
      MATCHED = 0
   end
   if MATCHED==1

      ResultName = x(D,2);
      ResultPlate = x(D,3);
      ResultTime = t;

      set(handles.edit3,'String', ResultName);
      set(handles.edit6,'String', ResultPlate);
      set(handles.edit7,'String', ResultTime);
      set(handles.edit2,'String', 'Matched');
   else
      ResultName = 'NA';
      ResultPlate = A;
      ResultTime = t;

      set(handles.edit3,'String', ResultName);
      set(handles.edit6,'String', ResultPlate);
      set(handles.edit7,'String', ResultTime);
      set(handles.edit2,'String', 'No match');
   end
   colnames={'Name','PlateNumber', 'Date_Time', 'Status'};
   tablename='Result';
   data=Result;
   data_table=cell2table(data, 'VariableNames', colnames)
   conn2=database('Result', 'root', 'naignalac');
   fastinsert(conn2, tablename, colnames, data_table)

   query2='SELECT * FROM Result ORDER BY qwert';
   curs2=exec(conn2, query2);
   curs2=fetch(curs2);

end

fprintf(s, '%s', MATCHED); % send answer variable content to arduino
fclose(s);
```

```
elseif (get(handles.togglebutton4,'value')==0) &&
(get(handles.togglebutton5,'value')==0)
    close(vidWriter);
    open(vidWriter2);
    inputImage = getappdata(handles.btnGo, 'capturedimage');
    Image = imcrop(inputImage,[0 300 2304 (1296-449)]);
    axes(handles.axes2);
    imshow(Image);

    try

        Image = imrotate(Image,10);
        Image = licPlateLocalization(Image);
        ImageGray = rgb2gray(Image);
        ImageGray = medfilt2(ImageGray,[5 5]);
        edgeImage = edge(ImageGray,'sobel');
        Dy    = strel('rectangle',[30,5]);      % Vertical Extension
        MBy   = imdilate(edgeImage,Dy);
        Dx    = strel('rectangle',[5,30]);      % Horizontal Extension
        MBx   = imdilate(edgeImage,Dx);
        morphMask = MBy.*MBx;
        binaryMask =
imbinarize(ImageGray,'adaptive','ForegroundPolarity','dark');
        binaryImage = morphMask.*binaryMask;
        binaryImage = imfill(binaryImage,'holes');
        [extractedImg,biggestBlob] = licPlateExtraction(Image,binaryImage);
    catch
        disp('Error Extraction')
    end

    try
        [persCorrImageBW,persCorrImageRGB] =
persCorrection(Image,biggestBlob);
    catch
        disp('Error Perspective')
    end

    try
        [recognizedChars,segmentedImagesCell] =
characterSegRec(persCorrImageBW);
    catch
        disp('Error Segment Recog')
    end

    axes(handles.axes4);
    imshow(extractedImg);
```

```
axes(handles.axes3);
imshow(persCorrImageRGB);

axes(handles.axes5);
dirOutput = dir(fullfile('*.jpg'));
fileNames = {dirOutput.name}';
montage(fileNames, 'Size', [1 length(segmentedImagesCell)]);

set(handles.edit1,'string',recognizedChars)

conn = database('databasefinal', 'root', 'naignalac', 'Vendor', 'MYSQL',
'Server', 'localhost', 'PortNumber', 3305);
curs = exec(conn, ['SELECT  employee.idemployee'...
   ' , employee.name'...
   ' , employee.platenum'...
   ' FROM  databasefinal.employee ']);

curs = fetch(curs);
x = curs.Data
A = recognizedChars;
B = x(:,3)                    % List of Plate Numbers in Database
C = strcmp(A,B);
t=datestr(datetime('now'));
for D=find(C==1);
   if any(D)
      MATCHED = 1
   else
      MATCHED = 0
   end
   if MATCHED==1

      ResultName = x(D,2);
      ResultPlate = x(D,3);
      ResultTime = t;

      set(handles.edit3,'String', ResultName);
      set(handles.edit6,'String', ResultPlate);
      set(handles.edit7,'String', ResultTime);
      set(handles.edit2,'String', 'Matched');
   else
      ResultName = 'NA';
      ResultPlate = A;
      ResultTime = t;

      set(handles.edit3,'String', ResultName);
```

```
            set(handles.edit6,'String', ResultPlate);
            set(handles.edit7,'String', ResultTime);
            set(handles.edit2,'String', 'No match');
        end
        colnames={'Name','PlateNumber', 'Date_Time', 'Status'};
        tablename='Result';
        data=Result;
        data_table=cell2table(data, 'VariableNames', colnames)
        conn2=database('Result', 'root', 'naignalac');
        fastinsert(conn2, tablename, colnames, data_table)

        query2='SELECT * FROM Result ORDER BY qwert';
        curs2=exec(conn2, query2);
        curs2=fetch(curs2);

    end

    fprintf(s, '%s', MATCHED); % send answer variable content to arduino
    fclose(s);

elseif (get(handles.togglebutton5,'value')==1) &&
(get(handles.togglebutton4,'value')==0)
    open(vidWriter2);
    close(vidWriter);

    while true
        inputImage = snapshot(handles.cam);
        inputImage2 = snapshot(handles.cam2);

        setappdata(handles.btnGo, 'capturedimage', inputImage)

        Image2 = imresize(inputImage2,[768 1366]);
        writeVideo(vidWriter2, Image2);
    end
end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set([handles.uipanel6], 'visible', 'off');
set([handles.uipanel17], 'visible', 'off');
set([handles.uipanel26], 'visible', 'off');
set([handles.uipanel10], 'visible', 'off');
set([handles.uipanel21], 'visible', 'on');
```

```
set(handles.edit8,'String','');
set(handles.edit9,'String','');
set(handles.edit9,'userdata','');

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a double


% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double


% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double


% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set([handles.uipanel6], 'visible', 'off');
set([handles.uipanel17], 'visible', 'on');
set([handles.uipanel26], 'visible', 'off');
set([handles.uipanel21], 'visible', 'off');
set([handles.pushbutton3], 'enable', 'off');
set([handles.pushbutton4], 'enable', 'on');
set([handles.pushbutton5], 'enable', 'on');

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% % handles    structure with handles and user data (see GUIDATA)
set([handles.uipanel17], 'visible', 'off');
set([handles.uipanel6], 'visible', 'on');
set([handles.uipanel26], 'visible', 'off');
set([handles.uipanel21], 'visible', 'off');
set([handles.pushbutton4], 'enable', 'off');
set([handles.pushbutton3], 'enable', 'on');
set([handles.pushbutton5], 'enable', 'on');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set([handles.uipanel17], 'visible', 'off');
set([handles.uipanel6], 'visible', 'off');
set([handles.uipanel26], 'visible', 'on');
set([handles.uipanel21], 'visible', 'off');
set([handles.pushbutton5], 'enable', 'off');
set([handles.pushbutton4], 'enable', 'on');
set([handles.pushbutton3], 'enable', 'on');

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)



function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of edit5 as a double


% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%        str2double(get(hObject,'String')) returns contents of edit6 as a double


% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%        str2double(get(hObject,'String')) returns contents of edit7 as a double


% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text
%        str2double(get(hObject,'String')) returns contents of edit8 as a double


% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit9_Callback(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%        str2double(get(hObject,'String')) returns contents of edit9 as a double


% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- If Enable == 'on', executes on mouse press in 5 pixel border.
% --- Otherwise, executes on mouse press in 5 pixel border or over edit8.
function edit8_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.edit8,'string','','enable','on','ForegroundColor',[0 0 0])
uicontrol(handles.edit8)


% --- If Enable == 'on', executes on mouse press in 5 pixel border.
% --- Otherwise, executes on mouse press in 5 pixel border or over edit9.
function edit9_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.edit9,'string','','enable','on','ForegroundColor',[0 0 0])
uicontrol(handles.edit9)


% --- Executes on button press in pushbutton11.
function pushbutton11_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close(gcf)

% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
if strcmp(get(handles.edit8,'string'), 'admin') &&
strcmp(get(handles.edit9,'Userdata'), 'admin123')
    set([handles.uipanel21], 'visible', 'off');
    set([handles.uipanel6], 'visible', 'off');
```

```
    set([handles.uipanel17], 'visible', 'on');
    set([handles.uipanel26], 'visible', 'off');
    set([handles.uipanel10], 'visible', 'on');
else
    disp('Wrong Username or Password!');
end


% --- Executes on key press with focus on edit9 and none of its controls.
function edit9_KeyPressFcn(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  structure with the following fields (see
MATLAB.UI.CONTROL.UICONTROL)
%   Key: name of the key that was pressed, in lower case
%   Character: character interpretation of the key(s) that was pressed
%   Modifier: name(s) of the modifier key(s) (i.e., control, shift) pressed
% handles    structure with handles and user data (see GUIDATA)
h = handles.edit9;
password = get(h,'Userdata');
key = get(gcf,'currentkey');

switch key
    case 'backspace'
        password = password(1:end-1); % Delete the last character in the
password
    case 'tab'  % Avoid tab triggering the OK button
        gui = getappdata(0,'logindlg');
        uicontrol(gui.OK);
    case 'escape'
        % Close the login dialog
        Escape(gcf,[])
    otherwise
        password = [password get(gcf,'currentcharacter')]; % Add the typed
character to the password
end


SizePass = size(password); % Find the number of asterisks
if SizePass(2) > 0
    asterisk(1,1:SizePass(2)) = '*'; % Create a string of asterisks the same size
as the password
    set(h,'String',asterisk) % Set the text in the password edit box to the asterisk
string
else
    set(h,'String','')
end
```

```
set(h,'Userdata',password) % Store the password in its current state


% --- Executes on button press in pushbutton12.
function pushbutton12_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on button press in pushbutton13.
function pushbutton13_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)



function edit12_Callback(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit12 as text
%        str2double(get(hObject,'String')) returns contents of edit12 as a double


% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit13_Callback(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit13 as text
%        str2double(get(hObject,'String')) returns contents of edit13 as a double


% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit14_Callback(hObject, eventdata, handles)
% hObject    handle to edit14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit14 as text
%        str2double(get(hObject,'String')) returns contents of edit14 as a double


% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit10_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit10 as text
%        str2double(get(hObject,'String')) returns contents of edit10 as a double


% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit11_Callback(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit11 as text
%        str2double(get(hObject,'String')) returns contents of edit11 as a double


% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in pushbutton14.
function pushbutton14_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on button press in pushbutton15.
function pushbutton15_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)



function edit17_Callback(hObject, eventdata, handles)
% hObject    handle to edit17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit17 as text
%        str2double(get(hObject,'String')) returns contents of edit17 as a double


% --- Executes during object creation, after setting all properties.
function edit17_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit18_Callback(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit18 as text
%        str2double(get(hObject,'String')) returns contents of edit18 as a double
```

```
% --- Executes during object creation, after setting all properties.
function edit18_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit19_Callback(hObject, eventdata, handles)
% hObject    handle to edit19 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit19 as text
%        str2double(get(hObject,'String')) returns contents of edit19 as a double


% --- Executes during object creation, after setting all properties.
function edit19_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit19 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit15_Callback(hObject, eventdata, handles)
% hObject    handle to edit15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit15 as text
%        str2double(get(hObject,'String')) returns contents of edit15 as a double


% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function edit16_Callback(hObject, eventdata, handles)
% hObject    handle to edit16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit16 as text
%        str2double(get(hObject,'String')) returns contents of edit16 as a double


% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit20_Callback(hObject, eventdata, handles)
% hObject    handle to edit20 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit20 as text
%        str2double(get(hObject,'String')) returns contents of edit20 as a double


% --- Executes during object creation, after setting all properties.
function edit20_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit20 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

licPlateLocalization.m

```
function PL = licPlateLocalization(Im)
In    = 1;
I     = im2double(rgb2gray(Im));
%% Vertical Sobel Masking
SM    = [1 0 -1;2 0 -2;1 0 -1];        % Sobel Vertical Mask
IS    = imfilter(I,SM,'replicate');     % Filter Image Using Sobel Mask
IS    = IS.^2;                          % Consider Just Value of Edges & Fray Weak
Edges
IS    = IS > 0.25;
IS    = edgeFilter(IS,10,60);
IS    = imclearborder(IS);
%% Normalization
IS    = (IS-min(IS(:)))/(max(IS(:))-min(IS(:))); % Normalization
%% Threshold (Otsu)
level = graythresh(IS);                 % Threshold Based on Otsu Method
IS    = im2bw(IS,level);
%% Histogram
S     = sum(IS,2);                      % Edge Horizontal Histogram
%% Plate Location
T1    = 0.35;                           % Threshold On Edge Histogram
PR    = find(S > (T1*max(S)));          % Candidate Plate Rows
%% Masked Plate
Msk   = zeros(size(I));
Msk(PR,:) = 1;                          % Mask
MB    = Msk.*IS;                        % Candidate Plate (Edge Image)
%% Morphology (Dilation - Vertical)
Dy    = strel('rectangle',[100,4]);     % Vertical Extension
```

```
MBy   = imdilate(MB,Dy);              % By Dilation
MBy   = imfill(MBy,'holes');          % Fill Holes
%% Morphology (Dilation - Horizontal)
Dx    = strel('rectangle',[4,100]);     % Horizontal Extension
MBx   = imdilate(MB,Dx);              % By Dilation
MBx   = imfill(MBx,'holes');          % Fill Holes
%% Joint Places
BIM   = MBx.*MBy;                     % Joint Places
%% Morphology (Dilation - Horizontal)
Dy    = strel('rectangle',[4,30]);      % Horizontal Extension
MM    = imdilate(BIM,Dy);             % By Dilation
MM    = imfill(MM,'holes');           % Fill Holes
%% Erosion
Dr    = strel('line',50,0);           % Erosion
BL    = imerode(MM,Dr);
%% Find Biggest Binary Region (As a Plate Place)
[L,num] = bwlabel(BL);                % Label (Binary Regions)
Areas   = zeros(num,1);
for i = 1:num                         % Compute Area Of Every Region
[r,c,v]  = find(L == i);              % Find Indexes
Areas(i) = sum(v);                    % Compute Area
end
[La,Lb] = find(Areas==max(Areas));      % Biggest Binary Region Index
%% Post Processing
[a,b]   = find(L==La);                % Find Biggest Binary Region (Plate)
[nRow,nCol] = size(im2double(Im))
FM      = zeros(nRow,nCol);           % Smooth and Enlarge Plate Place
T       = 70;                         % Extend Plate Region By T Pixel
jr      = (min(a)-T :max(a)+T);
jc      = (min(b)-T :max(b)+T);
jr      = jr(jr >= 1 & jr <= nRow);
jc      = jc(jc >= 1 & jc <= nCol);
FM(jr,jc) = 1;
[rowsek,colsek,~] = size(FM);

PL = imcrop(Im,[min(jc),min(jr),max(jc)-min(jc),max(jr)-min(jr)]);
end
```

edgeFilter.m

```
function filteredEdge = edgeFilter(inputEdge,minLen,maxLen)
   inputEdge = inputEdge > 0.75;
   [rows,cols,~] = size(inputEdge);
   M = zeros(rows,cols);
   N = zeros(rows,cols);

   for i=3:(rows-2)
```

```
    for j=3:(cols-2)
      if inputEdge(i,j)==1
        if (inputEdge(i-1,j-1)+inputEdge(i-1,j)+inputEdge(i-
1,j+1)+inputEdge(i,j-1)) > 0
          M(i,j) = max([M(i-1,j-1),M(i-1,j),M(i-1,j+1),M(i,j-1)]) + 1;
        else
          M(i,j) = max([M(i-2,j-1),M(i-2,j),M(i-2,j+1),M(i-1,j-2),M(i-
1,j+2),M(i,j-2)]) + 1;
        end
      end
    end
  end

  for k=(rows-2):3
    for l=(cols-2):3
      if inputEdge(k,l)==1
        if (inputEdge(k+1,l-
1)+inputEdge(k+1,l)+inputEdge(k+1,l+1)+inputEdge(k,l+1)) > 0
          N(k,l) = max([N(k+1,l-1),N(k+1,l),N(k+1,l+1),N(k,l+1)]) + 1;
        else
          N(k,l) = max([N(k+2,l-1),N(k+2,l),N(k+2,l+1),N(k+1,l-
2),N(k+1,l+2),N(k,l+2)]) + 1;
        end
      end
    end
  end

  for m=3:(rows-2)
    for n=3:(cols-2)
      if inputEdge(m,n)==1
        if ((M(m,n)+N(m,n)) > maxLen || (M(m,n)+N(m,n)) < minLen)
          inputEdge(m,n) = 0;
        end
      end
    end
  end

  filteredEdge = inputEdge;
end
```

ExtractNLargestBlobs.m

```
function binaryImage = ExtractNLargestBlobs(binaryImage,
numberToExtract)
  [labeledImage, numberOfBlobs] = bwlabel(binaryImage);
  blobMeasurements = regionprops(labeledImage, 'area');
  allAreas = [blobMeasurements.Area];
```

```
    [sortedAreas, sortIndexes] = sort(allAreas, 'descend');
    biggestBlob = ismember(labeledImage, sortIndexes(1:numberToExtract));
    binaryImage = biggestBlob > 0;
end
```

licPlateExtraction.m

```
function [extractedImg,extractedBlob] =
licPlateExtraction(image,binaryImage)
binaryImage = medfilt2(imfill(binaryImage, 'holes'));
biggestBlob = ExtractNLargestBlobs(binaryImage,1);

[labeledImage,numberOfBlobs] = bwlabel(biggestBlob,8);
blobMeasurements =
regionprops(labeledImage,'BoundingBox','Centroid','MajorAxisLength','Mino
rAxisLength');
imgBoundBox = [blobMeasurements.BoundingBox];
allWidths = imgBoundBox(3:4:end);
allHeights = imgBoundBox(4:4:end);
aspectRatio = [allWidths./allHeights ;  allHeights./allWidths];
aspectRatios = max(aspectRatio, [], 1);

axisMaj = [blobMeasurements.MajorAxisLength];
axisMin = [blobMeasurements.MinorAxisLength];
axisAspectRatio = [axisMaj./axisMin ;  axisMin./axisMaj];
axisAspectRatios = max(axisAspectRatio, [], 1);

[~,aspectRatiosIndex] = min(abs(aspectRatios-(39/14)));
aspectRatiosIndexVal = aspectRatios(aspectRatiosIndex);
[~,axisAspectRatiosIndex] = min(abs(axisAspectRatios-
aspectRatiosIndexVal));
axisAspectRatiosIndexVal = axisAspectRatios(axisAspectRatiosIndex);

compactIndexes = find(axisAspectRatios == axisAspectRatiosIndexVal);
binaryImage = ismember(labeledImage, compactIndexes);
extractedBlob = binaryImage;
blobProperties = regionprops(extractedBlob, 'BoundingBox');
blobProperties.BoundingBox;
extractedImg = imcrop(image, blobProperties.BoundingBox);
end
```

characterSegRec.m

```
function [recognizedCharacters,segmentedImagesCell] =
characterSegRec(image,plateType)

image = imresize(~im2bw(image),[140 390]);
```

```
horizontalProjection = sum(image,2);
horzless = horizontalProjection <= 50;
horzless = double(horzless);
xx = findseq(horzless);
yy = find(xx(:,1)==0);
xx = xx(yy,:);
[~,yy] = max(xx(:,4));
xx = xx(yy,:);
newImage = imcrop(image,[0 xx(2) 390 (xx(3)-xx(2))]);
newImage = bwareafilt(newImage,[500 3000]);
verticalProjection = sum(newImage,1);
vert = smooth(verticalProjection);
vertEdges = vert == 0;
vertEdges = double(vertEdges);
vertA = findseq(vertEdges);
vertB = find(vertA(:,1)==1);
vertA = vertA(vertB,:);
leftEdge = vertA(1,:);
rightEdge = vertA(length(vertB),:);

newImage = imcrop(newImage,[leftEdge(3) 0 rightEdge(2)-leftEdge(3)
(xx(3)-xx(2))]);
verticalProjection = sum(newImage,1);
vert = smooth(verticalProjection);
vertless = vert < 5;
vertless = double(vertless);
aa = findseq(vertless);
bb = find(aa(:,1)==1);
aa = aa(bb,:);
[~,bb] = max(aa(:,4));
aa = aa(bb,:);

if aa(4) > 15
    meanAA = (aa(2) + aa(3))/2;
    alphaImage = imcrop(newImage,[0 0 meanAA xx(3)-xx(2)]);
    numImage = imcrop(newImage,[meanAA 0 390 xx(3)-xx(2)]);
    gaborArray = gaborFilterBank(5,8,39,39);
    stats = regionprops(alphaImage);
    for index=1:length(stats)
        x = ceil(stats(index).BoundingBox(1));
        y = ceil(stats(index).BoundingBox(2));
        widthX = floor(stats(index).BoundingBox(3)-1);
        widthY = floor(stats(index).BoundingBox(4)-1);
        letSub(index) = {~alphaImage(y:y+widthY,x:x+widthX,:)};
        letSub{index} = medfilt2(imresize(letSub{index}, [70 35]));
        imwrite(imresize(letSub{index},[200 100]),sprintf('%d.jpg',index));
```

```
        outputCell{index} =
newLetterNet(gaborFeatures(~(letSub{index}),gaborArray,5,5));
    end

    stats2 = regionprops(numImage);
    for index=1:length(stats2)
        x = ceil(stats2(index).BoundingBox(1));
        y = ceil(stats2(index).BoundingBox(2));
        widthX = floor(stats2(index).BoundingBox(3)-1);
        widthY = floor(stats2(index).BoundingBox(4)-1);
        numSub(index) = {~numImage(y:y+widthY,x:x+widthX,:)};
        numSub{index} = medfilt2(imresize(numSub{index}, [70 35]));
        imwrite(imresize(numSub{index},[200
100]),sprintf('%d.jpg',index+length(stats)));
        outputCell2{index} =
newNumberNet(gaborFeatures(~(numSub{index}),gaborArray,5,5));
    end

    outputChars = letterGenerator(outputCell,length(stats));
    outputCharacters = outputChars(~cellfun('isempty',outputChars));
    outputChars2 = numberGenerator(outputCell2,length(outputCell2));
    outputCharacters2 = outputChars2(~cellfun('isempty',outputChars2));

    for j=1:length(outputCharacters)
        letterStr = outputCharacters(j);
        if j==1
            letterStrFinal = letterStr;
        else
            letterStrFinal = strcat(letterStrFinal,letterStr);
        end
    end

    for j=1:length(outputCharacters2)
        numberStr = outputCharacters2(j);
        if j==1
            numberStrFinal = numberStr;
        else
            numberStrFinal = strcat(numberStrFinal,numberStr);
        end
    end

    letterSegmentedImg = letSub(~cellfun('isempty',letSub));
    numberSegmentedImg = numSub(~cellfun('isempty',numSub));
    segmentedImagesCell = [letSub,numSub];
    recognizedCharacters = strcat(letterStrFinal,{' '},numberStrFinal);
```

```
else
    gaborArray = gaborFilterBank(5,8,39,39);
    stats = regionprops(newImage);
    for index=1:length(stats)
        x = ceil(stats(index).BoundingBox(1));
        y = ceil(stats(index).BoundingBox(2));
        widthX = floor(stats(index).BoundingBox(3)-1);
        widthY = floor(stats(index).BoundingBox(4)-1);
        numSub(index) = {~newImage(y:y+widthY,x:x+widthX,:)};
        numSub{index} = medfilt2(imresize(numSub{index}, [70 35]));
        imwrite(imresize(numSub{index},[200 100]),sprintf('%d.jpg',index));
        outputCell3{index} =
newNumberNet(gaborFeatures(~(numSub{index}),gaborArray,5,5));
    end

    outputChars3 = numberGenerator(outputCell3,length(stats));
    outputCharacters3 = outputChars3(~cellfun('isempty',outputChars3));

    for k=1:length(outputCharacters3)
        numberStr = outputCharacters3(k);
        if k==1
            numberStrFinal = numberStr;
        else
            numberStrFinal = strcat(numberStrFinal,numberStr);
        end
    end
    segmentedImagesCell = numSub;
    recognizedCharacters = numberStrFinal;
end
```

gaborFeatures.m

```
function featureVector = gaborFeatures(img,gaborArray,d1,d2)
if (nargin ~= 4)        % Check correct number of arguments
    error('Please use the correct number of input arguments!')
end

if size(img,3) == 3     % Check if the input image is grayscale
    warning('The input RGB image is converted to grayscale!')
    img = rgb2gray(img);
end

img = double(img);
[u,v] = size(gaborArray);
gaborResult = cell(u,v);
for i = 1:u
    for j = 1:v
```

```
        gaborResult{i,j} = imfilter(img, gaborArray{i,j});
    end
end

featureVector = [];
for i = 1:u
    for j = 1:v
        gaborAbs = abs(gaborResult{i,j});
        gaborAbs = downsample(gaborAbs,d1);
        gaborAbs = downsample(gaborAbs.',d2);
        gaborAbs = gaborAbs(:);
        gaborAbs = (gaborAbs-mean(gaborAbs))/std(gaborAbs,1);

        featureVector =  [featureVector; gaborAbs];
    end
end
```

gaborFilterBank.m

```
function gaborArray = gaborFilterBank(u,v,m,n)
if (nargin ~= 4)    % Check correct number of arguments
    error('There must be four input arguments (Number of scales and
orientations and the 2-D size of the filter)!')
end

gaborArray = cell(u,v);
fmax = 0.25;
gama = sqrt(2);
eta = sqrt(2);

for i = 1:u
    fu = fmax/((sqrt(2))^(i-1));
    alpha = fu/gama;
    beta = fu/eta;
    for j = 1:v
        tetav = ((j-1)/v)*pi;
        gFilter = zeros(m,n);

        for x = 1:m
            for y = 1:n
                xprime = (x-((m+1)/2))*cos(tetav)+(y-((n+1)/2))*sin(tetav);
                yprime = -(x-((m+1)/2))*sin(tetav)+(y-((n+1)/2))*cos(tetav);
                gFilter(x,y) = (fu^2/(pi*gama*eta))*exp(-
((alpha^2)*(xprime^2)+(beta^2)*(yprime^2)))*exp(1i*2*pi*fu*xprime);
            end
        end
        gaborArray{i,j} = gFilter;
```

```
   end
end
```

newLetterNet.m

```
function [Y,Xf,Af] = newNumberNet(X,~,~)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Generated by Neural Network Toolbox function genFunction, 07-Mar-2018
16:28:34.
%
% [Y] = myNeuralNetworkFunction(X,~,~) takes these arguments:
%
%   X = 1xTS cell, 1 inputs over TS timesteps
%   Each X{1,ts} = 3920xQ matrix, input #1 at timestep ts.
%
% and returns:
%   Y = 1xTS cell of 1 outputs over TS timesteps.
%   Each Y{1,ts} = 10xQ matrix, output #1 at timestep ts.
%
% where Q is number of samples (or series) and TS is the number of
timesteps.

%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1.xoffset = [-1.58557871323388;-1.36883858800001;-
1.28400396428064…
x1_step1.gain =
[0.655162671787491;0.454864364849124;0.429900944485071…
x1_step1.ymin = -1;

% Layer 1
b1 = [-1.3951018350005326;0.69328015782538432;-
0.0038894959725340669…
IW1_1 = [0.018420738558500384 -0.0067407244898310585
0.037144707480674451…

% Layer 2
b2 = [0.044998252617359437;0.48113585100098372;-
0.64140119121701522…
LW2_1 = [-1.2603482186191752 -1.062916836197592
0.74401312827759369…
```

```
% ===== SIMULATION ========

% Format Input Arguments
isCellX = iscell(X);
if ~isCellX, X = {X}; end;

% Dimensions
TS = size(X,2); % timesteps
if ~isempty(X)
   Q = size(X{1},2); % samples/series
else
   Q = 0;
end

% Allocate Outputs
Y = cell(1,TS);

% Time loop
for ts=1:TS

   % Input 1
   Xp1 = mapminmax_apply(X{1,ts},x1_step1);

   % Layer 1
   a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*Xp1);

   % Layer 2
   a2 = softmax_apply(repmat(b2,1,Q) + LW2_1*a1);

   % Output 1
   Y{1,ts} = a2;
end

% Final Delay States
Xf = cell(1,0);
Af = cell(2,0);

% Format Output Arguments
if ~isCellX, Y = cell2mat(Y); end
end

% ===== MODULE FUNCTIONS ========

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings)
y = bsxfun(@minus,x,settings.xoffset);
```

```
y = bsxfun(@times,y,settings.gain);
y = bsxfun(@plus,y,settings.ymin);
end

% Competitive Soft Transfer Function
function a = softmax_apply(n,~)
if isa(n,'gpuArray')
   a = iSoftmaxApplyGPU(n);
else
   a = iSoftmaxApplyCPU(n);
end
end
function a = iSoftmaxApplyCPU(n)
nmax = max(n,[],1);
n = bsxfun(@minus,n,nmax);
numerator = exp(n);
denominator = sum(numerator,1);
denominator(denominator == 0) = 1;
a = bsxfun(@rdivide,numerator,denominator);
end
function a = iSoftmaxApplyGPU(n)
nmax = max(n,[],1);
numerator = arrayfun(@iSoftmaxApplyGPUHelper1,n,nmax);
denominator = sum(numerator,1);
a = arrayfun(@iSoftmaxApplyGPUHelper2,numerator,denominator);
end
function numerator = iSoftmaxApplyGPUHelper1(n,nmax)
numerator = exp(n - nmax);
end
function a = iSoftmaxApplyGPUHelper2(numerator,denominator)
if (denominator == 0)
   a = numerator;
else
   a = numerator ./ denominator;
end
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n,~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end
```

letterGenerator.m

```
function outputCharacters = letterGenerator(outputCell,stats)
ls = stats;
for i=1:ls
```

```
if ~isempty(outputCell{i})
    %charL=ocator = find(outputCell{i}.');
    charLocator = vec2ind(outputCell{i});

    switch charLocator
        case 1
            charSel{i} = ('A');
        case 2
            charSel{i} = ('B');
        case 3
            charSel{i} = ('C');
        case 4
            charSel{i} = ('D');
        case 5
            charSel{i} = ('E');
        case 6
            charSel{i} = ('F');
        case 7
            charSel{i} = ('G');
        case 8
            charSel{i} = ('H');
        case 9
            charSel{i} = ('I');
        case 10
            charSel{i} = ('J');
        case 11
            charSel{i} = ('K');
        case 12
            charSel{i} = ('L');
        case 13
            charSel{i} = ('M');
        case 14
            charSel{i} = ('N');
        case 15
            charSel{i} = ('O');
        case 16
            charSel{i} = ('P');
        case 17
            charSel{i} = ('Q');
        case 18
            charSel{i} = ('R');
        case 19
            charSel{i} = ('S');
        case 20
            charSel{i} = ('T');
        case 21
```

```
            charSel{i} = ('U');
        case 22
            charSel{i} = ('V');
        case 23
            charSel{i} = ('W');
        case 24
            charSel{i} = ('X');
        case 25
            charSel{i} = ('Y');
        case 26
            charSel{i} = ('Z');
      end
   end
end

for j=1:ls
   if ~isempty(charSel{j})
      outputCharacters{j} = char(charSel{j});
   end
end
```

newNumberNet.m

```
function [Y,Xf,Af] = newNumberNet(X,~,~)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Generated by Neural Network Toolbox function genFunction, 07-Mar-2018
16:28:34.
%
% [Y] = myNeuralNetworkFunction(X,~,~) takes these arguments:
%
%   X = 1xTS cell, 1 inputs over TS timesteps
%   Each X{1,ts} = 3920xQ matrix, input #1 at timestep ts.
%
% and returns:
%   Y = 1xTS cell of 1 outputs over TS timesteps.
%   Each Y{1,ts} = 10xQ matrix, output #1 at timestep ts.
%
% where Q is number of samples (or series) and TS is the number of
timesteps.

%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
```

```
x1_step1.xoffset = [-1.58557871323388;-1.36883858800001;-
1.28400396428064…
x1_step1.gain =
[0.655162671787491;0.454864364849124;0.429900944485071…
x1_step1.ymin = -1;

% Layer 1
b1 = [-1.3951018350005326;0.69328015782538432;-
0.0038894959725340669…
IW1_1 = [0.018420738558500384 -0.0067407244898310585
0.037144707480674451…

% Layer 2
b2 = [0.044998252617359437;0.48113585100098372;-
0.64140119121701522…
LW2_1 = [-1.2603482186191752 -1.062916836197592
0.74401312827759369…

% ===== SIMULATION ========

% Format Input Arguments
isCellX = iscell(X);
if ~isCellX, X = {X}; end;

% Dimensions
TS = size(X,2); % timesteps
if ~isempty(X)
   Q = size(X{1},2); % samples/series
else
   Q = 0;
end

% Allocate Outputs
Y = cell(1,TS);

% Time loop
for ts=1:TS

   % Input 1
   Xp1 = mapminmax_apply(X{1,ts},x1_step1);

   % Layer 1
   a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*Xp1);

   % Layer 2
   a2 = softmax_apply(repmat(b2,1,Q) + LW2_1*a1);
```

```
    % Output 1
    Y{1,ts} = a2;
end

% Final Delay States
Xf = cell(1,0);
Af = cell(2,0);

% Format Output Arguments
if ~isCellX, Y = cell2mat(Y); end
end

% ===== MODULE FUNCTIONS ========

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings)
y = bsxfun(@minus,x,settings.xoffset);
y = bsxfun(@times,y,settings.gain);
y = bsxfun(@plus,y,settings.ymin);
end

% Competitive Soft Transfer Function
function a = softmax_apply(n,~)
if isa(n,'gpuArray')
    a = iSoftmaxApplyGPU(n);
else
    a = iSoftmaxApplyCPU(n);
end
end
function a = iSoftmaxApplyCPU(n)
nmax = max(n,[],1);
n = bsxfun(@minus,n,nmax);
numerator = exp(n);
denominator = sum(numerator,1);
denominator(denominator == 0) = 1;
a = bsxfun(@rdivide,numerator,denominator);
end
function a = iSoftmaxApplyGPU(n)
nmax = max(n,[],1);
numerator = arrayfun(@iSoftmaxApplyGPUHelper1,n,nmax);
denominator = sum(numerator,1);
a = arrayfun(@iSoftmaxApplyGPUHelper2,numerator,denominator);
end
function numerator = iSoftmaxApplyGPUHelper1(n,nmax)
numerator = exp(n - nmax);
```

```
end
function a = iSoftmaxApplyGPUHelper2(numerator,denominator)
if (denominator == 0)
   a = numerator;
else
   a = numerator ./ denominator;
end
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n,~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end
```

numberGenerator.m

```
function outputCharacters = numberGenerator(outputCell,stats)
ls = stats;
for i=1:ls
   if ~isempty(outputCell{i})
      %charL=ocator = find(outputCell{i}.');
      charLocator = vec2ind(outputCell{i});

      switch charLocator
         case 1
            charSel{i} = ('0');
         case 2
            charSel{i} = ('1');
         case 3
            charSel{i} = ('2');
         case 4
            charSel{i} = ('3');
         case 5
            charSel{i} = ('4');
         case 6
            charSel{i} = ('5');
         case 7
            charSel{i} = ('6');
         case 8
            charSel{i} = ('7');
         case 9
            charSel{i} = ('8');
         case 10
            charSel{i} = ('9');
      end
   end
end
```

```
for j=1:ls
   if ~isempty(charSel{j})
      outputCharacters{j} = char(charSel{j});
   end
end
```

persCorrection.m

```
function [persCorrectedImgBW,persCorrectedImgRGB] =
persCorrection(Image,biggestBlob)
   persCorrectedImgRGB = perscorr(biggestBlob, Image);
   persCorrectedImgBW = rgb2gray(persCorrectedImgRGB);
   persCorrectedImgBW =
imbinarize(persCorrectedImgBW,'adaptive','ForegroundPolarity','dark');
   persCorrectedImgBW = imresize(persCorrectedImgBW, [140 390]);
   persCorrectedImgRGB = imresize(persCorrectedImgRGB, [140 390]);
end
```

perscorr.m

```
function correctedImage = perscorr(Imager, Imageref)
Image = imfill(Imager,'holes');
[I,J]=find(Image>max(Image(:))/2);
   IJ=[I,J];
   [~,idx]=min(IJ*[1 1; -1 -1; 1 -1; -1 1].');
   corners=IJ(idx,:);
corx = corners(:,2);
cory = corners(:,1);

[corx cory]=sortPolyFromClockwiseStartingFromTopLeft(corx,cory);
base = ([1 1; size(Imageref,2) 1; size(Imageref,2) size(Imageref,1); 1
size(Imageref,1)]);
figure()
plot(base)
tf = fitgeotrans([corx cory],base,'projective');
tfRef = imref2d(size(Imageref),[1 size(Imageref,2)],[1 size(Imageref,1)]);
xf1 = imwarp(Imageref,tf,'OutputView',tfRef);
correctedImage=xf1;
end
```

sortPolyFromClockwiseStartingFromTopLeft.m

```
function [X, Y] = sortPolyFromClockwiseStartingFromTopLeft( X, Y )
top_vertices_Y = sortCoordinatesAccordToX(Y);
for i=1:4
   for j=1:4
      if top_vertices_Y(i) == Y(j)
```

```
            top_vertices_X(i) = X(j);
        end
    end
end

top_vertices_X = top_vertices_X';
X = top_vertices_X;
Y = top_vertices_Y;

if X(1) > X(2)
    top_vertices_X(1) = X(2);
    top_vertices_Y(1) = Y(2);
    top_vertices_X(2) = X(1);
    top_vertices_Y(2) = Y(1);
end

X = top_vertices_X;
Y = top_vertices_Y;

if X(3) < X(4)
    top_vertices_X(3) = X(4);
    top_vertices_Y(3) = Y(4);
    top_vertices_X(4) = X(3);
    top_vertices_Y(4) = Y(3);
end

X = top_vertices_X;
Y = top_vertices_Y;
end
```

ARDUINO CODE

```
#include <IRremote.h>
#define CCW 6
#define CW 7
#define IR1 2
#define IR2 3
#define IR3 4
#define IR4 5

char matlabData =0;

int RECV_PIN = 11;
IRrecv irrecv(RECV_PIN);
decode_results results;
int OK=4010536330;
//int UP=value;    //insert remote value
//int DOWN=value;    //insert remote value

void setup() {
  Serial.begin(9600);
  pinMode(IR1, OUTPUT) ;
  pinMode(IR2, OUTPUT) ;
  pinMode(IR3, OUTPUT) ;
  pinMode(IR4, OUTPUT) ;
  pinMode(CCW, OUTPUT); //Set CW as an output
  pinMode(CW, OUTPUT);
  digitalWrite(IR1, LOW) ;
  digitalWrite(IR2, LOW) ;
  digitalWrite(IR3, LOW) ;
  digitalWrite(IR4, LOW) ;
  digitalWrite(CCW, HIGH);
  digitalWrite(CW, HIGH);
  irrecv.enableIRIn(); // Start the receiver
}

void loop()
{

   // For Manual Operation

if (irrecv.decode(&results))                //Automatic UP and Down
{
  Serial.println(results.value,DEC);
  if(results.value==4010536330){
```

```
  digitalWrite(CCW,LOW);
  delay(3000);
  digitalWrite(CCW,HIGH);
  delay(5000);
  digitalWrite(CW, LOW);
  delay(2000);
  digitalWrite(CW,HIGH);
  delay(4000);
  }

//  if(results.value==UP){              //Up barrier
//     digitalWrite(CCW,LOW);
//    delay(3000);
//    digitalWrite(CCW,HIGH);
//    delay(5000);
//  }
//
//  if(results.value==DOWN){              //Down Barrier
//    digitalWrite(CW, LOW);
//    delay(2000);
//    digitalWrite(CW,HIGH);
//    delay(4000);
//  }


irrecv.resume();
}


//OPeration from Matlab after Database Matching
//Matlab

 if(Serial.available()>0){
 matlabData=Serial.read();


  if(matlabData=='1')
  {

    digitalWrite(CCW,LOW);
    delay(3000);
    digitalWrite(CCW,HIGH);
    delay(5000);
    digitalWrite(CW, LOW);
    delay(2000);
    digitalWrite(CW,HIGH);
```

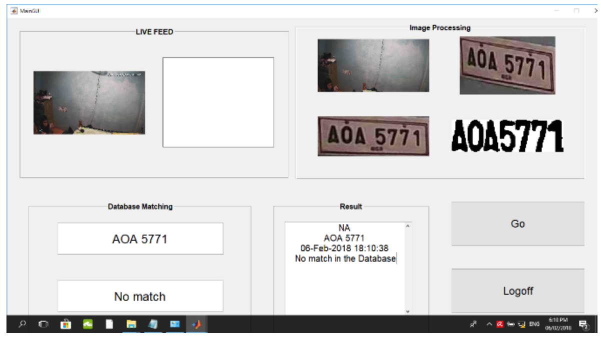
```
      delay(4000);

      }
}


  //trigger camera when entering
  while ((digitalRead(IR1)==LOW))
  {
      if ((digitalRead(IR2) == LOW))
      {
       // Serial.println(1);
       Serial.write(1);
  }
}

//Closing of barrier during exit
while (digitalRead(IR2)==LOW)
  {
    while ((digitalRead(IR1) == LOW)&&(digitalRead(IR2)==LOW))
  {
    if ((digitalRead(IR2)==HIGH))
    {
    //Serial.println("CLOSE BARRIER");
    digitalWrite(CW, LOW);
    delay(2000);
    digitalWrite(CW,HIGH);
    delay(4000);
  }
}
  }

//trigger of camera for exit
 while ((digitalRead(IR4)==LOW))
  {
      if ((digitalRead(IR3) == LOW))
      {
       // Serial.println(1);
       Serial.write(1);
  }
}

//Closing of barrier during enter
while (digitalRead(IR3)==LOW)
  {
    while ((digitalRead(IR3) == LOW)&&(digitalRead(IR4)==LOW))
```

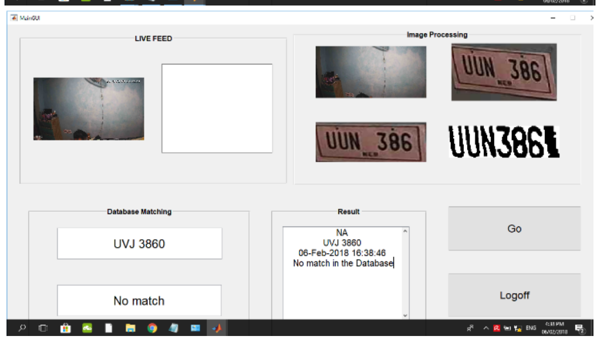
```
 {
  if ((digitalRead(IR3)==HIGH))
  {
  //Serial.println("CLOSE BARRIER");
  digitalWrite(CW, LOW);
  delay(2000);
  digitalWrite(CW,HIGH);
  delay(4000);
 }
}
 }

}
```

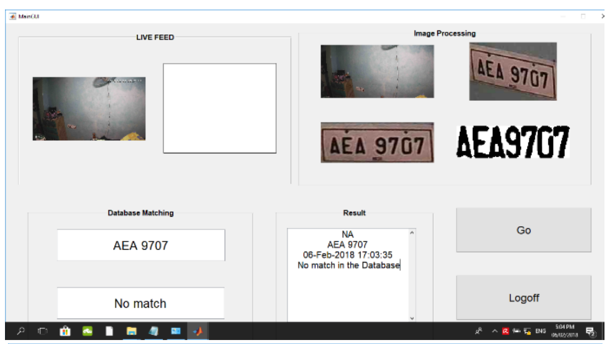
APPENDIX D

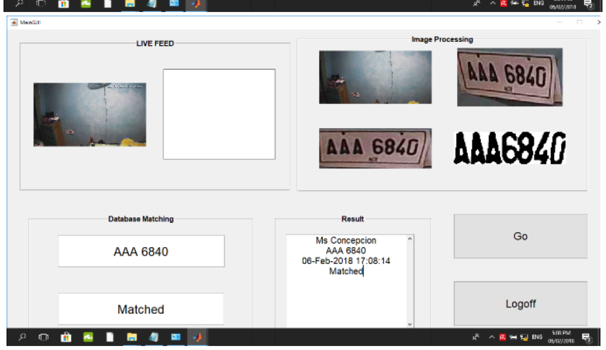
DATA

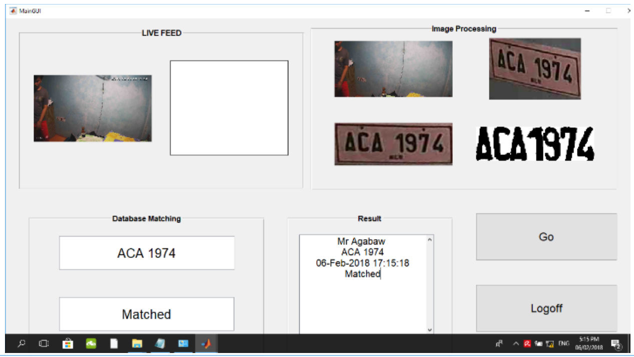
GUI of the 54 tested dummy licensed plates

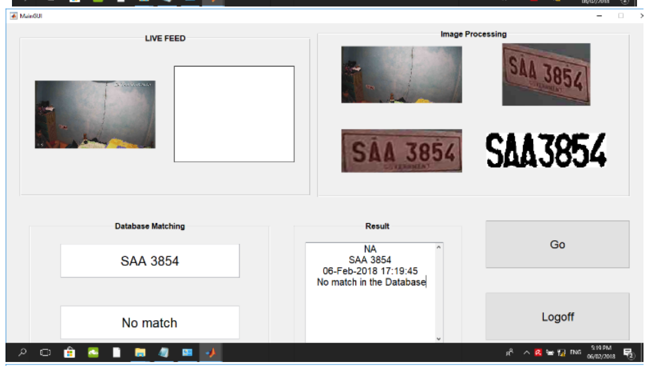


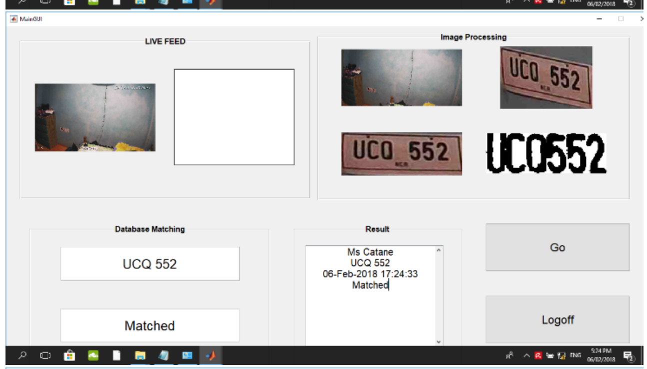


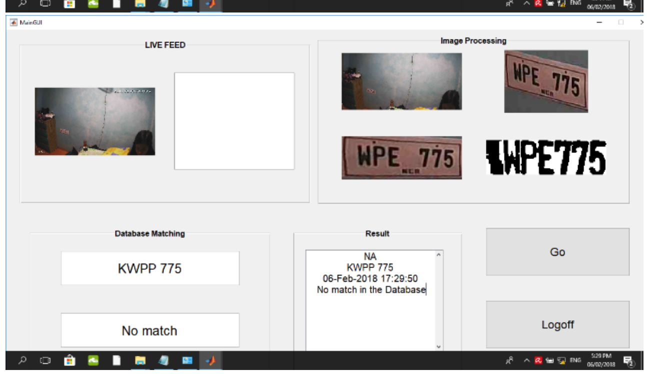


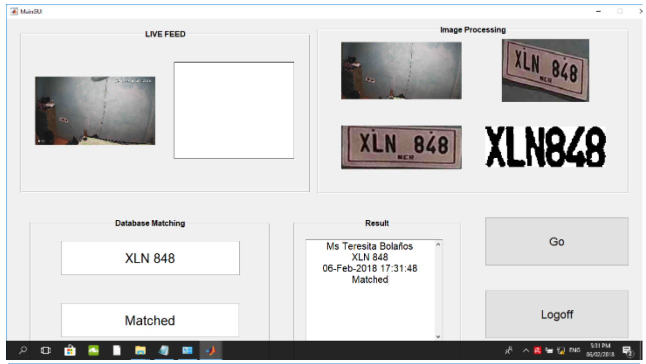


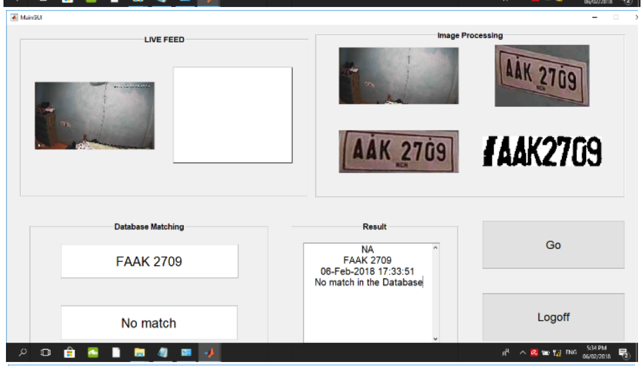

LIVE FEED
Image Processing
Database Matching
ACA 1974
Matched
Result
Mr Agabaw
ACA 1974
06-Feb-2018 17:15:18
Matched
Go
Logoff

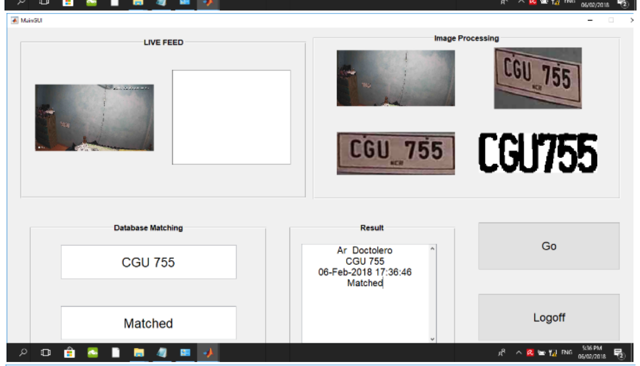

LIVE FEED
Image Processing
Database Matching
SAA 3854
No match
Result
NA
SAA 3854
06-Feb-2018 17:19:45
No match in the Database
Go
Logoff

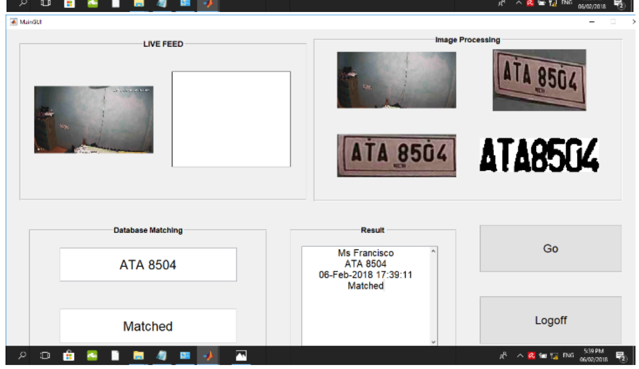

LIVE FEED
Image Processing
Database Matching
UCQ 552
Matched
Result
Ms Catane
UCQ 552
06-Feb-2018 17:24:33
Matched
Go
Logoff

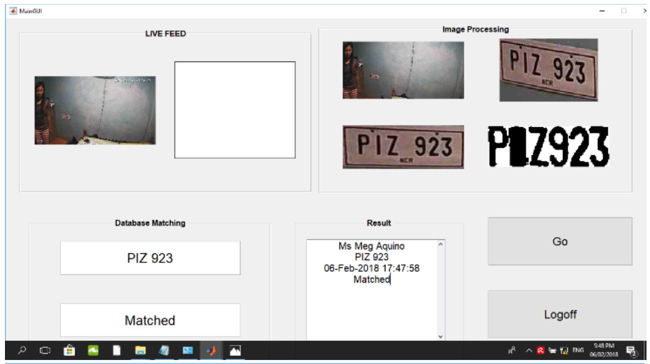
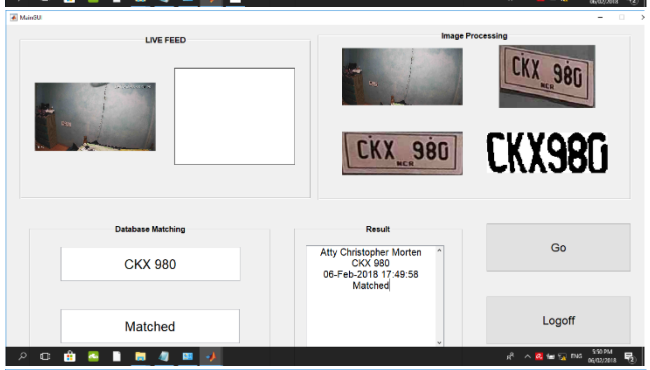
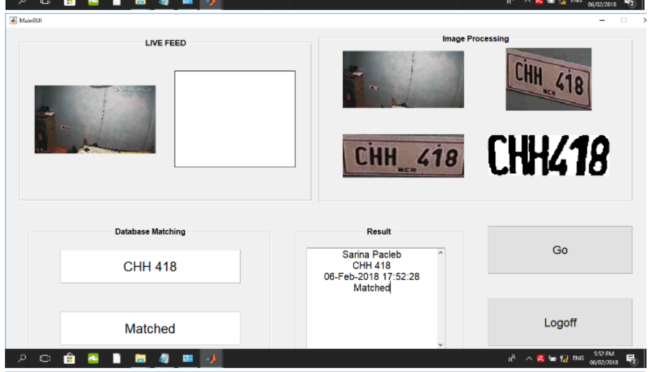
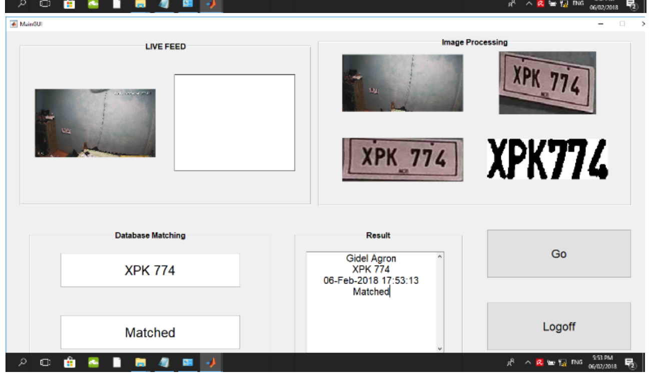

LIVE FEED
Image Processing
Database Matching
KWPP 775
No match
Result
NA
KWPP 775
06-Feb-2018 17:29:50
No match in the Database
Go
Logoff

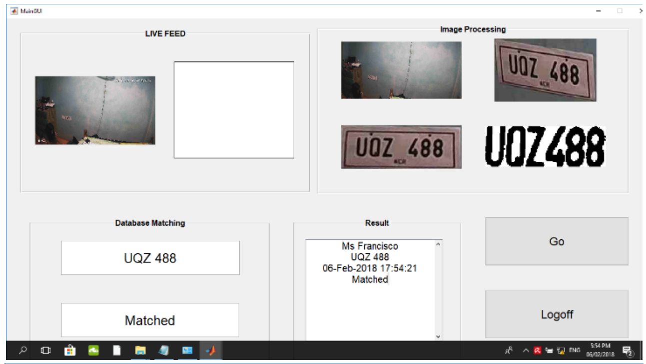

MainGUI
LIVE FEED
Image Processing
XLN848
Database Matching
XLN 848
Matched
Result
Ms Teresita Bolaños
XLN 848
06-Feb-2018 17:31:48
Matched
Go
Logoff

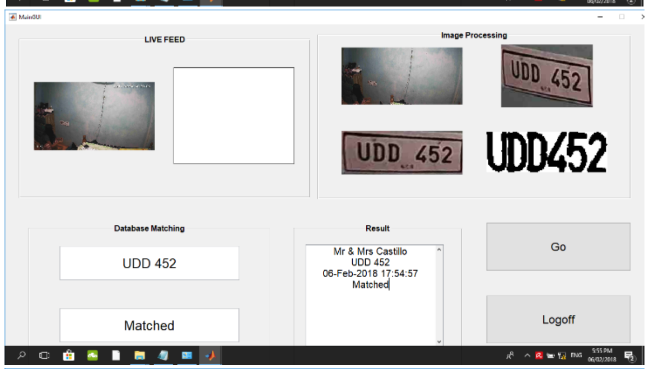

MainGUI
LIVE FEED
Image Processing
Database Matching
FAAK 2709
No match
Result
NA
FAAK 2709
06-Feb-2018 17:33:51
No match in the Database
Go
Logoff

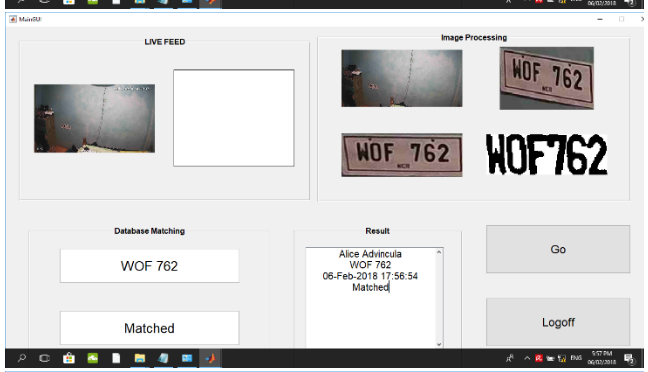

MainGUI
LIVE FEED
Image Processing
CGU755
Database Matching
CGU 755
Matched
Result
Ar Doctolero
CGU 755
06-Feb-2018 17:36:46
Matched
Go
Logoff

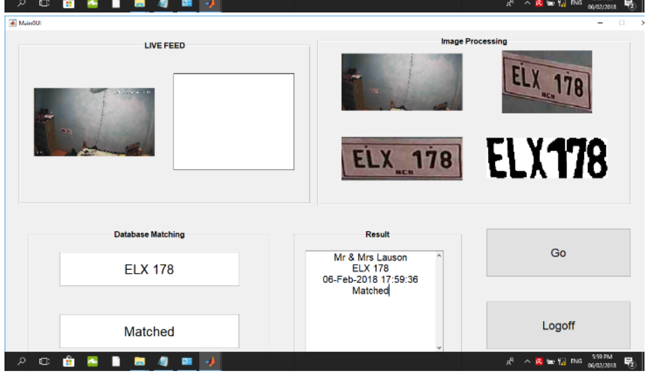

MainGUI
LIVE FEED
Image Processing
Database Matching
ATA 8504
Matched
Result
Ms Francisco
ATA 8504
06-Feb-2018 17:39:11
Matched
Go
Logoff

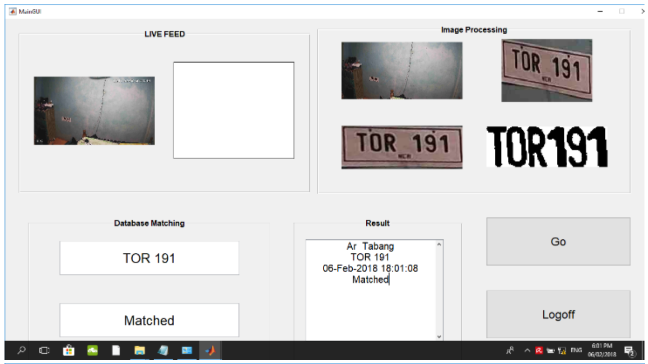

LIVE FEED
Image Processing
PIZ 923
Database Matching
PIZ 923
Matched
Result
Ms Meg Aquino
PIZ 923
06-Feb-2018 17:47:58
Matched
Go
Logoff

LIVE FEED
Image Processing
CKX 980
Database Matching
CKX 980
Matched
Result
Atty Christopher Morten
CKX 980
06-Feb-2018 17:49:58
Matched
Go
Logoff

LIVE FEED
Image Processing
CHH 418
CHH418
Database Matching
CHH 418
Matched
Result
Sarina Pacleb
CHH 418
06-Feb-2018 17:52:28
Matched
Go
Logoff

LIVE FEED
Image Processing
XPK 774
Database Matching
XPK 774
Matched
Result
Gidel Agron
XPK 774
06-Feb-2018 17:53:13
Matched
Go
Logoff

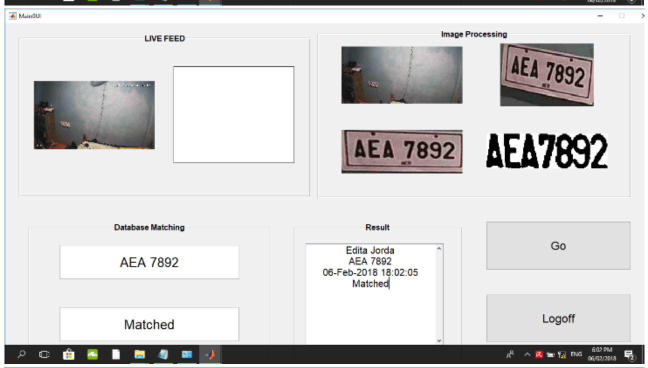

LIVE FEED
Image Processing
UQZ 488
UQZ488
Database Matching
UQZ 488
Matched
Result
Ms Francisco
UQZ 488
06-Feb-2018 17:54:21
Matched
Go
Logoff

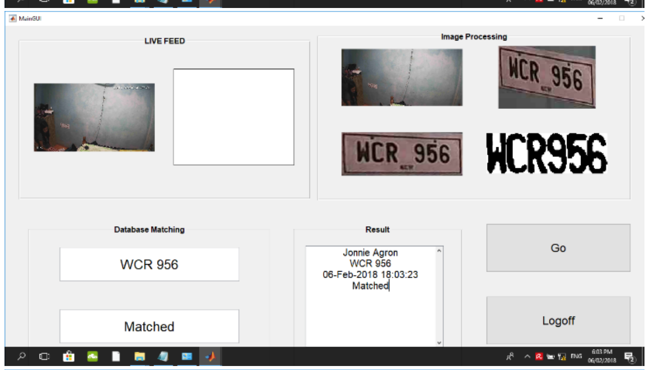

LIVE FEED
Image Processing
UDD 452
UDD452
Database Matching
UDD 452
Matched
Result
Mr & Mrs Castillo
UDD 452
06-Feb-2018 17:54:57
Matched
Go
Logoff

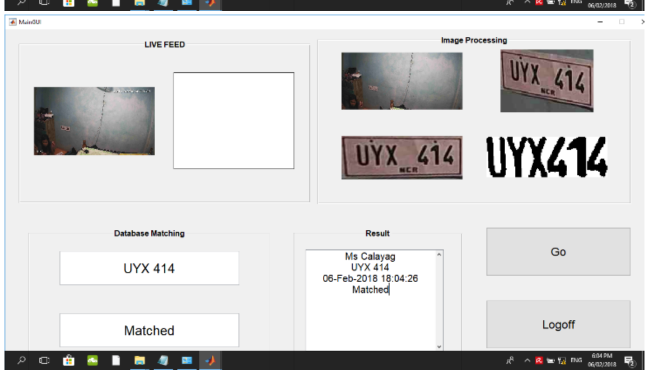

LIVE FEED
Image Processing
WOF 762
WOF762
Database Matching
WOF 762
Matched
Result
Alice Advincula
WOF 762
06-Feb-2018 17:56:54
Matched
Go
Logoff

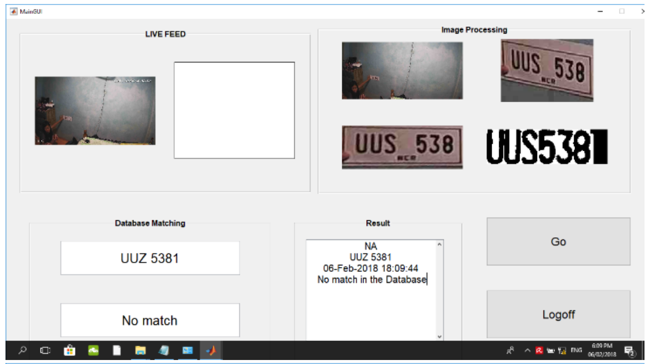

LIVE FEED
Image Processing
ELX 178
ELX178
Database Matching
ELX 178
Matched
Result
Mr & Mrs Lauson
ELX 178
06-Feb-2018 17:59:36
Matched
Go
Logoff

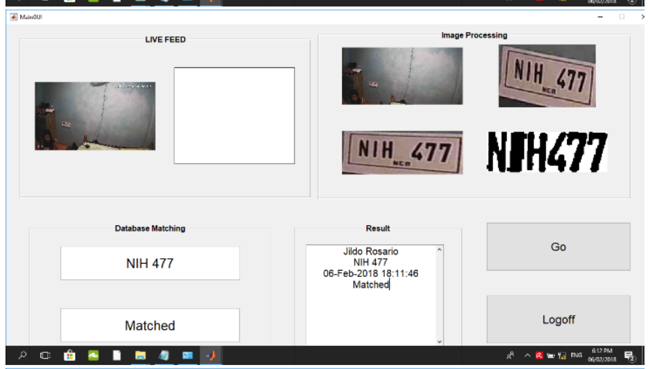

LIVE FEED
Image Processing
TOR 191
TOR191
Database Matching
TOR 191
Matched
Result
Ar Tabang
TOR 191
06-Feb-2018 18:01:08
Matched
Go
Logoff

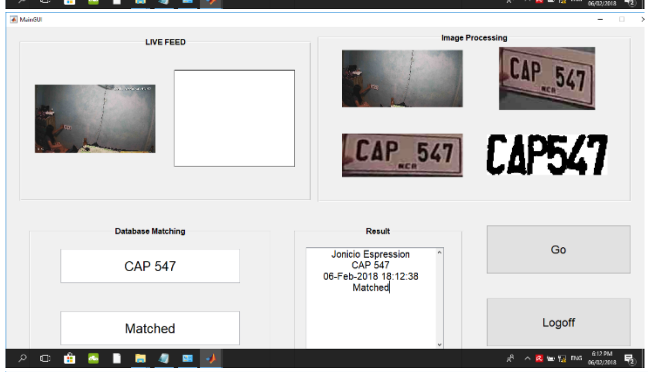

LIVE FEED
Image Processing
AEA 7892
AEA7892
Database Matching
AEA 7892
Matched
Result
Edita Jorda
AEA 7892
06-Feb-2018 18:02:05
Matched
Go
Logoff

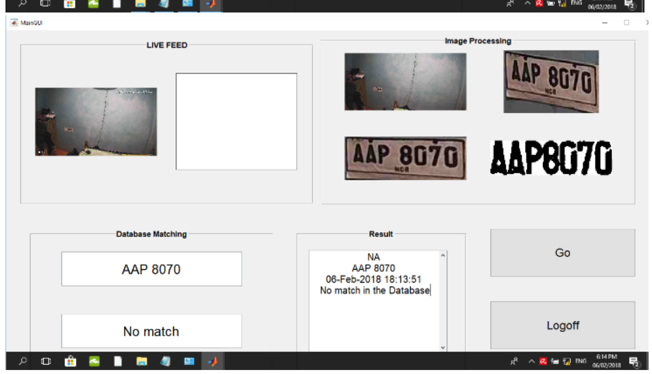

LIVE FEED
Image Processing
WCR 956
WCR956
Database Matching
WCR 956
Matched
Result
Jonnie Agron
WCR 956
06-Feb-2018 18:03:23
Matched
Go
Logoff

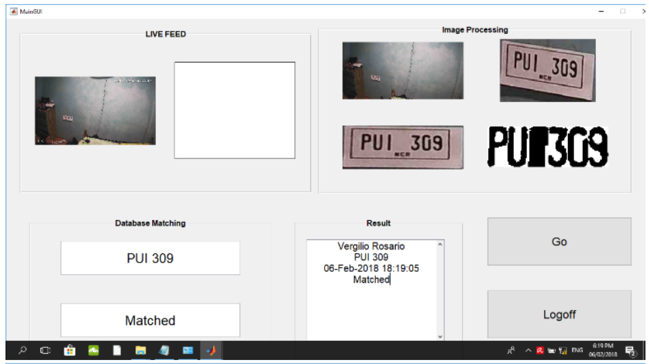

LIVE FEED
Image Processing
UYX 414
UYX414
Database Matching
UYX 414
Matched
Result
Ms Calayag
UYX 414
06-Feb-2018 18:04:26
Matched
Go
Logoff

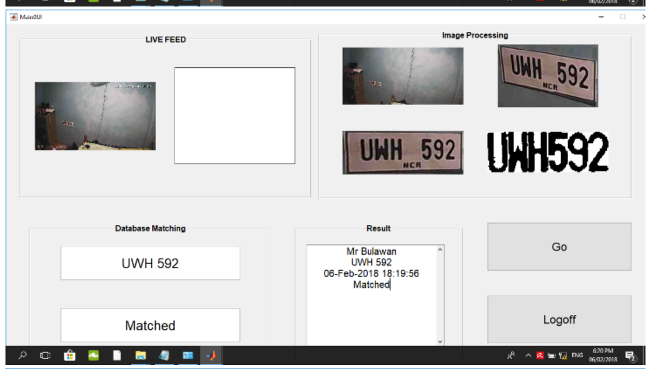

LIVE FEED
Image Processing
UUS 538
UUS5381
Database Matching
UUZ 5381
No match
Result
NA
UUZ 5381
06-Feb-2018 18:09:44
No match in the Database
Go
Logoff

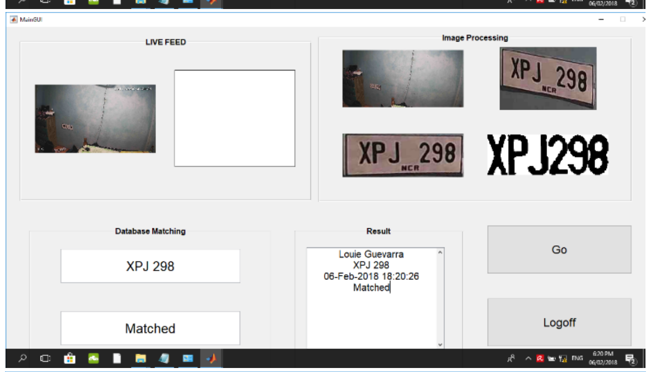

LIVE FEED
Image Processing
NIH 477
Database Matching
NIH 477
Matched
Result
Jildo Rosario
NIH 477
06-Feb-2018 18:11:46
Matched
Go
Logoff

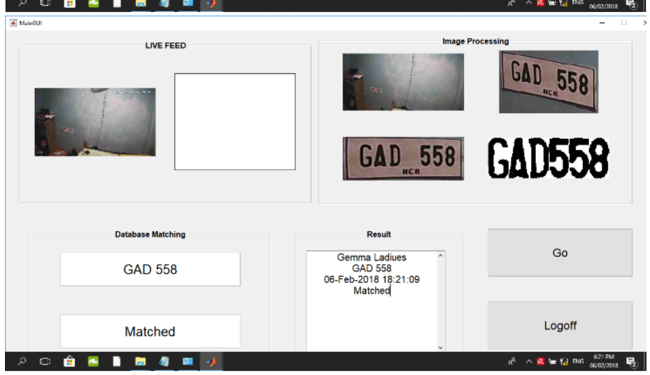

LIVE FEED
Image Processing
CAP 547
CAP547
Database Matching
CAP 547
Matched
Result
Jonicio Espression
CAP 547
06-Feb-2018 18:12:38
Matched
Go
Logoff

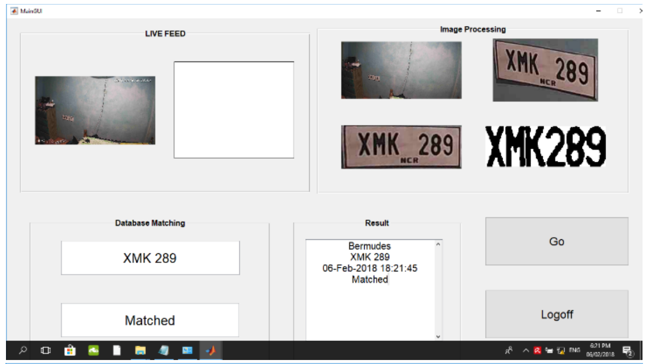

LIVE FEED
Image Processing
AAP 8070
AAP8070
Database Matching
AAP 8070
No match
Result
NA
AAP 8070
06-Feb-2018 18:13:51
No match in the Database
Go
Logoff

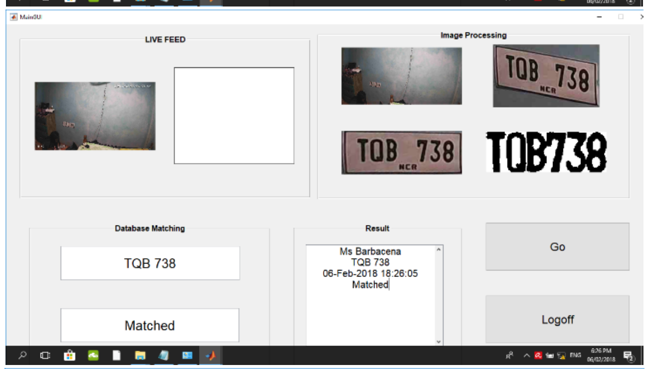

LIVE FEED
Image Processing
Database Matching
PUI 309
Matched
Result
Vergilio Rosario
PUI 309
06-Feb-2018 18:19:05
Matched
Go
Logoff

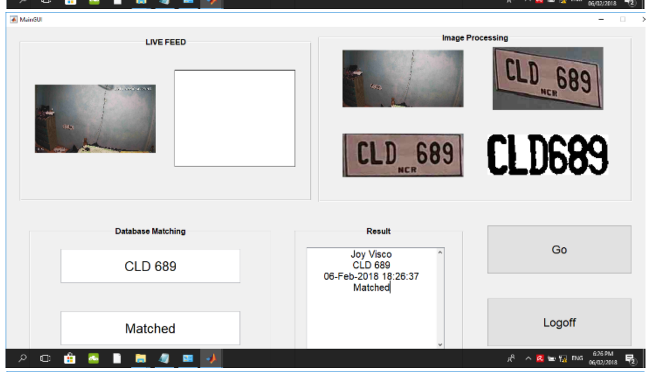

LIVE FEED
Image Processing
Database Matching
UWH 592
Matched
Result
Mr Bulawan
UWH 592
06-Feb-2018 18:19:56
Matched
Go
Logoff

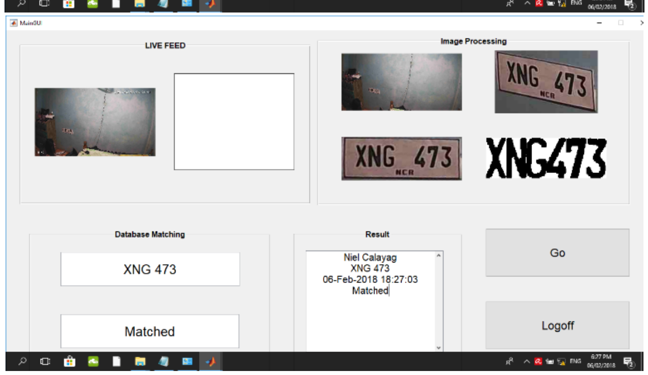

LIVE FEED
Image Processing
Database Matching
XPJ 298
Matched
Result
Louie Guevarra
XPJ 298
06-Feb-2018 18:20:26
Matched
Go
Logoff

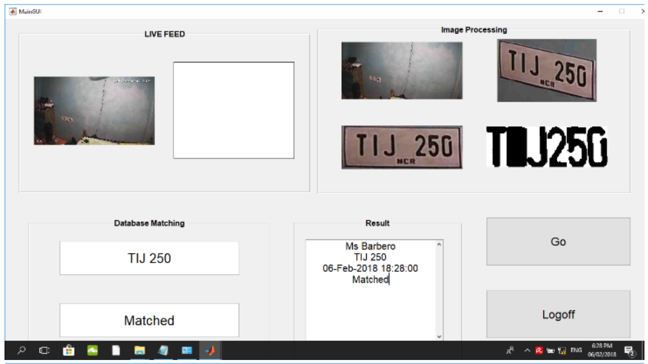

LIVE FEED
Image Processing
Database Matching
GAD 558
Matched
Result
Gemma Ladiues
GAD 558
06-Feb-2018 18:21:09
Matched
Go
Logoff

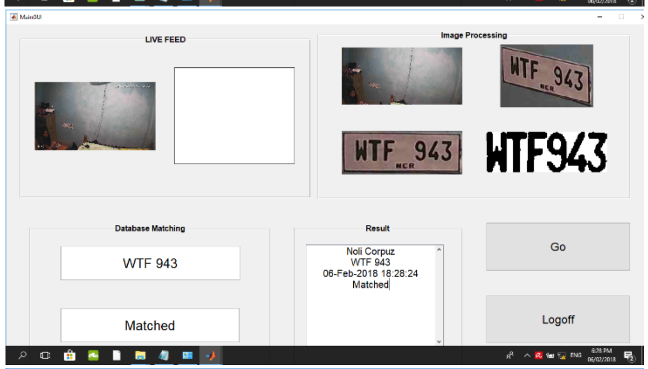

LIVE FEED
Image Processing
XMK 289
XMK 289
XMK289
Database Matching
XMK 289
Matched
Result
Bermudes
XMK 289
06-Feb-2018 18:21:45
Matched
Go
Logoff

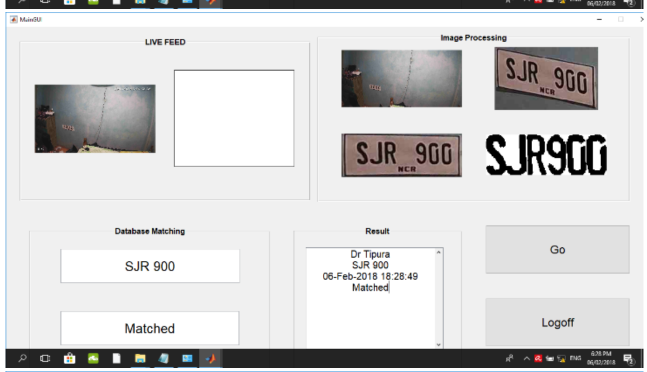

LIVE FEED
Image Processing
TQB 738
TQB 738
TQB738
Database Matching
TQB 738
Matched
Result
Ms Barbacena
TQB 738
06-Feb-2018 18:26:05
Matched
Go
Logoff

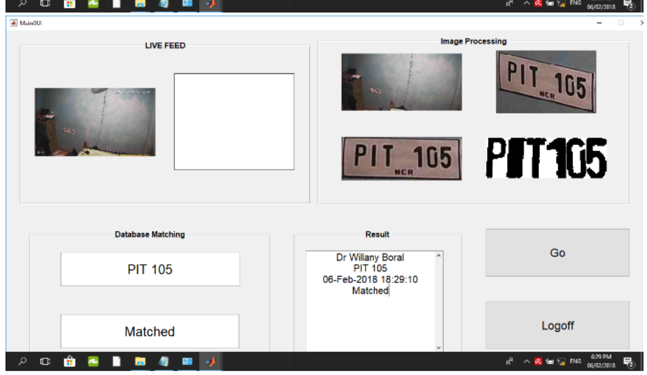

LIVE FEED
Image Processing
CLD 689
CLD 689
CLD689
Database Matching
CLD 689
Matched
Result
Joy Visco
CLD 689
06-Feb-2018 18:26:37
Matched
Go
Logoff

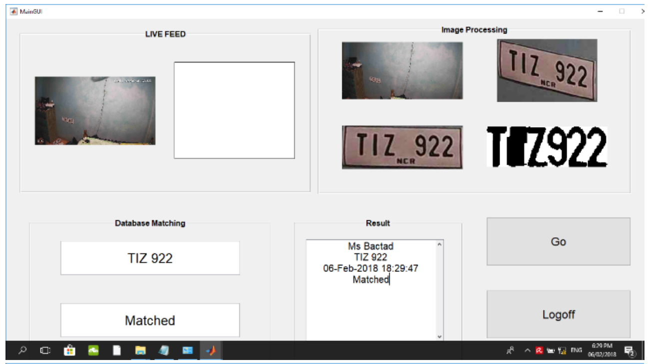

LIVE FEED
Image Processing
XNG 473
XNG 473
XNG473
Database Matching
XNG 473
Matched
Result
Niel Calayag
XNG 473
06-Feb-2018 18:27:03
Matched
Go
Logoff

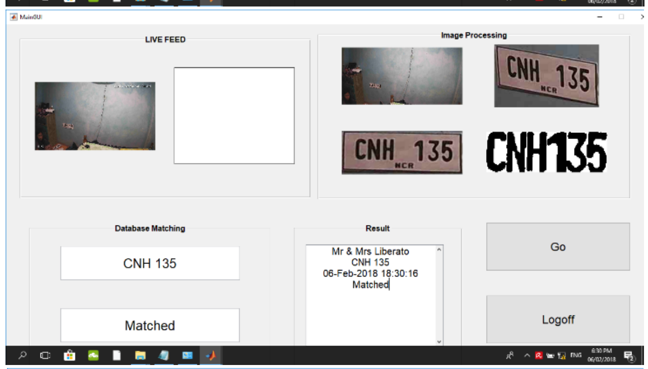

LIVE FEED
Image Processing
TIJ 250
Database Matching
TIJ 250
Matched
Result
Ms Barbero
TIJ 250
06-Feb-2018 18:28:00
Matched
Go
Logoff

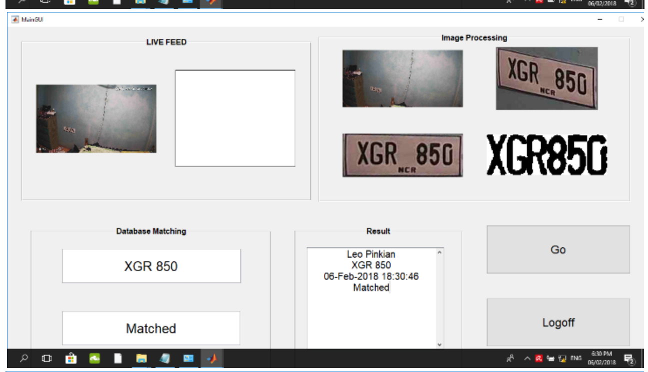

LIVE FEED
Image Processing
WTF 943
Database Matching
WTF 943
Matched
Result
Noli Corpuz
WTF 943
06-Feb-2018 18:28:24
Matched
Go
Logoff

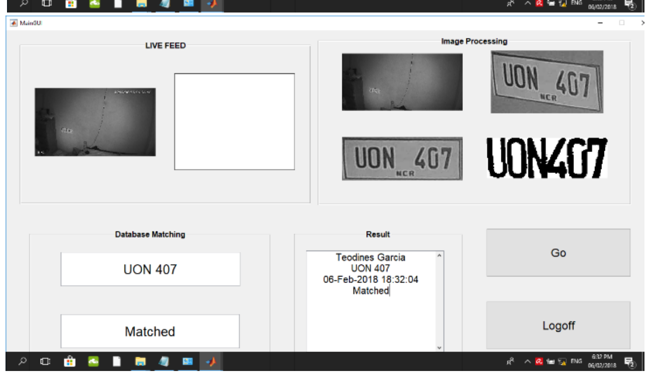

LIVE FEED
Image Processing
SJR 900
Database Matching
SJR 900
Matched
Result
Dr Tipura
SJR 900
06-Feb-2018 18:28:49
Matched
Go
Logoff

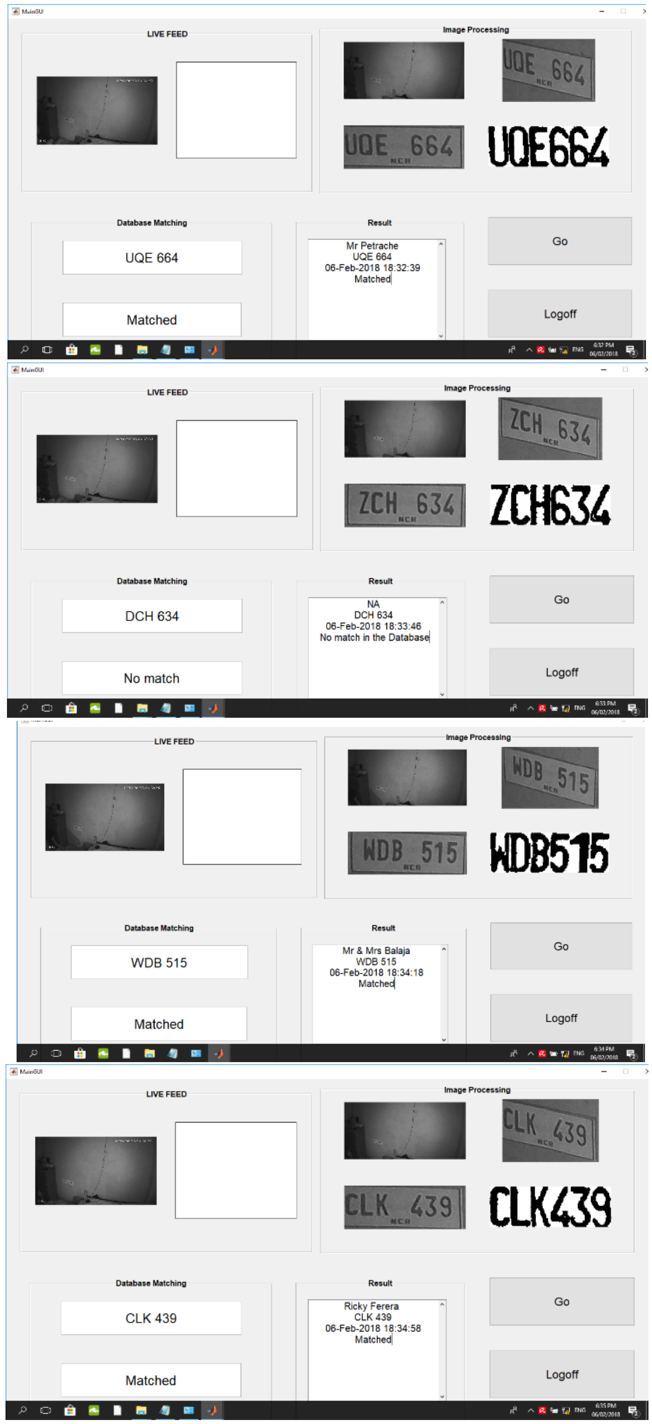

LIVE FEED
Image Processing
PIT 105
Database Matching
PIT 105
Matched
Result
Dr Willany Boral
PIT 105
06-Feb-2018 18:29:10
Matched
Go
Logoff

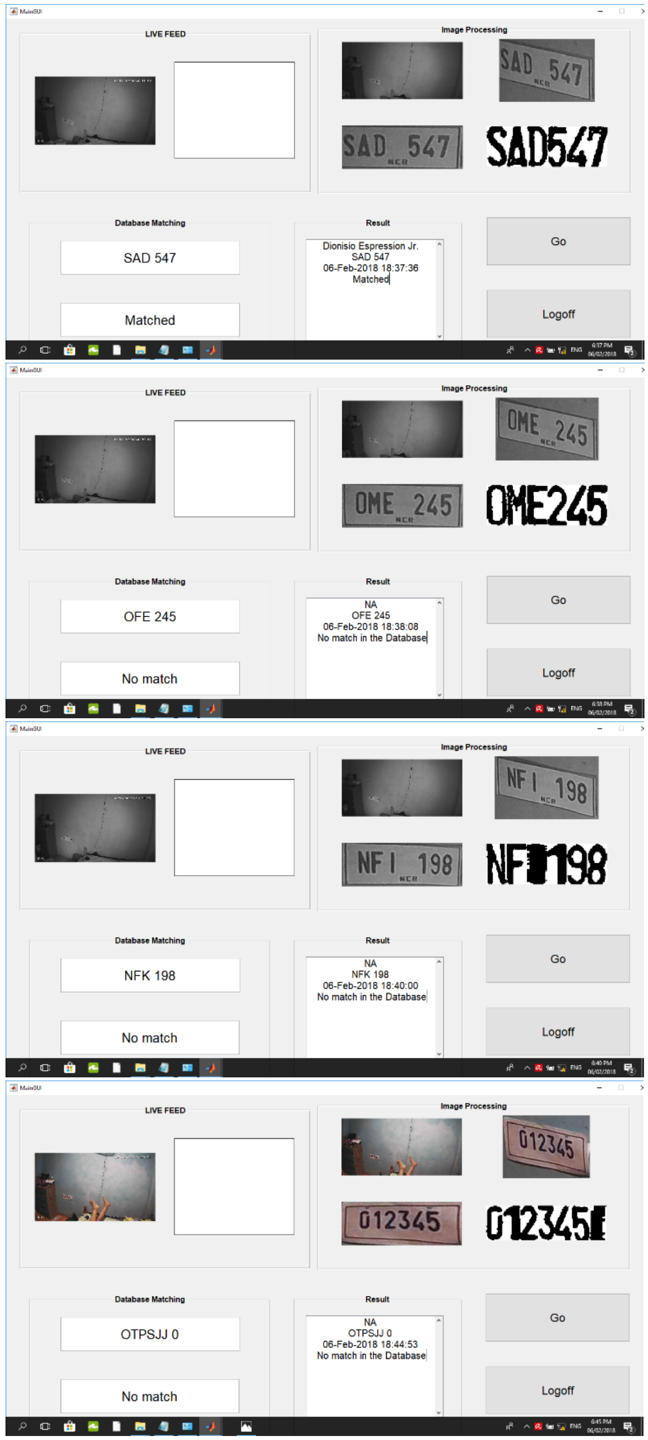

LIVE FEED
Image Processing
TIZ 922
Database Matching
TIZ 922
Matched
Result
Ms Bactad
TIZ 922
06-Feb-2018 18:29:47
Matched
Go
Logoff

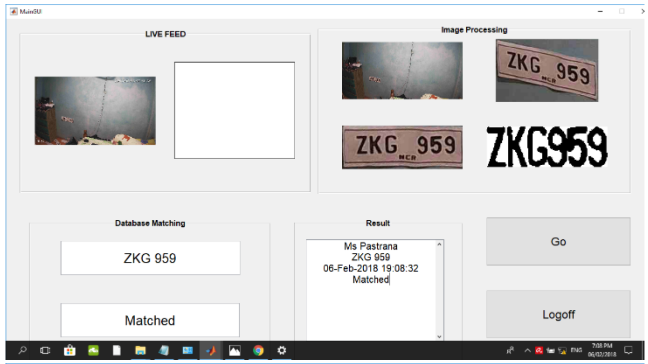

LIVE FEED
Image Processing
CNH 135
Database Matching
CNH 135
Matched
Result
Mr & Mrs Liberato
CNH 135
06-Feb-2018 18:30:16
Matched
Go
Logoff

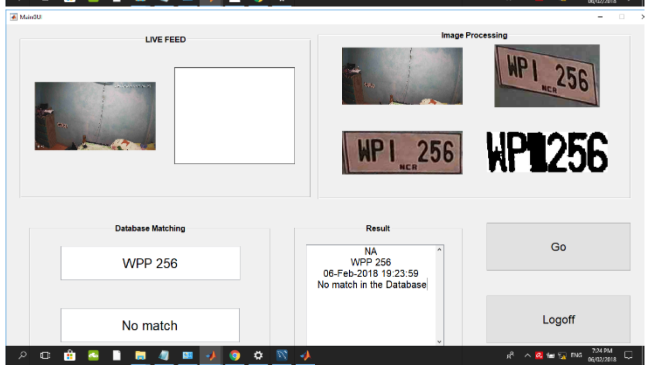

LIVE FEED
Image Processing
XGR 850
Database Matching
XGR 850
Matched
Result
Leo Pinkian
XGR 850
06-Feb-2018 18:30:46
Matched
Go
Logoff


LIVE FEED
Image Processing
UON 407
Database Matching
UON 407
Matched
Result
Teodines Garcia
UON 407
06-Feb-2018 18:32:04
Matched
Go
Logoff


LIVE FEED
Image Processing
UQE664
Database Matching
UQE 664
Matched
Result
Mr Petrache
UQE 664
06-Feb-2018 18:32:39
Matched
Go
Logoff
LIVE FEED
Image Processing
ZCH634
Database Matching
DCH 634
No match
Result
NA
DCH 634
06-Feb-2018 18:33:46
No match in the Database
Go
Logoff
LIVE FEED
Image Processing
WDB515
Database Matching
WDB 515
Matched
Result
Mr & Mrs Balaja
WDB 515
06-Feb-2018 18:34:18
Matched
Go
Logoff
LIVE FEED
Image Processing
CLK439
Database Matching
CLK 439
Matched
Result
Ricky Ferera
CLK 439
06-Feb-2018 18:34:58
Matched
Go
Logoff


LIVE FEED
Image Processing
SAD547
Database Matching
SAD 547
Matched
Result
Dionisio Espression Jr.
SAD 547
06-Feb-2018 18:37:36
Matched
Go
Logoff
LIVE FEED
Image Processing
OME245
Database Matching
OFE 245
No match
Result
NA
OFE 245
06-Feb-2018 18:38:08
No match in the Database
Go
Logoff
LIVE FEED
Image Processing
Database Matching
NFK 198
No match
Result
NA
NFK 198
06-Feb-2018 18:40:00
No match in the Database
Go
Logoff
LIVE FEED
Image Processing
012345
Database Matching
OTPSJJ 0
No match
Result
NA
OTPSJJ 0
06-Feb-2018 18:44:53
No match in the Database
Go
Logoff

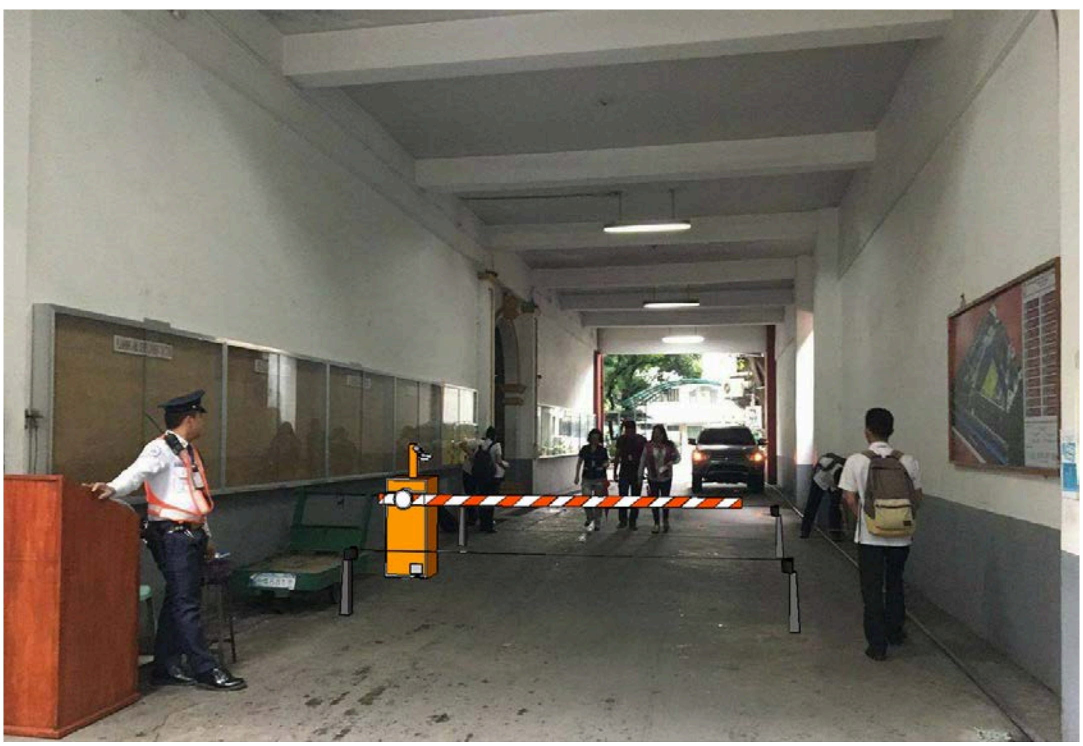

LIVE FEED
Image Processing
ZKG 959
ZKG 959
ZKG959
Database Matching
ZKG 959
Matched
Result
Ms Pastrana
ZKG 959
06-Feb-2018 19:08:32
Matched
Go
Logoff


LIVE FEED
Image Processing
WPI 256
WPI 256
Database Matching
WPP 256
No match
Result
NA
WPP 256
06-Feb-2018 19:23:59
No match in the Database
Go
Logoff

APPENDIX E

Documentation

Consultation with the University Head Security



Proposed deployment layout

The proponents after topic defense

Project Presentation for the Client

Site Surveying

Demonstration of CCTV functionalities

System's Circuit Development

Software Testing

Making of the Barrier's Prototype

Construction of the Car Barrier

Testing of the Car barrier’s mechanism

Installation of the Car barrier

Electrical Wiring of the Project

Installed Project at Gate 1 of TUP - M

APPENDIX F

User's Manual

Development of License Plate Recognition (LPR) System for Automatic Car Barrier Using Image Processing And Artificial Neural Network

autoPLAKA security solutions

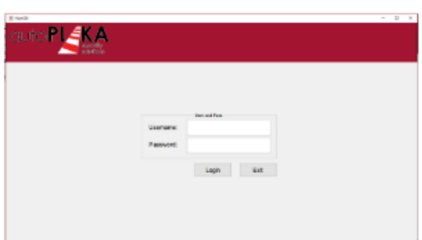
USER'S MANUAL

Adviser:
Engr. Timothy M. Amado

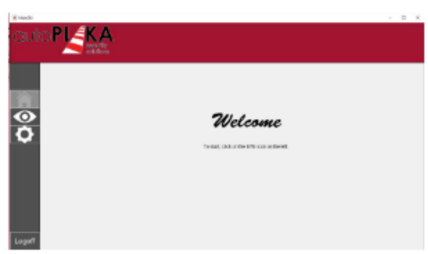
Proponents:
Alacapa, Maria Angelica P.
Calangian, Adriane Roy F.
De Guzman, Romel A.
Merin, Emmanuel Joshua C.
Minguez, Grason B.

Logging-in in the System

1. In the login screen, you must put the username and password to access the system. The default username is "admin" and the default password is "admin123".
2. Then press Login to enter the system.
3. If you want to exit, just press the Exit button and it will terminate the program.

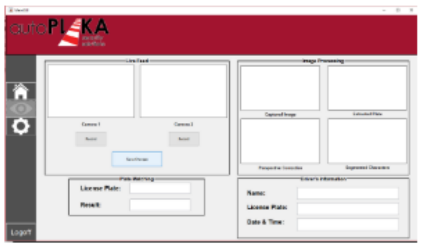


After entering the correct username and password, it will take you to a page with a greeting and instructing the user to click the eye button on the left to start operation.



After clicking the eye button, this page will appear. Here you can see that there are four panels, namely: Live Feed, Image Processing, Plate Matching and Drivers' Information.

In the live feed, there are 2 cameras that will be used for the system. The buttons under the cameras that has a name "Record" is a button that allows the user to record videos of the camera feed.



Recording videos from the live feed and starting program.

1. To do this, all you have to do is click the Record button and click save changes.
2. After this, the camera will start and will record the feed.
3. If you don't want to record videos of the feed just leave everything as is and click save changes. By doing this, the camera feed will start without recording.

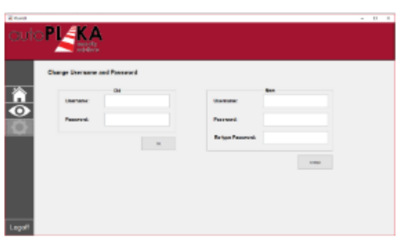
Now, the system is ready for operation. The image taken by the camera will go to the Image Processing stage. The panel of Image Processing will show the different outputs from the different stages of Image Processing up to Character Segmentation.

The next panel is the Plate Matching panel. This is where the characters are being recognized and matched to the database.

The last panel is the Drivers' Information panel. It is where the necessary details of the driver are shown and also the time when it got in.

Changing Username and Password.

1. Enter your old username and password.
2. Then, press ok.
3. Enter your new username and your new password.
4. Then, press Change.



APPENDIX G

Proponents' Profile

MARIA ANGELICA PALOMO ALACAPA

BLK 4 LOT 8 VERAVILLE ALEGRIA, TALON IV, LAS PIÑAS CITY, 1747
(+63)9268033067/ (+63)929765126, gelalacapa@gmail.com

CAREER OBJECTIVE

To obtain training in a company where I can develop my knowledge, enhance my skills in the field of Electronics especially microelectronics and which enables me to grow in support for potential career opportunities.

EDUCATIONAL BACKGROUND

Tertiary **TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES – Manila**
Ayala Blvd., Ermita, Manila
Bachelor of Science in Electronics Engineering
June 2013 - Present

Secondary **LAS PINAS EAST NATIONAL HIGHSCHOOL – Talon Village Annex**
Rose of Heaven Dr., Talon IV, Talon Village, Las Piñas City
June 2009 – March 2013

Primary **Talon Elementary School**
San Antonio Valley Road, Talon Uno, Las Pinas City
June 2003 – March 2009

AFFILIATION

2016 – Present	**Institute of Electronics Engineers of the Philippines Inc.** Member, Membership No. 16-05550
2016 – Present	**Organization of Electronics Engineering Students (OECES)** Vice President for Internal Affairs
2015 – 2016	**Organization of Electronics Engineering Students (OECES)** Publication and Certification, Junior Committee
2015 – Present	**Institute of Electronics Engineers of the Philippines – Manila Student Chapter (MSC)** Member
2012 – 2013	**Supreme Student Government** President

ACCOMPLISHMENT

- **First Honorable Mention** 2013
- **Vice President Leadership Award** 2013
- **Gerry Roxas Leadership Award** 2013
- **Salutatorian** 2009
- **Electronics Technician**
  License No. 0012262
  November 2016

SEMINARS AND CONFERENCES ATTENDED

- Synergy: Electrical And Electronics Engineering Summit 2017 (November 21, 2017)
- Engineering Leadership Conference 2018 (January 27,2018)
- Introduction to Data Science (January 22, 2018)
- CAD Workshop (February 3, 2018)
- Data Science Congress 2018 (February 25, 2018)

I hereby certify that the above information is true and correct to the very best of my knowledge and belie

Maria Angelica P. Alacapa

PERSONAL INFORMATION

AGE: 20

DATE OF BIRTH: February 13, 1997

PLACE OF BIRTH: Las Piñas City

NATIONALITY: Filipino

CIVIL STATUS: Single

RELIGION: Catholic

HEIGHT: 5"

WEIGHT: 46 kgs

LANGUAGES SPOKEN: Filipino, English

QUALIFICATION

- Computer literate particularly in Microsoft Office
- Above average communication skills
- Can work independently as well as collaboratively
- Flexible and can multi-task
- Basic Knowledge in Arduino and Matlab

CHARACTER REFERENCE

Engr. Lean Tolentino
Professor, ECE Department
TUP College of Engineering
09958925845

Lerma A. Laguitan
Teacher
Las Pinas East National High School-TVA
09176256215

ADRIANE ROY F. CALANGIAN

441 Banaba St., Cembo, Makati City
•09397719220 • roydriane@gmail.com

CAREER OBJECTIVE

To acquire an engineering position where skills and knowledge will be applied and developed particularly on microelectronics; and new practical experiences for the foundation and outlook of employment as an electronics engineer.

EDUCATIONAL BACKGROUND

Tertiary **TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES – Manila**
Ayala Blvd., Ermita, Manila
Bachelor of Science in Electronics Engineering
June 2013 - Present

Secondary **PITOGO HIGH SCHOOL**
Negros St., Pitogo, Makati City
June 2009 – March 2013

Primary **CEMBO ELEMENTARY SCHOOL**
Acacia St., Cembo, Makati City
June 2003 – March 2009

AFFILIATION

2016 – Present **Institute of Electronics Engineers of the Philippines Inc.**
Member

2016 – Present **Institute of Electronics Engineers of the Philippines – Manila Student Chapter (MSC)**
Membership No. 16-06017

2013 – Present **Organization of Electronics Engineering Students (OECES)**
Member

ACCOMPLISHMENT

- **Fifth Honorable Mention**
  Pitogo High School
  March 2013
- **Electronics Technician**
  License No. 0012258
  October 2016

EXPERIENCE

Internship TRICOM Dynamics, INC. (TDI)
April – May 2017

SEMINARS AND CONFERENCES ATTENDED

- "Expanding Electronics Engineers' Horizon" (July 2013)
- "Bakit ECE? Eh Meron Namang Iba" (March 2014)
- Autodesk Fusion 360 (November 2016)
- Synergy: Electrical And Electronics Engineering Summit 2017 (November 21, 2017)
- Engineering Leadership Conference 2018 (January 27,2018)
- Introduction to Data Science (January 22, 2018)
- CAD Workshop (February 3, 2018)
- Data Science Congress 2018 (February 25, 2018)

PERSONAL INFORMATION

AGE: 20

DATE OF BIRTH: March 17, 1997

PLACE OF BIRTH: Makati City

NATIONALITY: Filipino

CIVIL STATUS: Single

RELIGION: Catholic

HEIGHT: 5' 7"

WEIGHT: 75 kgs

LANGUAGES SPOKEN: Filipino, English

QUALIFICATION

- Computer literate
- Proficient in Microsoft Office Programs
- Basic knowledge in programming using MATLAB
- Capable of designing using Fusion 360 and Cadence Virtuoso
- Exemplary skills by implementing processes for circuit analysis
- Capable of working under extreme pressure and deadlines

CHARACTER REFERENCE

Engr. Lean Tolentino
Professor, ECE Department
TUP College of Engineering
09958925845

Ms Concepcion A. Garcia
Acting Deputy Director and Head
Bangko Sentral ng Pilipinas
09297255212

I hereby certify that the above information is true and correct to the very best of my knowledge and belief.

Adriane Roy F. Calangian

ROMEL A. DE GUZMAN

KAMPI Association, Green Valley San Nicolas III Bacoor, Cavite 4102
•09995844542/09773166525 • romeldeguzman15@gmail.com

CAREER OBJECTIVE

To seek an employment in an organization which will help utilize my skills, broaden my knowledge about my field of study, especially in microelectronics and which will serve as my spring board to move ahead by providing me interesting career opportunities.

EDUCATIONAL BACKGROUND

Tertiary — **TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES – Manila**
Ayala Blvd., Ermita, Manila
Bachelor of Science in Electronics Engineering
June 2013 - Present

Secondary — **LAS PINAS EAST NATIONAL HIGHSCHOOL – Talon Village Annex**
Rose of Heaven Dr., Talon IV, Talon Village, Las Piñas City
June 2009 – March 2013

Primary — **Talon Elementary School**
San Antonio Valley Road, Talon Uno, Las Pinas City
June 2003 – March 2009

AFFILIATION

2016 – Present — **Institute of Electronics Engineers of the Philippines Inc.**
Member

2016 – Present — **Institute of Electronics Engineers of the Philippines – Manila Student Chapter (MSC)**
Member

2013 – Present — **Organization of Electronics Engineering Students (OECES)**
Member

ACCOMPLISHMENT

- **Fourth Honorable Mention**
  LAS PINAS EAST NATIONAL HS
  March 2013
- **Electronics Technician**
  License No. 0012260
  November 2016

SCHOLARSHIP

DOST Science Education Institute RA 7687 SCHOLARSHIP
June 2013 – Present

EXPERIENCE

Internship — **Electronics Security System Corporation (ESSCOR)**
April – May 2017

SEMINARS AND CONFERENCES ATTENDED

- Synergy: Electrical And Electronics Engineering Summit 2017 (November 21, 2017)
- Engineering Leadership Conference 2018 (January 27,2018)
- Introduction to Data Science (January 22, 2018)
- CAD Workshop (February 3, 2018)
- Data Science Congress 2018 (February 25, 2018)

PERSONAL INFORMATION

AGE: 22

DATE OF BIRTH: January 15, 1996

PLACE OF BIRTH: Las Piñas City

NATIONALITY: Filipino

CIVIL STATUS: Single

RELIGION: Catholic

HEIGHT: 5’ 4”

WEIGHT:50 kgs

LANGUAGES SPOKEN: Filipino, English

QUALIFICATION

- Computer Literate
- Ability to work under pressure and to work on a team
- Highly Adaptable and Flexible
- Data entry using Microsoft Word, Excel and PowerPoint
- Basic knowledge in Arduino, Multisim and Matlab
- Basic knowledge in computer aided designing using Autodesk Fusion 360 and Cadence Virtuoso

CHARACTER REFERENCE

Engr. Lean Tolentino
Professor, ECE Department
TUP College of Engineering
09958925845

Antoinette R. Bravo, CPA
Accountant
Concepcion Business Services Inc
09297255212

I hereby certify that the above information is true and correct to the very best of my knowledge and belief.

Romel A. De Guzman

EMMANUEL JOSHUA C. MERIN

76-B Sta. Cecilia St., San Antonio Valley 14, San Isidro, Paranaque City, 1700
•09150761098 • joshuamerin26@gmail.com

CAREER OBJECTIVE

To be part of an institution where my theoretical and practical skills in electronics engineering and related fields will be utilized.

EDUCATIONAL BACKGROUND

Tertiary — **TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES – Manila**
Ayala Blvd., Ermita, Manila
Bachelor of Science in Electronics Engineering
June 2013 - Present

Secondary — **PARANAQUE NATIONAL HIGH SCHOOL**
Kaytalise St., San Dionisio, Paranaque City
June 2009 – March 2013

Primary — **MARY IMMACULATE SCHOOL**
Sta. Natividad St., San Isidro, Paranaque City
June 2003 – March 2009

AFFILIATION

2016 – Present — **Institute of Electronics Engineers of the Philippines Inc.**
Member

2016 – Present — **Institute of Electronics Engineers of the Philippines – Manila Student Chapter (MSC)**
Member

2013 – Present — **Organization of Electronics Engineering Students (OECES)**
Member

ACCOMPLISHMENT

- **Electronics Technician**
  License No. 0012179
  November 2016
- **ECE Awards 2017 – Technological University of the Philippines – Manila**
  Electronics Technician Licensure Examination Passer (October 2016)

EXPERIENCE

Internship — **PLDT, Inc. – Main Distribution Frame Department**
April – May 2017

SEMINARS AND CONFERENCES ATTENDED

- “Expanding Electronics Engineers’ Horizon” (July 2013)
- “Bakit ECE? Eh Meron Namang Iba” (March 2014)
- Synergy: Electrical And Electronics Engineering Summit 2017 (November 21, 2017)
- Introduction to Data Science (January 22,2018)
- Engineering Leadership Conference 2018 (January 27,2018)
- CAD Workshop (February 3, 2018)
- Data Science Congress 2018 (February 25, 2018)

PERSONAL INFORMATION

AGE: 21

DATE OF BIRTH: December 26, 1996

PLACE OF BIRTH: San Juan City, MM

NATIONALITY: Filipino

CIVIL STATUS: Single

RELIGION: Roman Catholic

HEIGHT: 5’ 9”

WEIGHT: 79 kgs

LANGUAGES SPOKEN: Filipino, English

QUALIFICATION

- Problem-solving skills.
- Microelectronic circuit analysis and design skills. (Cadence Virtuoso)
- Telecommunication skills.
- Software development skills. (C, C++, C#, Java, XML and MATLAB)
- Web development skills. (HTML, CSS and Javascript)
- Basic 2D and 3D Design skills. (Autodesk Fusion 360 and Blender)
- Good communication skills.
- Ability to work in harmony with a team.

CHARACTER REFERENCE

Engr. Lean Tolentino
Professor, ECE Department
TUP College of Engineering
09958925845

I hereby certify that the above information is true and correct to the very best of my knowledge and belief.

Emmanuel Joshua C. Merin

GRASON B. MINGUEZ

084 San Pedro St., Seacom Compound, Sucat, Parañaque City
• 09278844401/7767657 • gminguez12@gmail.com

CAREER OBJECTIVE

To work in an environment where I can use and hone my knowledge and skills in the field of Electronics Engineering and to utilize my abilities within a prestigious company and to contribute to its growth and success.

EDUCATIONAL BACKGROUND

Tertiary — **TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES – Manila**
Ayala Blvd., Ermita, Manila
Bachelor of Science in Electronics Engineering
June 2013 – Present

Secondary — **PARAÑAQUE NATIONAL HIGH SCHOOL**
Kay Talise St., Dr. A. Santos Avenue, San Dionisio, Parañaque City
June 2009 – March 2013

Primary — **ST. CYR ACADEMY**
Dr. A Santos Avenue, Parañaque City
June 2003 – March 2009

AFFILIATION

2016 – Present — **Institute of Electronics Engineers of the Philippines Inc.**
Member

2016 – Present — **Institute of Electronics Engineers of the Philippines – Manila Student Chapter (MSC)**
Membership No. 16-07203

2013 – Present — **Organization of Electronics Engineering Students (OECES)**
Member

ACCOMPLISHMENT

- **Electronics Technician**
  License No. 0012177
  November 2016

EXPERIENCE

Internship — **PLDT, Inc. – Main Distribution Frame Department**
April – May 2017

SEMINARS AND CONFERENCES ATTENDED

- "Expanding Electronics Engineers' Horizon" (July 2013)
- "Bakit ECE? Eh Meron Namang Iba" (March 2014)
- Synergy: Electrical And Electronics Engineering Summit 2017 (November 21, 2017)
- Introduction to Data Science (January 22,2018)
- Engineering Leadership Conference 2018 (January 27,2018)
- CAD Workshop (February 3, 2018)
- Data Science Congress 2018 (February 25, 2018)

PERSONAL INFORMATION

AGE: 20

DATE OF BIRTH: June 10, 1997

PLACE OF BIRTH: Parañaque City

NATIONALITY: Filipino

CIVIL STATUS: Single

RELIGION: Aglipayan

HEIGHT: 5' 9"

WEIGHT: 81 kgs

LANGUAGES SPOKEN: Filipino, English

QUALIFICATION

- Good Communication Skills
- Computer Literate (Specializes in MS Word, MS Excel, MS Powerpoint)
- Basic knowledge in JAVA programming language, Arduino, Android Development, MATLAB
- Basic knowledge in Electronics and Troubleshooting
- Can work properly under pressure and on a team
- Basic knowledge in computer aided designing using Autodesk Fusion 360 and Cadence Virtuoso

CHARACTER REFERENCE

Engr. Lean Tolentino
Professor, ECE Department
TUP College of Engineering
09958925845

I hereby certify that the above information is true and correct to the very best of my knowledge and belief.

Grason B. Minguez