明志科技大學
MING CHI UNIVERSITY OF TECHNOLOGY

Department of Electronics Engineering

Master Thesis

The Vehicle Plate Detection Using Multi-views From a Single Fisheye Camera

Graduate Student:

Heru Syah Putra

Advisor:

Dr. Chuang Jan-Chang, Dr. Shu-Lin Hwang

December 2022

Ming Chi University of Technology, Taiwan

Recommendation Letter from the Thesis Advisor

This thesis is by Heru Syah Putra of The Department of Electronic Engineering, Ming Chi University of Technology, Taiwan.

Entitled :
The Vehicle Plate Detection Using Multi-views From a Single Fisheye Camera

Which is written under my supervision, and I agree to propose it for examination.

Advisor: Dr. Chuang-Jan Chang, Shu-Lin Hwang
111/ 00/ 00

Ming Chi University of Technology
Thesis Oral Defense Committee Certification

This thesis is by Heru Syah Putra of The Department of Electronic Engineering, Ming Chi University of Technology, Taiwan.

Entitled :

The Vehicle Plate Detection Using Multi-views From a Single Fisheye Camera

Who is qualified for master degree through the verification of the committee.

Convener of the degree examination committee (Signature)

Committee members

Department Chair or Program Director

/ o/ o (mm/dd/yyyy)

Acknowledgments

Alhamdu lillahi Rabbil ‘Alamin. The praise and gratitude to Allah SWT Almighty for the abundance of his grace and gifts so that the author can complete the thesis with the title: Introduction to The Vehicle Plate Detection Using Multi-View From Single Fisheye Camera. This is to fulfill one of the requirements and complete the study to obtain a Master of Science in Engineering at the Electronics Engineering department at Ming Chi University of Technology, Taiwan.

Appreciation and sincere thanks to my supervisor, Professor Dr. Chuang-Jan Chang, Ming-Chi Omnidirectional Imaging Laboratory (MOIL) has allowed me to conduct research and has always provided invaluable guidance from the first time the author came to Taiwan until throughout this research. His dynamism, vision, sincerity and motivation inspire me. May Allah SWT always bestow grace, health, bounty, and blessings in this world and the hereafter for the kindness given to the author. The author also realizes that without the help, advice, and guidance of those who have helped, the author would not have been able to complete this thesis. For that, the author would like to thanks:

1. The writer's parents (Mr. Aburudin and Mrs. Hartuti) and sisters (Elva Nirma Gustina and Henny Puspita Dwi) have given support and prayers to the author.
2. Professor Dr. Jin-Wei Liang has provided the opportunity and financial support for the author to complete his education.
3. Professor Dr. Ya-Fen Wu, as the Head of the Electronics Engineering program, has provided the opportunity to direct and motivate the author in completing this thesis.
4. Professor Dr. Shu-Lin Hwang is a supervisor who is full of ability and willingness to spend quite a lot of time guiding me in completing this thesis.
5. All lecturers and staff employees, as well as the entire academic community of Ming Chi University of Technology Electronics Engineering department.
6. All members (Haryanto, MS., Aji Tri Pamungkas, Adam, Ada, Slender, Jimmy, Andy, and Huan) of the Ming-Chi Omnidirectional Imaging Laboratory. Who always helped me in every way while studying at Ming Chi University of Technology.
7. All members of the Indonesian Student Association (PPI MCUT) are part of my family and always provide support and encouragement.

Finally, the author would like to thank all parties, and the author also realizes that this thesis is still far from perfect in writing. Therefore, the authors ask for suggestions and constructive criticism for perfection, which may be useful for all of us. Aamiin!

Heru Syah Putra
New Taipei, December 2022

Abstract

Abstract: In this study, we built a vehicle traffic monitoring system using a fisheye camera with a field of view (FoV) of 180–220 degrees. This is taken from the many cases that allow multiple cameras to obtain license plate information at every angle according to information needs. Proposing an algorithm from Ming-Chi Omnidirectional Imaging and Laboratory (Moil) that can provide a solution for this case, which converts the original fisheye image into different multi-views depending on the target's needs, is one of the advantages of our proposed method. The resulting image is a rectangular display by adopting the YOLO4 algorithm, which is believed to be able to contribute to the detection of vehicle plate traffic. In this study, experiments were carried out in two places, first in a laboratory environment with a distance of 1-3 meters, to detect fake vehicle plates with prediction accuracy parameters on the front view (inside), which averaged 81.46% prediction accuracy. On the rear view (outside). by 80.35%. After that, experiments were carried out outside the laboratory with a detection distance of 1–30 meters, and the results proved that when the algorithm detects the vehicle plate, the image quality, such as light, the speed of the vehicle driving, and the vehicle plate size, do not affect the recognition of the resulting vehicle plate. These two experiments were conducted to test the YOLO4 algorithm running under a lot of light outside the environment using a fisheye camera.

Index-Terms: Monitoring system, Fisheye camera, Multi-views, The plate detection

Table of Contents

List of Figures

List of Table

List of Abbreviations

FoV	-	Field of View
RCNN	-	Re-Current Neural Network
YOLO	-	You only look once
SSD	-	Single-Shot Multi-box Detection
MAP	-	Mean Average Precision
CGA	-	Conformal Geometry Algebra
WAC	-	Wide-Angle Cameras
MOIL	-	Ming-Chi Omnidirectional Imaging Laboratory
CamOdoCal	-	Camera Odometry Calibration
RPi	-	Raspberry Pi
ARM	-	Advanced RISC Machines
RAM	-	Random-Access Memory
GPIO	-	General-Purpose Input/Output
PC	-	Personal Computer
OS	-	Operating System
GUI	-	Graphical User Interfaces
OOP	-	Object-Oriented Programming
IDE	-	Integrated Development Environment
CUDA	-	Compute Unified Device Architecture
APIs	-	Application Programming Interfaces
VCR	-	Video Cassette Recorder
DVR	-	Digital Video Recorder
CNN	-	Convolution Neural Network
FPN	-	Feature Pyramid with the Network
PANet	-	Path Aggregation Network
UML	-	Unified Modelling Language
ERD	-	Entity Relationship Diagram

Chapter I Introduction

1.1 Research Motivation

Nowadays, with the development of technology, computer vision automates repetitive tasks that require human capabilities for continuous monitoring and timely decision-making on the road. One of the real cases is vehicular traffic on highways, parking areas, and many others. Various applications have been developed to meet the needs of these technologies, such as intelligent technology in detecting, identifying, and tracking various objects of interest [1]. Advances in the development of various object-observed vision applications allow the use of multiple sensors, such as sight cameras installed in demanding environments, such as intelligent technology infrastructure, visual sensor networks, intelligent surveillance, security cameras, and plate recognition in vehicles with complex depth series to address computer vision problems. To enter the intelligent technology environment that detects and identifies large numbers in an area [2] [3].

As one means of transportation that is often used, of course, with the number of vehicles that has skyrocketed in recent years in large and metropolitan cities, the use of cameras on every corner of the area requires smart technology called area monitoring technology that needs to get vehicle information from cameras. Intelligent with computer vision tries to recognize license plate detection either in a combination of vehicles or in bulk or large quantities [4]. A manufacturing system with a detection model has been developed to reduce the burden of humans as task control operators on time. This causes a large burden on controlling vehicles, such as the entry and exit of vehicles, and securing vehicles in the parking area is the only way to carry out supervision by detecting, identifying, and explicitly recognizing various types of vehicles [5] and knowing the identity of number plates in the area. High density using multiple cameras to obtain the license plate detection information of the supervised vehicle [6], a surveillance system or monitoring of an area using video cameras installed in certain places, assembled into a closed network and can be monitored from a control room.

The use of a camera is the main alternative to obtain information by introducing a wide-angle fisheye lens camera is one method used to obtain a wide field of view as an alternative to cases on streets, parking areas, or corners that require a lot of installed surveillance. The number of surveillance cameras in each corner of the target using a fisheye camera model with a projection with a nonlinear geometric imaging structure, the fisheye camera image experiences several barrel distortions, including descending distortion, light prism distortion, and radial distortion, which require calibration to determine the camera parameters used. Therefore, camera calibration using concentric and symmetrical patterns helps parameterize the camera's fisheye lens by ensuring optical parameters consisting of the main point, focal length constant, and lens projection function for the development of the system that will be built by *Chuang-Jan Chang* [7].

Based on the background drawn from the cases [2], or issues with smart technology when the vehicle plate detection system is one of the implementations that will be built by utilizing a fisheye-shaped camera that will be used as multi-views to obtain multiple angles of the desired target side with various views based on needs, using the above-described method. Apply to change fisheye images into multi-views with rectilinear images that can monitor the coverage area that has been determined based on input from parameters with pixel coordinates, making the image not distorted so that by using this method, you can see from every angle without having to use multiple cameras. By adopting this method, Yolov4 to detect possible vehicle objects can help to getting information, which makes this task the main target with a security system, which is an inexpensive alternative way without having to spend a lot of money with the use of multiple cameras to build a surveillance system in and out of vehicles as well as security-prone areas that make a good alternative system.

1.2 Thesis Objectives

The thesis aims to study the application of the fisheye camera in the field of monitoring systems on vehicle plates by utilizing the coverage of the Field of View (FoV) camera from the Fisheye camera.

1. Use one fisheye camera to capture two different directions at equal Fields of View (FoV) coverage.
2. To investigate the performance of the fisheye camera as the main sensor for getting more information.
3. Implement the Yolov4 method to detect, identify, numbers, and letters on vehicle license plates.
4. The implementation of the Moildev library to create multi-views that can provide an alternative as a substitute for the many cameras on every corner of the area.

1.3 Related Work

The various vehicles are an inseparable part of daily life to facilitate the management of traffic vehicles in several smart technologies developed for traveling using cameras at every angle. Sistem tracking plate recognition camera is used as a vision to obtain information generated from a large number of cameras, such as vehicle detection and tracking, which is useful as an alternative to digital traffic crossing by proposing YOLOv4 as a vehicle plate detector with a character recognition kind of feature extraction method that is from rough to smooth [8]. YOLO was originally designed to provide fast detection speeds but with lower speed accuracy. Although YOLOv2 improves the speed and accuracy of objects, we propose a two-stage architecture based on YOLOv4 for vehicle insert detection and identification of vinyl records in a single channel. We collect data for different types of vehicle plates. License character recognition is a kind of feature extraction method called rough to smooth. The framework system is designed to detect vinyl records from different directions, which can improve performance when the image recognition plate is obtained from the camera information [9] [10].

Axxon Traffic Control Server
Distributed Information System
Federal Information Systems Regional databases
Data Processing Center

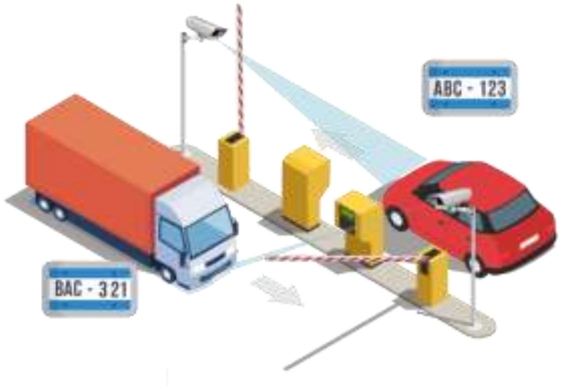


Figure 1. 1 Monitoring system on the road [9] Figure 1. 2 The vehicle plate detection [10]

The introduction of related plates that target similar plate licenses and share methods. Detection of the plate recognition layout is included in the system's structure. The object recognition methods RCNN, YOLOv4 and SSD are faster for vehicle recognition. Their findings [11] include YOLOv4 performance in F1 scores, accuracy, Remember, and MAP. Compared to previous studies, this study investigated YOLOv4 in license plate recognition. We used YOLOv4 to identify vehicle plate disks that showed that performance. Proposing research with a framework system designed to utilize one fisheye camera with a large area can provide another alternative to the many cameras used so that at low cost, it can improve the performance of the digital traffic system as a monitoring camera [12].

1.4 Contribution of Thesis

From the research that will be carried out in making this thesis, there are several main contributions to be achieved, including:

1. Using a Fisheye camera provides a wider Field of View (FoV) coverage and can be used to create multi-views.
2. Utilizing the wide angle generated from the fisheye camera, which makes it the main source of information.
3. Provides high accuracy in vehicle plate detection using a fisheye camera.
4. Development of Moildev library to build a monitoring system for vehicle entry and exit.

1.5 Thesis Organization

This thesis follows the structure of the research process consisting of several Chapters and is arranged as follows:

Chapter I. Introduction. This chapter contains the background that causes the problem of why this system is needed. Besides that, this chapter also discusses research benefits, related work, research boundaries, contributions, and organizational theses.

Chapter II. Literature Review. This chapter contains the theoretical basis and quotes from various sources related to research, as well as the basis for making the system. Such as a literature review that contains basic theories as a source in conducting research, such as theories related to the research of vehicle plate recognition systems that are used as a supporting tool in research.

Chapter III. Analysis and Methodology. This chapter contains the methods used in the research description of the materials and methodologies in conducting research, such as the tools and materials used for research as a solution to solving problems.

Chapter IV. Result and Discussion. This Chapter contains the results and discussion and a detailed description of the research conducted by the proposed method.

Chapter V Conclusion And Future Work. This chapter contains a summary of the conclusions of the research results that have been carried out based on the objectives and results obtained as well as some suggestions for future work.

Chapter II Literature Review

This chapter describes the theoretical background that will be used as a conceptual framework for research on computing software and hardware, starting from Camera Modelling, Camera Calibration, Digital Image Processing, License Plate Recognition System, and License Plate Recognition System. Device Material Used, which facilitates image processing in conducting research as thesis preparation, for a more detailed explanation as follows;

2.1 Camera Modelling

The camera model is a scene that can be optimized to look very good from a series of cameras that are used as the main target of the shooting. Camera models include standard pinhole camera models, lens distortion models, and cathartic cameras with parabolic mirrors. The camera model, in general, is very suitable to be represented in conformal geometry algebra (CGA) because it is based on inversion operations. Such as some camera models below;

2.1.1 Camera Perspective

A perspective camera model is a mathematical model of an ideal pinhole camera that follows perspective projection. An ordinary camera with a camera or target position and a field of view describing the correspondence between the observed point in the world and the camera resolution can be adjusted using the pixel Field of View (FoV) property in the captured image. To describe the transformation from a 3D point in the world to a 2D point in an image, we need to represent the camera with a coordinate frame. The weak perspective model is suitable when the object being observed is far from the camera relative to its size. In such a situation, the perspective distortion is relatively small. Therefore, a full perspective model is unnecessary, and a simpler and more robust model can be used.

2.1.2 Camera orthographic

An orthographic camera, also known as an orthogonal projection, displays an image of an object using several projection planes in two dimensions. The greatest benefit of visualizing orthographic images is that the image avoids distortion or the appearance of shortening because everything always seems parallel to the image plane. The pictures show actual sizes according to cut scale and proportions. View the scene without perspective projection. Given the direction of view, it will automatically scale the camera dimensions to summarize the scene. The pixel size can be set small enough (in world coordinates) to ensure the camera doesn't miss small details.

2.1.3 Camera of all directions

A 360-degree camera capable of viewing the scene from any direction is determined by the point from which the camera will view the view in all directions. 360 camera technology allows us to capture and record objects at a 360° angle, with the camera resolution can be increased using the property's pixel field of view. This camera is increasingly being applied in various fields, from the world of entertainment to the world of sports to the property business. Camera 360 technology is considered capable of seeing still or moving objects. A 360 camera is a camera that allows us to take pictures from all angles to produce images that can be seen from various sides – front, back, side, top and bottom.

2.2 Camera Calibration

Camera calibration is an indispensable procedure in 3D computer vision processes. This step is important for extracting three-dimensional information from two-dimensional image data, which is equivalent to determining intrinsic and extrinsic camera parameters. Intrinsic parameters relate to the internal characteristics of a camera, such as focal length, tilt, distortion, and image center, while extrinsic parameters describe the position and orientation in world coordinates. Matrix K is a 3x4 matrix containing intrinsic camera parameters that describe the interior orientation of the camera determined by;

$$\mathrm{K} = \begin{bmatrix} a_x & s & p_x & 0 \\ 0 & a_y & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Which consists of the camera parameters of intrinsic focal length (f), coordinates of the main point p_x, p_y and tilt s. Variables and $a_x = f * m_x a_y = f * m_y$, which represent focal lengths taking into account the scaling factor [13].

Besides the camera matrix at K intrinsic, camera calibration also has a rotation matrix R, a translational vector t. The two together constitute extrinsic parameters, and each determines the orientation of the camera's position in the world coordinate system. Extrinsic parameters consist of six degrees of freedom, three coordinates, namely X, Y, Z, and three angles α, β, γ. The names following the flight rules for the three corners are roll, pitch, and yaw.

- Rotation around the x-axis: roll

$$Rx = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\ (\gamma) & -\sin\ (\gamma) \\ 0 & \sin\ (\gamma) & \cos\ (\gamma) \end{bmatrix}$$

- Rotation around the y-axis: pitch

$$Ry = \begin{bmatrix} \cos\ (\beta) & 0 & sin\ (\beta) \\ 0 & 1 & 0 \\ -\sin\ (\beta) & 0 & \cos\ (\beta) \end{bmatrix}$$

- Rotation around the z-axis: yaw

$$Rz = \begin{bmatrix} cos\ (\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos\ (\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- The result in

$$R = R_x R_y R_z$$

2.2.1 Intrinsic camera calibration

Using a camera with a large angle wide-angle camera (WAC) or fisheye camera is the size Field of View (FoV). A wide-angle camera has an field of view (FoV) of 100 - 130, while a fisheye camera has a Field of View (FoV) above 180, as in Figure 2.1.

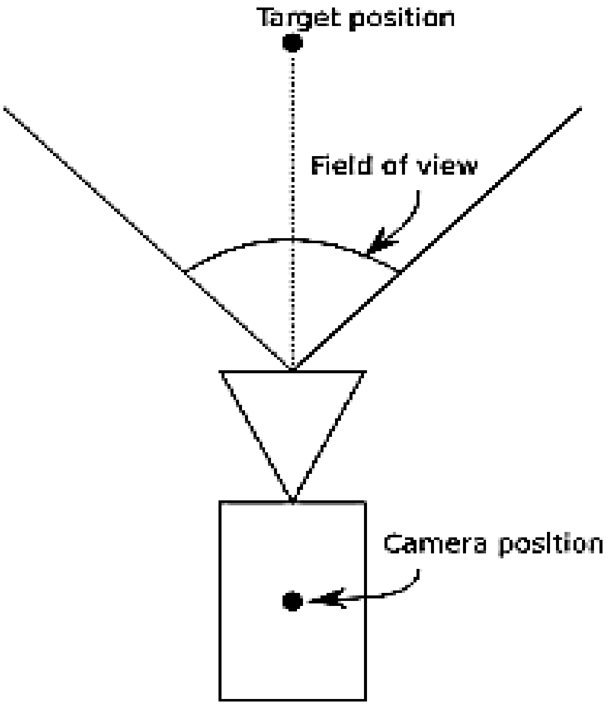


Figure 2. 1 Field of View (FoV)

However, apart from that, there are some drawbacks to a wide-angle camera, namely distortion that makes straight-line images curve. There are two types of radial distortion in this type of wide-angle camera, namely barrel distortion and bearing distortion, as shown in Figure 2 below;

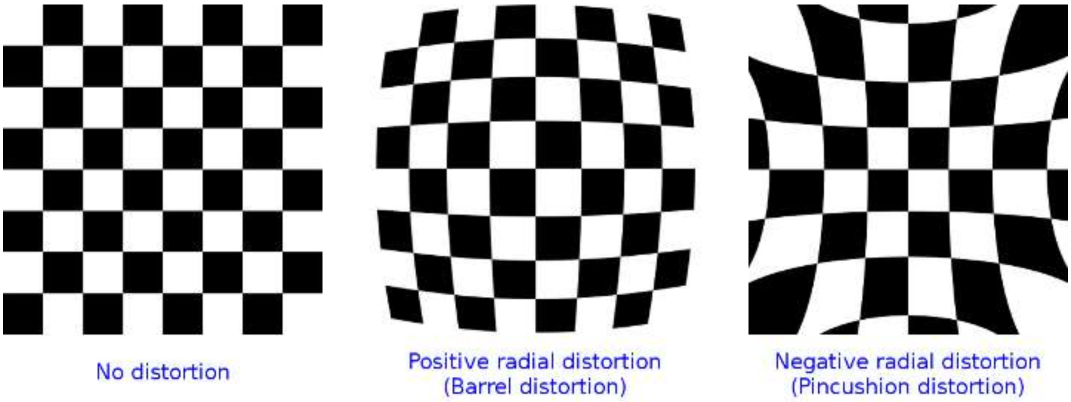

No distortion
Positive radial distortion (Barrel distortion)
Negative radial distortion (Pincushion distortion)

Figure 2. 2 Types of radial distortion [14]

Therefore, a camera model is needed to handle this distortion, where this model uses parameters that can describe distortions that occur on a camera. The process for estimating these parameters is called intrinsic camera calibration. Some methods to obtain the intrinsic parameters of the camera are the camCalib Scaramuzza toolbox [15], may toolbox calibration [16], OpenCV camera calibration [17], and MOIL camera calibration [18]. Especially for the calibration of MOIL cameras, this is a method that was fully developed in the MOIL laboratory.

2.2.2 *Extrinsic camera calibration*

Extrinsic camera calibration is calculating a value of extrinsic parameters on the camera. Extrinsic consists of a rotation R matrix and a transitional matrix with the symbol t. By knowing the external parameters, we can see the orientation between the camera and the object captured by the camera. With extrinsic parameters, we can also perform calculations between one image and another, which is very useful in reconstruction, as for the methods used in calculating calibration, such as the CamOdoCal calibration method. The use of this method depends on what is needed and the level of accuracy to be achieved [19].

2.3 Digital Image Processing

A two-dimensional (2D) function f (x, y) where x and y are the spatial coordinates of the image definition. The intensity or degree of the grayness of the image is the amplitude of the fat of any pair of coordinates (x, y). When x, y, and the amplitude value of f is finite and discrete quantities, these shadows are called digital shadows [20]. Digital image processing methods have importance in two main areas of application. The first is the improvement of visual information for human interpretation, and the second is the processing of image data for storage, transmission and representation for machine perception.

Digital image processing applies computer algorithms and digital image processing to achieve some expected targets, such as improvement, compression, etc. Digital image processing refers to applying computer algorithms and digital image processing to achieve typical targets, such as improvement, compression, etc. An image processing operation can be thought of as the workflow shown in Figure 2.3. The workflow starts by reading the image and then is processed using low-level, mid-level, or high-level operations. After processing, the image will be written to disk or visualized.

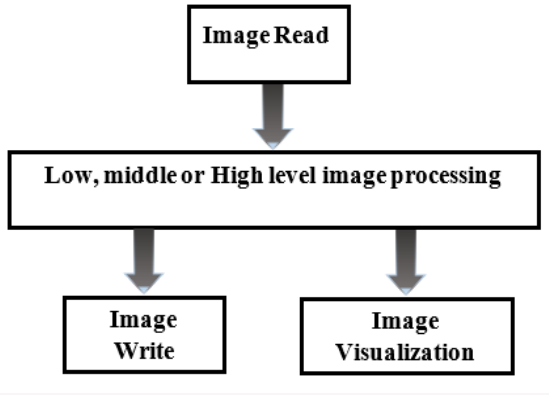


Figure 2. 3 Alur image processing

2.3.1 Low-Level Processing

An operation that uses the pixel value of an image to change individual pixels in an image is called low-level image processing. It's a pre-processing of digital image processing that aims to reduce noise and increase the contrast in the image used for further analysis. It can be divided into point-to-point, environment-to-point, and global-to-point operations. Traditional methods for low-level image processing include low-pass filtering for noise cancellation, grayscale operations such as histogram even distribution for contrast enhancement, edge detection using image gradients and related techniques.

2.3.2 Middle-Level Processing

Mid-level image processing mainly involves extracting scene descriptions from low-level extracted image descriptions. In reducing information, mid-level image processing obtains output from other data structures, such as detected objects (for example, faces) or statistics. Image processing operations such as Hough transformation, the center of gravity calculation, and object labeling are examples of mid-level image processing. Mid-level image processing operations are more limited to data parallelism than low-level operations.

2.3.3 High-Level Processing

Classification and recognition operations in image processing are examples of high-level image processing. This operation attempts to identify previously detected areas or features and interpret them. Artificial intelligence is the most popular technique that uses a high level of image processing, and this method shows promising results for the recognition and classification of images.

2.4 License Plate Detection System

At the beginning of the 20th century, the automotive industry developed rapidly, and the number of motor vehicles increased rapidly. From 1890 to 1910, the world witnessed the transition from horses to cars. As the numbers increased, law enforcement officials faced problems maintaining vehicle records and tracking them. As a result, in 1890, the first license plate was introduced by France and Germany was also followed by its introduction in 1993. Massachusetts was the first state to introduce license plates in the United States in 1903 with proper. The Netherlands became the first country to introduce national license plates in 1899 by starting with license plates with the number 1, which reached 2001 in 1906 because they chose a different way for license plate numbers. Number. 1 indicates some of the initial license plates introduced by various countries [21].

Figure 2. 4 License plate design in different [22]

The existing plate detection licensing system can be broadly divided into two: categories as multistage and single-stage methods. Most existing solutions to the license task of plate recognition have been considered a multistage method, which consists of three main steps. The first stage is the detection or extraction of license plates. The existing algorithm uses traditional computer vision techniques with object detection to find license plates in images. Traditional computer vision techniques are mainly based on the features of license plates, such as shape [23] [24] [25]. It makes it possible to share parameters and has fewer parameters than a typical two-stage model requires. As a result, they can be faster and more efficient than comparable two-stage methods [26].

2.4.1 License Plate Correction

License plates recognition has been focused on multistage processes; recently, there have been several successful attempts. All these efforts are for the best of vehicle license plate recognition knowledge. License plate recognition can be regarded as a special case in object detection with detectors; this model can take advantage of the fact of license plate detection and highly correlated recognition [27].

2. 4. 2 Character Segmentation

Character segmentation is the process of parsing images containing sequences of characters into sub-images of individual symbols. In vehicle plate recognition systems, character segmentation is used to identify the limiting rectangles of each character in the plate image. The proposed character segmentation method to apply to the same problem in the plate recognition domain is the only appropriate strategy for the proposed vehicle plate recognition system to obtain the right segmentation results.

License plate detectors using the above approach or candidate screening combined with CNN can also be found in the literature [3], [2], [27]. However, they tend to be computationally inefficient because they do not share computations as in modern meta-architectures for object detection, such as YOLO, SSD [28], and faster R-CNN [29]. Although the Scene Text Spotting (STS) method mostly focuses on large font variations and lexical/semantic information, it is worth mentioning several approaches that deal with rotated/distorted text and can be explored for detection in skewed views. Jaderberg and colleagues [30] presented a CNN-based approach to text recognition in landscapes using a fully synthetic data set to train their model. Although the results are good, they are highly dependent on N-grams, which is not true for atomic plate detection. Also explored synthetics data collection by realistically pasting text into real images, mostly focusing on localizing text. The output is a bounding box rotated with text around, which finds the constraint for rotation outside the common plane.

Recently, Wang et al [31]. presented an approach for detecting text in various geometric positions called the Instance Transformation Network (ITN). It is essentially a composition of three CNNs:

- A backbone network for calculating features.
- A transformed network for inferring affine parameters where there should be text in the feature map.
- A final classification network whose inputs are constructed by sampling features according to affine parameters.

While this approach can (in theory) handle out-of-plane rotation, it cannot properly infer a transformation that maps a region of text to a rectangle, as there is no obvious physical (or psychological) delimiting region around the text that should be mapped to a rectangle in an undistorted view. The vehicle plate system is rectangular and planar by construction, and we explore this information for regression of transformation parameters.

2.5 Device Material Used

Building a vehicle plate recognition tracking system requires additional hardware interconnected, such as the components to be used, such as raspberry pi, camera and the software used to support a system that runs well during the implementation process in the applied area. The hardware that will be implemented is as follows:

2.5 1 Raspberry Pi

The Raspberry Pi (RPi) is a minicomputer with low cost and the size of a credit card. The Raspberry Pi Foundation, located in England, is a fully functional developer of mini-board computers of small size and cheap packaging [32]. Whether you're looking for a device to surf the web or play a gaming game, are interested in learning how to program your own, or want to use your circuits and physical devices, the Raspberry Pi and its amazing community will have your back every step of the way. The Raspberry Pi is called a single-board computer, which sounds similar to a desktop, laptop or smartphone but is built on a single printed circuit of the board. Like most single-board computers, the Raspberry Pi is roughly the same size as a credit card - but that doesn't mean it's not powerful: it can do almost anything. Larger, power-hungry computers can do this, though it doesn't have to be this fast [33].

The Raspberry Pi board contains several components, including Broadcom-based ARM Processors, RAM using micro SD, GPIO, power features for input/output purposes, Graphics, Chips, and other connectors for external devices. Another advantage of this device is that it has an operating procedure similar to a personal computer (PC). Additional hardware, such as Power Supply Monitor, Keyboard, Mouse, and SD Card with the operating system (OS) installed as the operating system (OS) is required to use this device. The Raspberry Pi 4 Model B board can be seen in Figure 2.5. The full specifications of the Raspberry Pi 4 are in Table 2.1

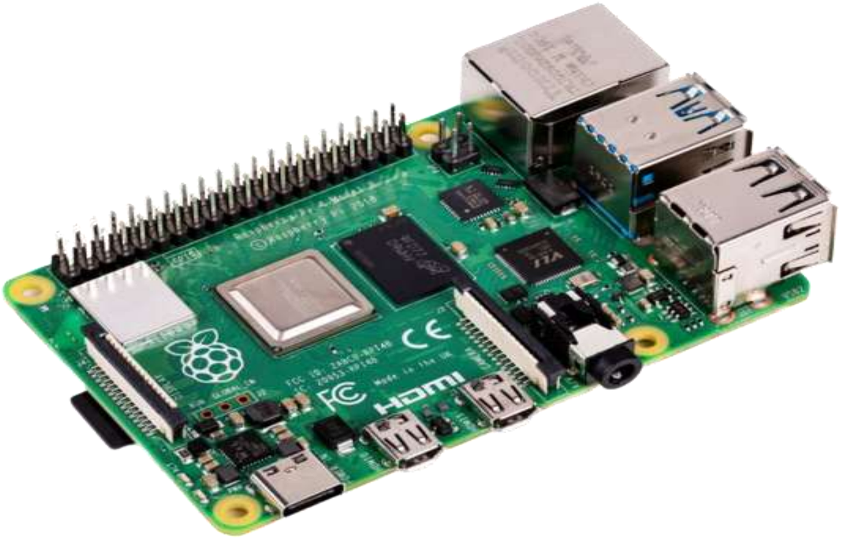


Figure 2. 5 Raspberry Pi board [34]

Raspberry Pi facilitates a free and open source operating system based on Linux to support the Operating System as its backbone. To date, more than 30 operating systems have been launched based on different flavors of Linux. Furthermore, the foundation of the Raspberry Pi provides various accessories such as Gertboard, Pi Camera, etc. For more detailed details, see Table 2.1.

Table 2. 1 Specification of the Raspberry Pi 4 model b

Raspberry Pi 4 Model B	
Processor	*Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz*
GPU	*Broadcom Video Core VI*
Video & sound	*2 × micro HDMI ports (up to 4Kp60 supported)* *2-lane MIPI DSI display port* *2-lane MIPI CSI camera port* *4-pole stereo audio and composite video port*
RAM	*1GB, 2GB, or 4GB LPDDR4 SDRAM*
Bluetooth	*Bluetooth 5.0, Bluetooth Low Energy (BLE)*
GPIO	*Standard 40-pin GPIO header*
Storage	*microSD*
Connectivity	*2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE* *Gigabit Ethernet* *2 × USB 3.0 ports* *2 × USB 2.0 ports.*
Dimensions	*Size: 88 mm × 58 mm × 19.5 mm,* *Weight: 46 g*

As mentioned earlier, the Raspberry Pi supports the Pi Camera, a portable and lightweight camera module. The camera display can be seen in Figure 2. 5. This camera uses the MIPI serial interface protocol, which is used to communicate between the camera module and the Raspberry Pi. Camera modules are commonly used in image processing, machine learning, or surveillance projects with 5MP color images and 2592*1944 pixels resolution armed on them [35].

2.5.2 Camera Fisheye (Entaniya)

Fisheye lenses, also known as fisheye lenses, are a type of wide-angle lens that can cause a convex effect on objects. The shape of this lens does resemble a convex fish eye. On a fisheye lens, the object will be distorted and appear round. There are two types of fisheye lenses, namely circular and diagonal types. The circular type will produce a convex photo in a circular frame, and the area is surrounded by black. At the same time, the diagonal type is a convex effect photo that appears intact in the image.

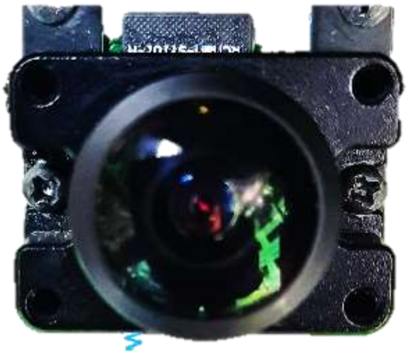


Figure 2. 6 Entaniya camera module

The Entaniya camera model is a fisheye camera with a wide Field of View (FoV), which is believed to be one of the methods used in this implementation. The lens can cover a very large landscape area, even if distorted. Fisheye lenses are not only used for landscape photos, but we can also use them to create unique portrait photos. Many people think that fisheye lenses and wide-angle lenses are the same. It is not the same, but the two lenses have something in common: the wide range with a short focal length compared to other lenses. While the difference lies in the convex distortion effect produced, as well as a wide angle of view [36].

2.6 Software and Library Used

This section will describe the software and libraries used in this project. The programming language uses Python, and the PyQt library builds the graphical user interfaces (GUI). Other libraries used are OpenCV and Moildev library, which have functions for Image Processing, and Docker simplifies deploying the apps and is supported by.

2.6.1 PyQt6

PyQt is a library that makes it possible to use the Qt framework from Python. Qt is a free and open-source cross-platform application written in C++ and is a widget toolkit that creates a graphical user interface (GUI). Qt offers programming solutions with powerful object-oriented programming (OOP) concepts: where programming problems are solved in object instances with attributes, functions, and interactions between objects. PyQt has tools that simplify the development of programs and graphical interfaces on desktops, embedded systems, and mobile devices, such as the Qt Framework, the Qt Creator IDE, and the toolchain. Using the Qt library from Python, it is possible to build applications faster without sacrificing a lot of C++ speed.

2.6.2 OpenCV-Python

OpenCV stands for Open Source Computer Vision Library. It is a free and open-source library for machine learning and computer vision. Where it was conventionally launched in 1999, it was developed by Gary Brodsky at Intel, written in the C/C++ programming language. The OpenCV source code and an additional module called OpenCV contribute are available on GitHub [37].

Therefore, this code is easily modified for specific needs. OpenCV is available for various programming languages such as C++, C, Java, MATLAB, and Python and supports operating systems such as Linux, Microsoft Windows, Mac OS, and Android. Because it leverages MMX and SSE instructions when available alongside the CUDA and OpenCL interfaces, most real-time vision applications are possible. Established companies such as Microsoft, Intel, Google, IBM, Yahoo, Sony, Honda, and Toyota use this library.

2.6.3 Moildev Library

The Moildev library is a collection of functions used to develop fisheye images of applications. The library was developed by The Omnidirectional Imaging and Surveillance Laboratory (OIL) at Ming Chi University of Technology, Taiwan, under the guidance of professors. *Chuang-Jan Chang*. Initially, this library was written in C++ to take advantage of the computing speed provided by this programming system. Currently, this library is available for python programming languages, which use application programming binding interfaces (APIs) that provide adhesive code created specifically to enable a programming language to use foreign libraries or operating system services at the same speed. The Moildev library is compatible with Linux and Windows operating systems (OS).

The Moildev library has functions such as rectilinear-1 mode, rectilinear-2 mode, and panorama. Each function has its parameters and controls with a given zenithal angle and an azimuth angle to reach a certain region of interest. Before applying the Moildev Function, the camera init configuration is required. This gives the camera properties the intrinsic and extrinsic parameters of the camera calibration [38].

2.6.4 YOLO3 up to YOLO4

YOLOv4 is further improved based on YOLOv3. Compared with the previous YOLO algorithm, YOLOv4 is verified on the COCO data set, and its running speed is twice as fast as YOLOv3 in Efficient Det. While improving the target detection speed, the detection accuracy is also improved (AP and FPS). Based on the original YOLOv3 target detection architecture, the YOLOv4 algorithm introduces some optimization methods from data processing so that the model has reached the best match in terms of detection speed and accuracy [39]. YOLOv4 uses a new feature extraction method, CSPDarknet53, to extract the features of the input image through the feature extraction network and then divide the input image into S*S grids. The grid where the target center is located is responsible for the detection of the target.

2.6.5 Others Library

Other libraries need to be used to complete the application depending on the design and what will be used in the future. PyPI provides hundreds of thousands of libraries that can be used to achieve our goals.

Chapter III Analysis and Methodology

This chapter discusses the material and methodology of fisheye image processing as a proposed method to establish a vehicle plate recognition monitoring system in the parking area with the application of multi-view to obtain information at each angle according to the target equipped with detection methods and identify character recognition on vehicle plates including the hardware settings used in this experiment.

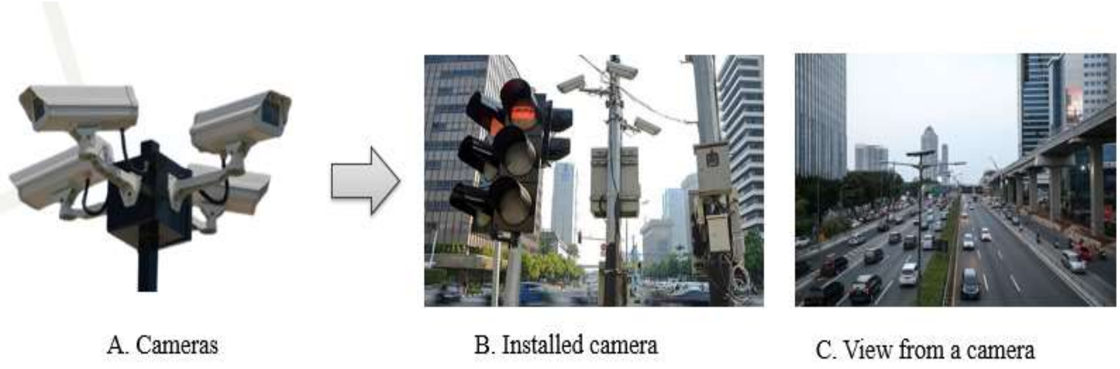
3.1 Analysis System

System analysis, an attempt to conduct an analysis understanding of a problem, is carried out by the study, which aims to gain a certain understanding of a problem being studied. A system is a process assembled or built from several components that become a unit to achieve a certain goal. Such as analyzing the system in this study, namely by understanding the first step, which is to identify problems in the vehicle plate monitoring system to be built.

This vehicle plate recognition monitoring system is made and provides solutions to problems contained in monitors that use many cameras to see on several sides of the angle. This causes a heavy burden from the costs incurred to buy each camera to know and detect vehicle plates. The vehicle plate recognition monitoring system using a wide area coverage fisheye camera is designed to contribute as an alternative to the number of cameras used to monitor the area to be built using the python programming language. The existence of a vehicle plate monitoring system on got provides solutions to existing problems.

3.2 Camera Monitoring System

In this study, several components of the hardware used for the camera system series that are paired at the point of placing the camera that you want to use as a monitor to be connected with the software can be seen in Figure 3. 1 as the series we created. A surveillance System or the so-called monitoring system controls all activities visually (images) in a certain area that is installed by a device in the form of a camera. Its function is to directly monitor, observe and record events in a certain place, room or area; this tool consists of cameras, video recorders, and monitors that are integrated into an online network system or can be implemented internally.



A. Cameras

B. Installed camera

C. View from a camera

Figure 3. 1 (A), (B), (C) The position of camera for capture information

The purpose of anyone using a monitoring system is to monitor a large area and perhaps far from a location that is difficult to control and reach simultaneously. The camera functions to capture or take pictures that are processed on the system. Installed in the areas/places to be observed. In its function, the camera can be categorized according to the needs and desires, such as standard, fisheye, and pinhole. Technologically, the types of cameras in use are wireless, outdoor or indoor, and also functionally, there are movable ones (pan, tilt, zoom / PTZ).

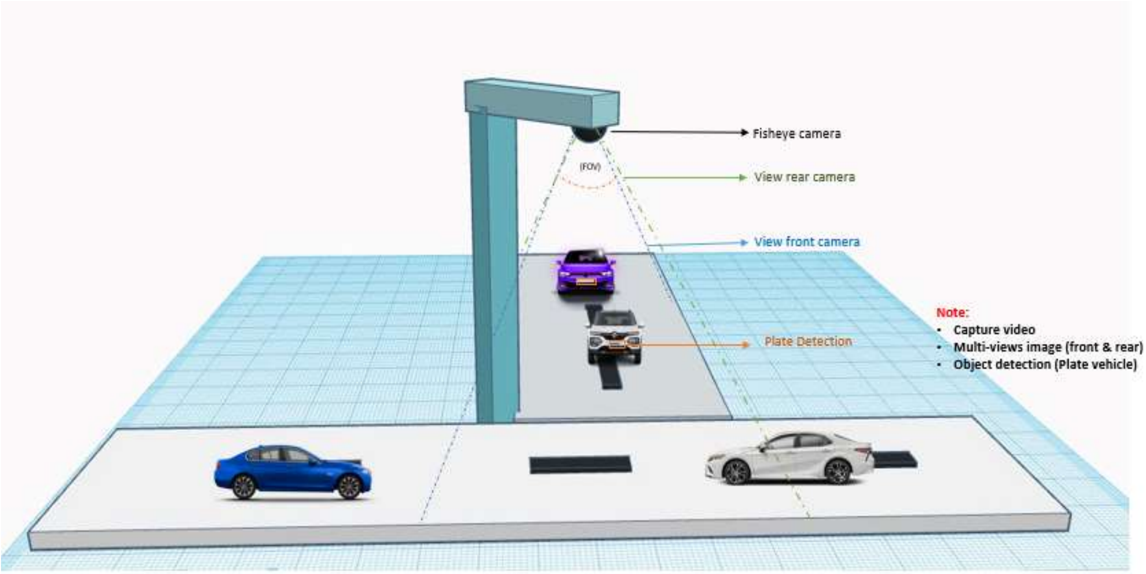


Figure 3. 2 Simulation of design the case camera two views

The quality of the camera can be determined by several things, such as the Image sensor, namely: the part that functions to capture images; the higher the resolution and sensitivity (illumination), the better the quality. Image sensors that are often used are 0.25", 1.3", 0.5" and 1". The ability to process images on the controller, such as automatic gain control, white balance, etc. The lens directs the image to the image sensor, the range and extent of the area you want to take pictures in is called the focus. This system monitoring technology using Video Recorder is a technology that can be accessed/viewed from various places that already have a good computer network and online have a high ability to access it and have been set up, making it easier for users to be able to send image data remotely. The function of the multiplexer is to regulate the display and recording of images from the camera to a monitor and VCR. DVR Has the ability as a multiplexer and VCR. With today's digital computer technology, all data is converted and processed in digital form; DVR technology is currently based on Personal Computer (PC) with specifications to those on the market today.

In the process of sending image data on a network to the monitor, this camera only requires a very small bandwidth, so it doesn't cause the network to feel slow, even though simultaneously there are 5-6 cameras displayed on a monitor; this is because the process of sending images to the monitor on do one by one on a cable network so that it does not require a large bandwidth.

3.3 YOLOv4 Algorithm Core

The YOLOv4 network structure introduces mosaic data enhancement at its input. It differs from CutMix's technique, which splices together two images. Instead, the four images are arbitrarily resized and arranged to achieve splicing. The data features are boosted to realize the feature extraction of small targets better and increase the network's robustness by data enhancement at the input, in the data set including varied size targets simultaneously. By increasing the input data, GPU consumption is simultaneously decreased. Also included in the input end enhancements are cmBN, SAT self-antagonism training, etc. The foundation of the entire method is the YOLOv4 backbone network CSPDarknet53 [40], YOLOv4 Darknet53 adds CSP to each large residual block. Part of the feature map is subjected to convolution operation, and the other part is combined with the previous convolution result. In target detection, CSP can effectively improve the learning ability of CNN while reducing the amount of calculation. These categories are likewise covered by the regression network's cost function. Regressing the coordinates using the Mean Squared Error is the conventional method.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}i - yi)^2 \qquad (3.1)$$

According to the paper, it treats these points as independent variables but disregards the object's integrity. An IoU [41] loss that considers both the projected Boundary Box (BBox) and the basic truth Boundary Box has been proposed to improve this.

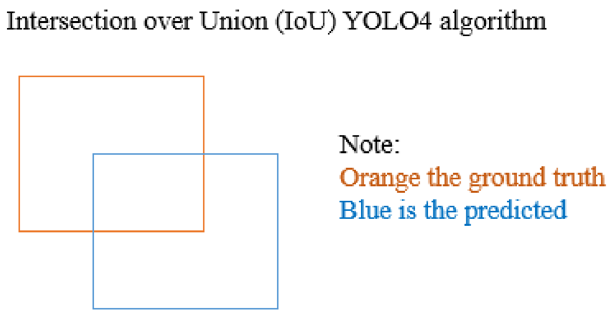


Figure 3. 3 Intersection IoU YOLO algorithm

$$\mathrm{IoU} = \frac{(\mathrm{B} \cap \mathrm{B_{gt}})}{(\mathrm{B} \cup \mathrm{B_{gt}})} \qquad 3.2$$

The disappearance of GIOU [42] expanded on this concept by adding the shape and orientation of an object to the covered area.

$$CIoU = 1 - Iou + \frac{|C - \mathrm{B} \cup \mathrm{B_{gt}}|}{|C|} \quad (3.3)$$

But this loss function tends to expand the prediction boundary box first until it is overlapped with the ground truth. Then it shrinks to increase IoU. This process requires more iterations than theoretically needed.

First, Distance-IoU Loss (DioU) is introduced as:

$$RDIoU = \frac{p^2(b,\ b^{gt})}{c^2} \quad (3.4)$$

Where:

b and b^{gt} denote the central point of B and B^{gt}

P(.) is the Euclidean distance

c is the diagonal length of the smallest enclosing box covering thethe two boxes.

$$CIoU = 1 - Iou + \frac{p^2(b,\ b^{gt})}{c^2} \quad (3.5)$$

However, CIoU drawbacks are also discussed and take into account the aspect ratio, overlapping area, and distance between center points.

The activation function of CSPDarknet53 uses the Mish activation function and adds a Dropout module that can alleviate overfitting. This module can reduce the complexity of the network structure by randomly deleting part of the insensitive information of the convolutional layer. The deleted information convolutional layer can be re-learned through adjacent activation units. The neck part's main function is to generate a feature pyramid. The feature pyramid can detect objects at different scales. At the same time, PLANET adopts a new bottom-up path to enhance the FPN structure so that low-level features are transmitted more effectively. For the Prediction part, the anchor frame mechanism of the output layer is the same as YOLOv3. The main improvement is the loss function CIOU_Loss during training, and the NMS filtered by the prediction frame becomes DIOU_nms.

Unlike the YOLOv3 algorithm that uses FPN (feature pyramid with the network) for up-sampling, YOLOv4 first propagates the semantic information of the high-level features to the low-level network through up-sampling. It then merges with the high-resolution information of the underlying features. The detection effect of high and small targets. Then increase the information transmission path from the bottom to the top and enhance the feature pyramid through downsampling. Finally, feature maps of different layers are fused to make predictions. PANet (Path Aggregation Network) makes full use of feature fusion. YOLOv4 also changes the fusion method from addition to multiplication so that the network can obtain more accurate target detection capabilities.

3.4 Fisheye Lens Technology

Lensa fish eye is the unique super wide angle fish eye image sensor lens with a short focal length that produces strong visual distortion intended to create a wide hemispherical. The most significant feature of fisheye cameras is the wide turning field. The fisheye camera, also called a fisheye image sensor, can capture clear images with a Field of View (FoV) of more than 180 degrees more but longer barrel distortion. Pola forms the corresponding image on the image plane behind the fisheye lens. The spatial optical projection path, which is equivalently represented by combining a fisheye lens and an image plane in unity shown in Figure 3.3.

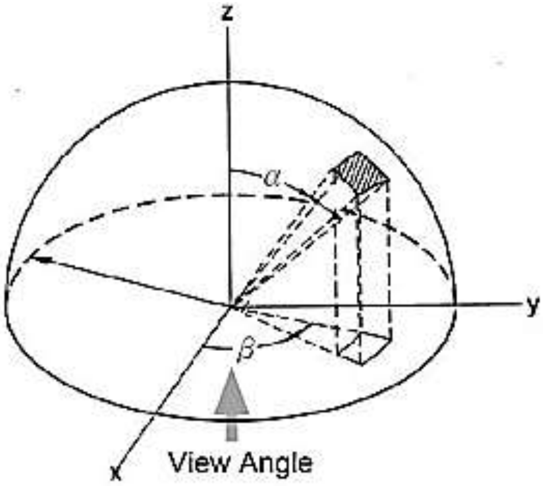


Figure 3.4 The hemisphere coordinate system

The target position is adjusted until the image is completely composed of concentric circles. The method of presenting fisheye camera images integrates multiplicator metrology and cartography to describe the fisheye camera's projection mechanism systematically. The fisheye camera in our proposed method generates a hemisphere coordinate system. Then the image center is connected to find the optical axis of the Fisheye lens.

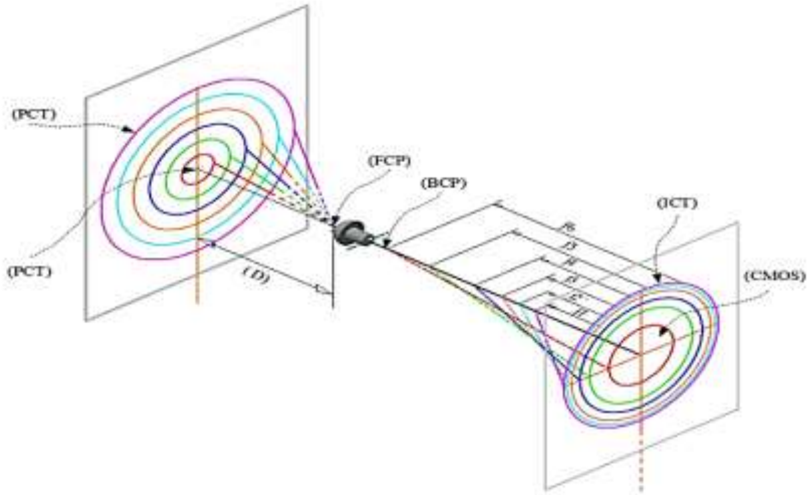


Figure 3. 5 The fisheye projection lens

The image point position refers to the main point on the plane of the direct image reflecting the corresponding distance of zenith alpha and the distance of the azimuth from the rays of vision in space to normalize the point imaged to the small sphere presented through the coordinates of the system below:

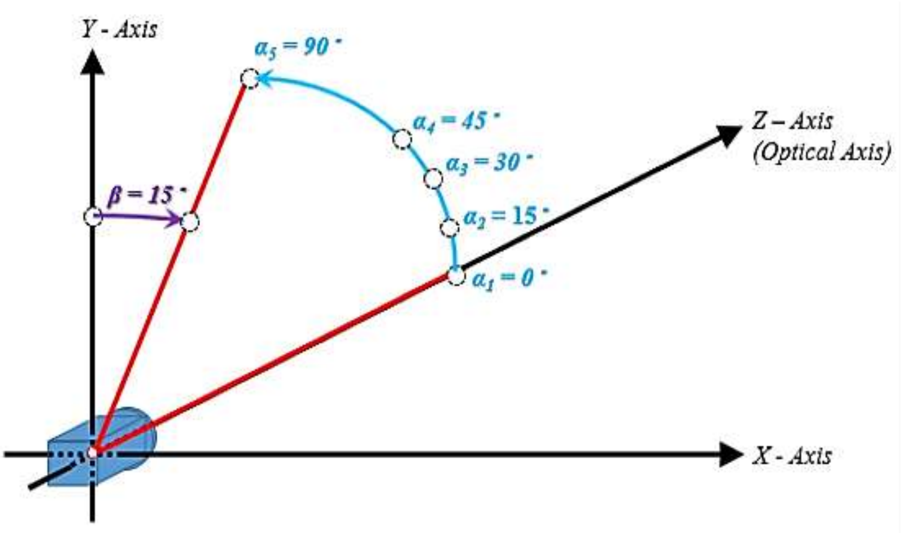


Figure 3. 6 The fisheye projection lens

In the definition of a coordinate system, the angle of each optical axis is the zenith angle (alpha), and (beta) is the azimuth angle, that is, the angle surrounding the optical axis. It has to do with the coordinate systems X, Y, and Z, in which the Z axis defines the optical axis. The zenith angle is the angle from the vertical optical axis to the X-axis and the Y-axis, as shown in Figure 3.5

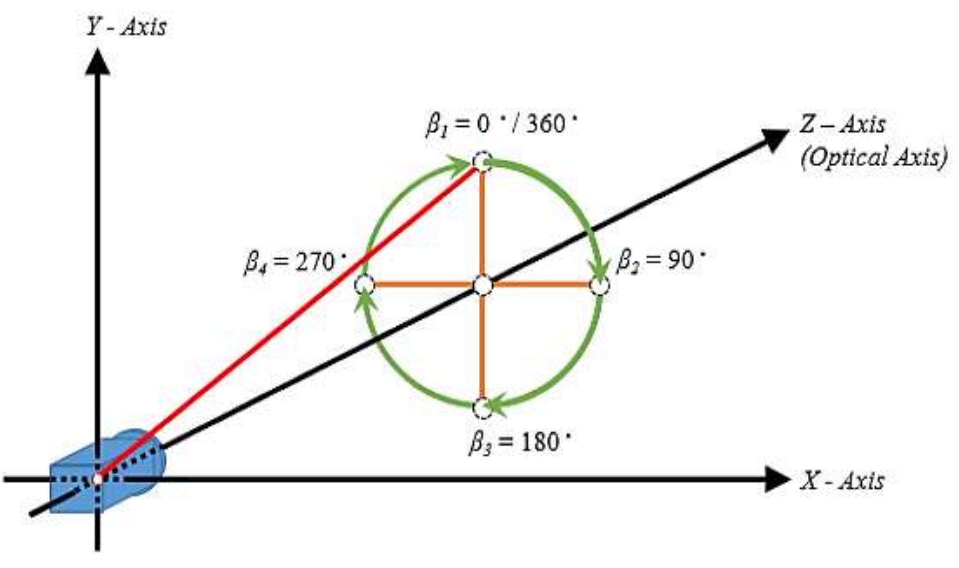


Figure 3. 7 The azimuth angle

Meanwhile, the azimuth angle is defined as the positive angle, Y is the reference point with a value of 0 degrees, and the Z axis is used as the axis of rotation, as shown in Figure 3.6. The oasis around the optical axis is the angle of the Y axis starting from the positive direction and clockwise around the X axis to apply multi-views with different angles from the camera.

3.4 A Rectilinear View

A lens scheme projection of more than 180 degrees of general wide angle with the optical s axis coinciding with the Z axis of the coordinate system, and the angle coming from the coming rays is measured as a zenith angle (alpha) which distinguishes from ordinary cameras; fisheye lens images have severe lens barrel distortion. Axisymmetric distortion in the observation room and symmetry center in Citra requires calibration. In contrast, calibration is an important step that must be done first to obtain camera parameters before image processing. The camera calibration method uses the MOIL camera calibration developed by professor *Chuang-Jan Chang* of the omnidirectional imaging library [7].

The parameters of fisheye lenses using this calibration method and the calibration target use a physical concentric test pattern. The calibration target is placed in front of the Field of View (FoV) fisheye lens to obtain and ensure optical parameters consisting of the focal length constant, the main point, and the projection function of the lens fisheye camera. However, as previously stated, an image by the lens of the fish's eye causes psychological discomfort due to the distortion of its barrel, although the ultra-wide angle extracted by means of capturing object information far from the optical axis does not appear natural to the naked eye.

In this case, the most addressed image is a rectilinear image which can be obtained by pointing the camera in the direction of the object according to the target. A camera that can physically provide such an image is equipped with rectilinear lenses that can be oriented in the direction that requires the most attention. In addition, images can continue to be produced while dynamically tracking moving objects, such as in the implementation of the vehicle plate recognition monitoring system to be built next, a method to realize its functionality with software such as altering the coordinates of the plane of the fisheye image into hemispherical coordinates due to the mathematical nature of the three dimensions of rotation; different images are obtained depending on the two operations namely zenith (alpha) and azimuth (beta) by moving the optical axis to the angle specified with the zoom operation allowed.

The coordinate system that describes the object before the zenith (alpha) operation is a world coordinate system. The x-z plane coincides with the horizontal plane, and the y axis coincides with the vertical. Next, assume that the optical axis is aligned with the z-axis before the zenith (alpha) operation. The coordinate system after zenith (alpha) surgery is the first world coordinate system, and the coordinate system after azimuth (beta) surgery is the second coordinate system. The x, y, and z axes of the guinea coordinates of the two systems are aligned with the x, y, and z axes of the rectilinear coordinate system, respectively, as shown below:

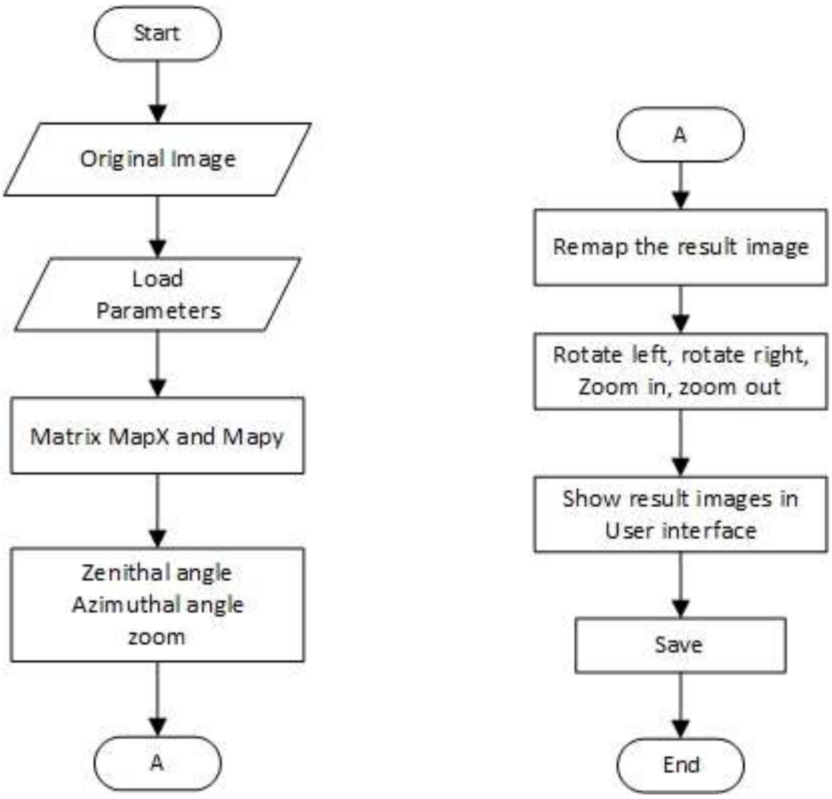


Figure 3. 8 The flowchart remap image

Changing the image with a rectilinear view is an image that has not been distorted in a certain area according to the input coordinates by returning an image with a rectilinear view so that it produces a beta rotation offset rotation degrees around the Z axis (roll axis) after alpha Degree rotation offset around the X axis (pitch axis) so that it can be implemented into the system you want to build, for more details to know the work specifications of the system to be built, you can see the process workflow in Figure 3.9;

3.5 System Workflow

The system flow is the result of the design of an introduction monitoring system that will be built that tells the steps of the system's work from the beginning of the start of the use of the system through the processes that will be used to see the extent of the system processes carried out from the beginning of the system until the system ends. The data needed in this system is data from vehicle plate detection information. The member Ming-Chi Omnidirectional Imaging Laboratory will test this system. To provide an understanding of the ways or workflows of the vehicle plate monitoring system in this study can be seen in the following Figure 3.8:

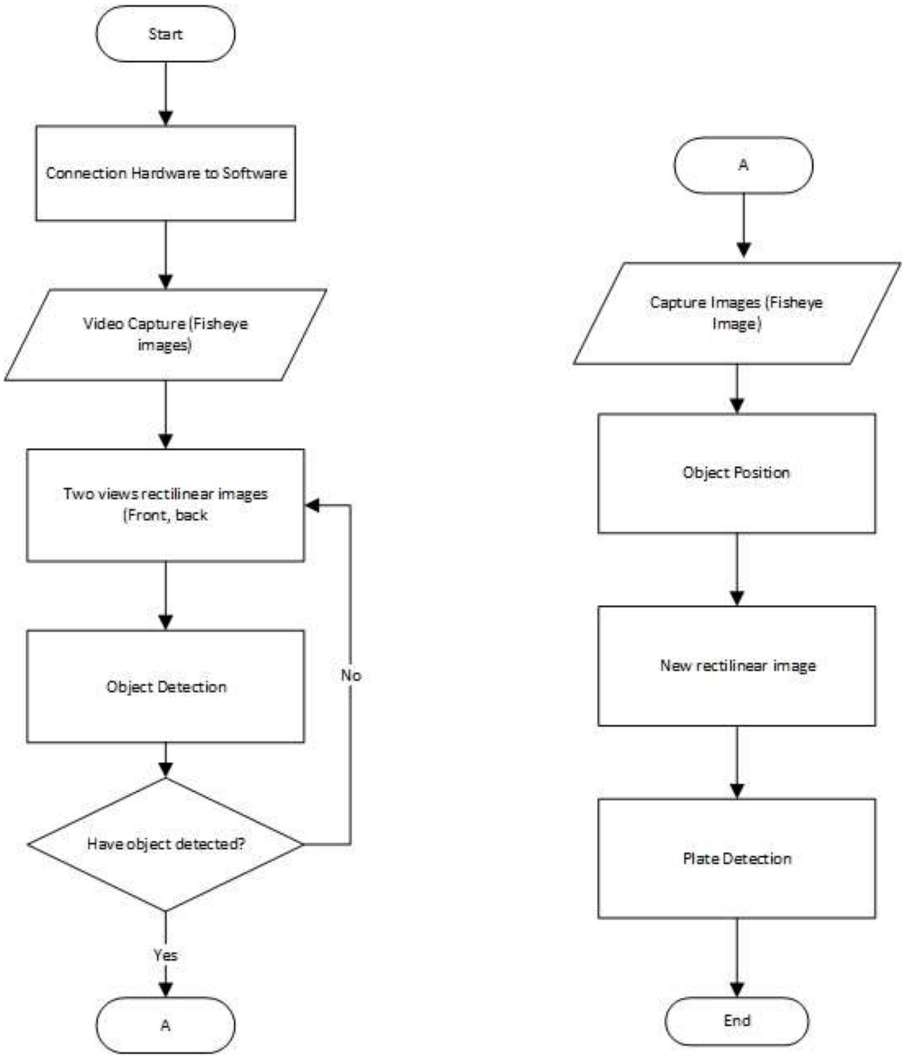

Start
Connection Hardware to Software
Video Capture (Fisheye images)
Two views rectilinear images (Front, back
Object Detection
Have object detected?
No
Yes
A
A
Capture Images (Fisheye Image)
Object Position
New rectilinear image
Plate Detection
End

Figure 3. 9 The flowchart multi-views system

The process of implementing a vehicle plate detection monitoring system using a fisheye camera by dividing the two main areas that the target angle wants to know as the monitoring side; this system includes two stages of the process; the first, the stage of tracking the vehicle plate from the information on the side of the angle implemented, the second, the stage of identifying the introduction of vehicle character plates which is processed directly on the surveillance system in the area so that the vehicle plate detection monitoring system as a smart security system.

3.6 Analysis Design

System design is a stage that aims to provide a clear and complete picture of the design and implementation of the vehicle plate recognition monitoring system to be made. This system design stage is carried out after completing the problem identification and system analysis stage. The design includes Unified Modelling Language (UML) diagram, *interface* design and storyboard design.

3.6.1 Unified Modelling Language (UML)

A flowchart is a chart that shows the workflow or what is being worked on in the system as a whole and describes the sequence of process procedures performed by the system. In other words, this flowchart is graphical decryption of the combined sequence of procedures that make up a system. This system design is a conceptualization stage, which is a stage that requires system design software to try to know for sure about the things that are the needs and expectations of users so that later the applications created are indeed needed by *users* and satisfy their needs and expectations.

The conceptualization in analyzing this system uses the concept of Unified Modelling Language (UML), one of the tools or models for designing object-oriented software development. Some literature mentions that UML provides four types of diagrams: communication diagrams, sequence diagrams and representation diagrams combined into interaction diagrams. The diagrams used in this monitoring system includes:

3.6.2 Use case Diagram

Using Case Diagram is modelling to describe the behavior of the system to be made. Use Case diagrams to describe an interaction between one or more of the systems to be created. With a quick understanding, use case diagrams are used to determine what functions are in a system and who has the right to use these functions..

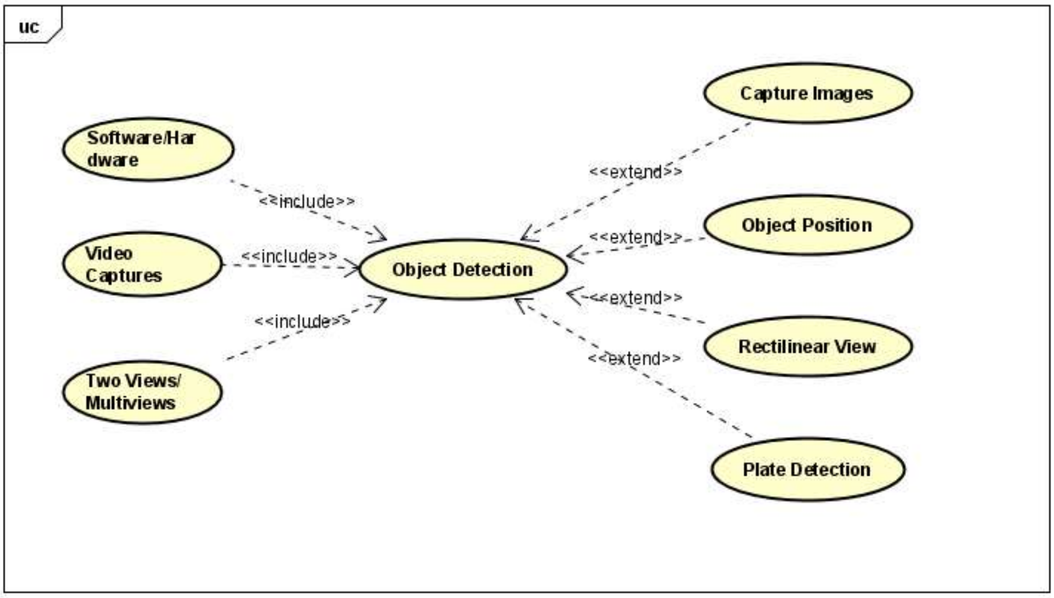


Figure 3. 10 The use case diagrams

In this case, Vehicle Plate Recognition Using Multi-views From a Single Fisheye Camera only involves a system that interacts directly with vehicular traffic and performs direct system management. Here is a Figure 3.9 of the Use case Diagram:

1. *Use case Software/Hardware*

In this software/hardware use case, it functions to connect all components that will run on the system, such as raspberry pi components, cameras, Lan or Wi-fi cables, and computers as a means of running the system.

2. *Use case Video Captures*

This video capture section is always running or active to capture passing vehicles if there is a license plate on a running vehicle.

3. *Use case Two View/Multi-views*

From a fisheye image known as a wide view with a Field of View of more than 180 degrees, it can be divided into two views which is an application of our method.

4. *Use case Object Detection*

Object detection is one of the methods applied, which functions when a vehicle crosses the camera and detects the vehicle's license plate.

5. *Use case Capture Images*

When the vehicle plate is detected, the system will automatically take the image to be processed as input.

6. *Use case Object Position*

When the image becomes the input for processing, the system will automatically change the position of the vehicle plate to a large straight; this is useful for the system to make it easy to recognize characters on the vehicle plate.

7. *Use case Rectilinear View*

The rectilinear view is an implementation of the method we use where it serves to convert fisheye images to a rectilinear view.

8. *Use case Plate Recognition*

After going through several processing stages, the system will recognize the characters one by one on the vehicle plate, either in letters or numbers.

3.6.3 Activity Diagram

The activity diagram illustrates the various flows of activity in the system being designed, how each flow begins, the decisions that may occur, and how they end. Activity diagrams can also illustrate parallel processes that may occur on multiple runs. In system monitoring, the introduction of vehicle plates and activity diagrams, where each diagram explains the process of running the system from various operations contained in the system. Here is a Figure of the activity diagram.

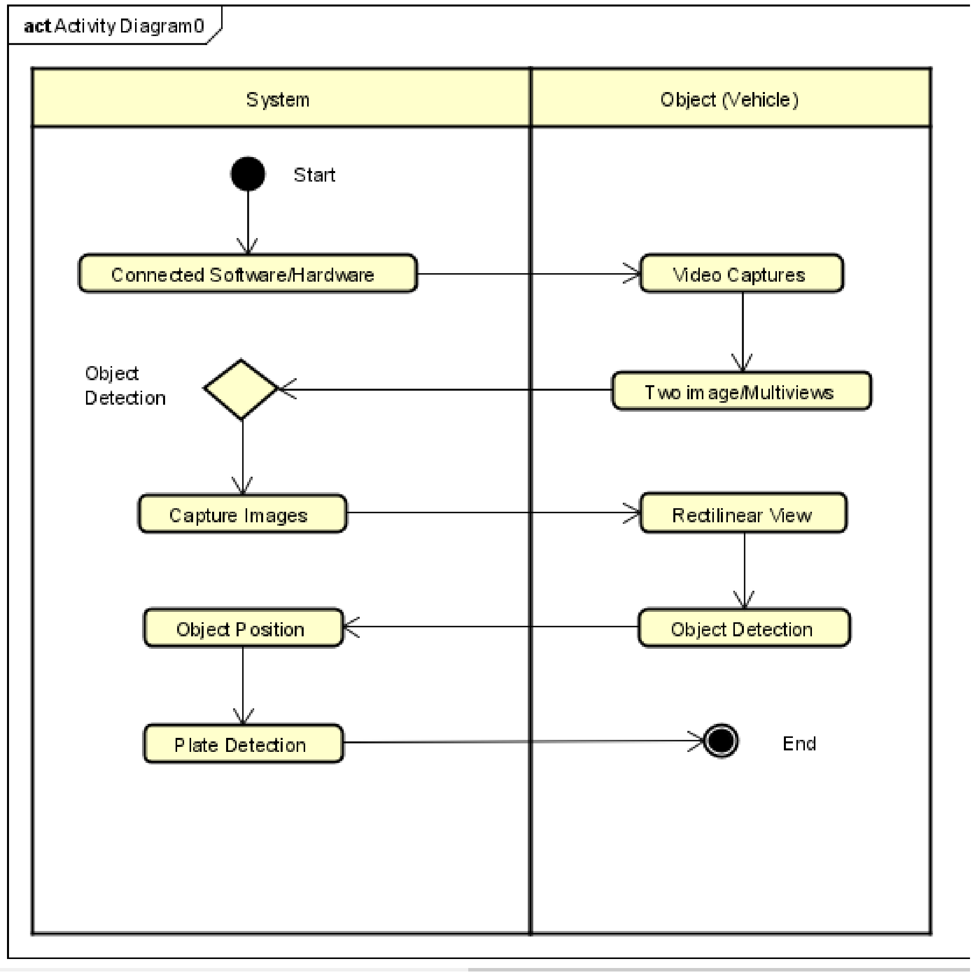


Figure 3. 11 The activity diagrams

The Figure 3.10 can explain that after the software/hardware devices are connected to run the monitoring system, the system will display two images that point forward and backward with an active status or video capture. A multi-view camera with object detection will capture the vehicle plate, automatically capturing the image as input for the system to process and then identify the vehicle plate.

The system will process the image to be converted to object position or enlarge the vehicle plate section, which forms a rectilinear view for vehicle plate recognition. After all these processes are done, the system will return to its original state. It will always be active for video captures, waiting for passing vehicles until the process is repeated.

3.6.4 Sequence Diagram

A sequence diagram is a diagram that depicts the interaction between objects and indicates communication between these objects. This diagram also shows messages exchanged by objects that perform a specific task or action. This diagram is useful for illustrating the interaction between objects, emphasizing the sequence of processes or events. Sequence diagrams are commonly used to describe a scenario or series of steps performed in response to an event to produce a specific output. The diagram shows a sequence diagram of the system monitoring the vehicle plate. The first sequence diagram is a sequence diagram for the user. The following is a sequence diagram of the vehicle plate monitoring system user diagram:

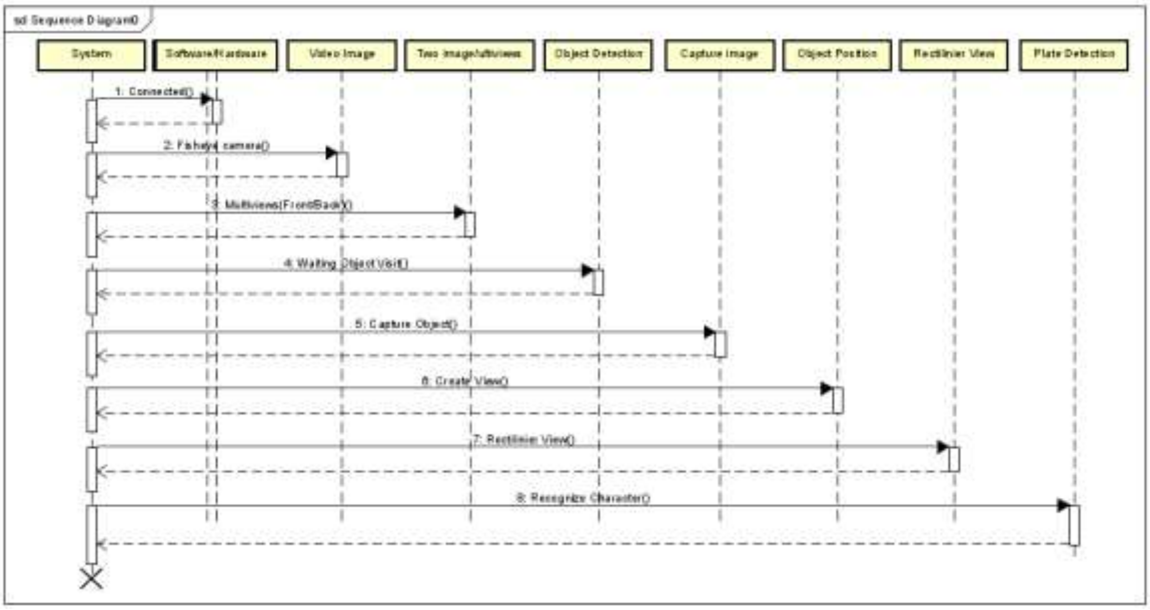


Figure 3. 12 The sequence diagrams

In the Figure 3.11, it can be explained as follows; the monitoring system will connect all software/hardware device components to the system by dividing one camera into two front and back views; after that, the system will activate automatically with video image status or wait for input to pass, then after the vehicle that enters the image automatically will be detected with object detection embedded in the monitoring system and will then capture the image, which is the processing input on the system.

The result of the image capture will be taken from the vehicle plate, which will be converted into an enlarged and straight object position in the plane so that it becomes a rectilinear view to be read by entering the process or the final stage, namely character recognition, all these processing sequences will be controlled and returned to the automatic system.

3.6.5 User Interface Design

The interface is a medium that connects to the system to be built. As a configuration setting, a system must have an interface suitable for the application and easy to use by the user. Therefore, it's necessary to design an interface so that the interface built on the system monitoring system can work as expected. The interface design of this system, which will be built using the Entity Relationship Diagram (ERD), is as below.

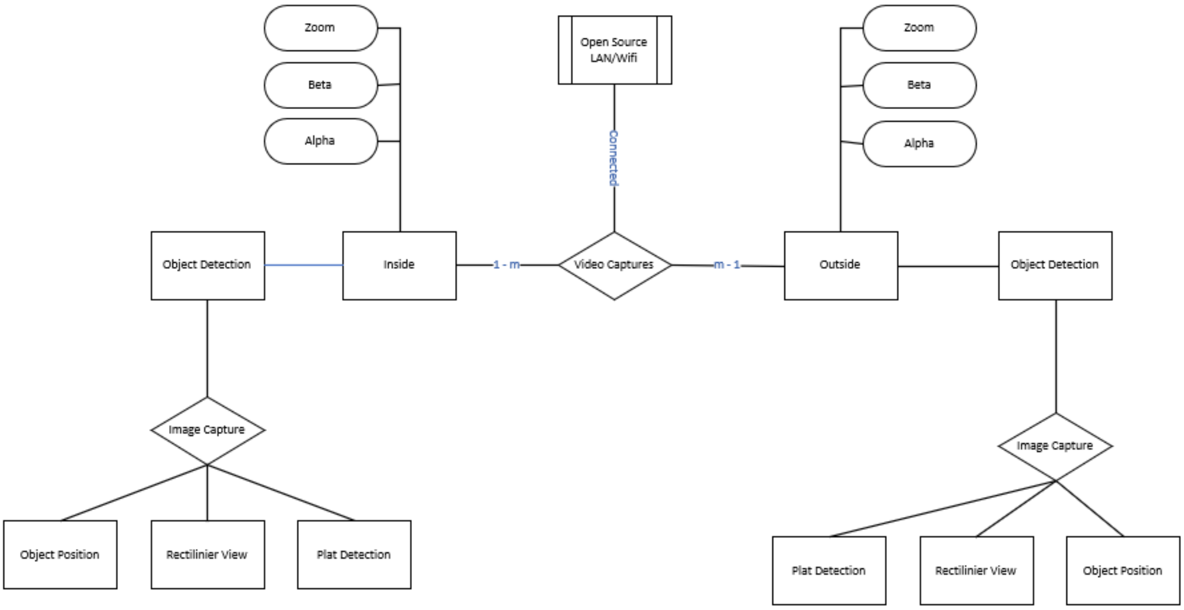


Figure 3. 13 Entity Relationship Diagram (ERD) system

Chapter IV Result and Discussion

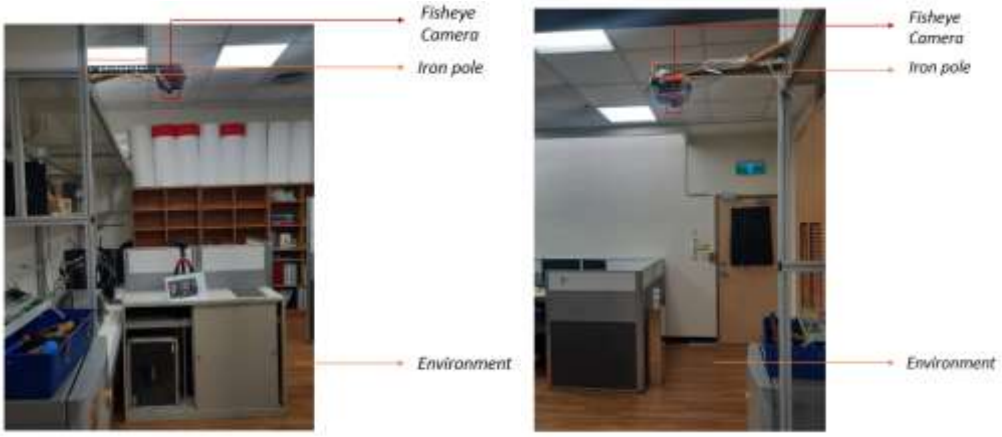
This session will explain the experimental results obtained from this research and the discussion carried out. There are several topics discussed, namely experimental results, a rectilinear view, and user interface design, explanations for more details are as follows:

4.1 Experiment Result

This section is the result of the implementation of the application, where this research was created and designed using the Python programming language, resulting in an application, namely a vehicle plate detection monitoring system for incoming and outgoing traffic in the required area. In this experiment, 2 experiments will be carried out. The first will be a simulation process in a laboratory environment with a distance of 1-2 meters. This process uses an entaniya camera to see the algorithm running with good image quality in a range. Not too far (short). Furthermore, the second process will be carried out outside the laboratory with the same goal but at a distance of 1–30 meters further using the same camera in this application. Let's take a closer look at the process carried out below:

4.1.1 Installation camera inside laboratory

The device or component is installed to determine the camera's distance, range, and position at the height of 1.7 meters and the state of the surrounding environment before taking datasets in the laboratory environment. As shown in Figure 4.1, a fisheye camera was installed in the Ming-Chi Omnidirectional Imaging Laboratory environment.



A. From the right of the camera

B. From the left of the camera

Figure 4. 1 (A) and (B) the position of fisheye camera use in laboratory

Figure 4.1 shows a fisheye camera inside the Ming-Chi Omnidirectional Imaging Laboratory environment. This is done to find out when the camera is positioned and see the image quality at a close distance when taking the dataset, as shown in Figure 4.2, the results of taking the dataset in the environment obtained.

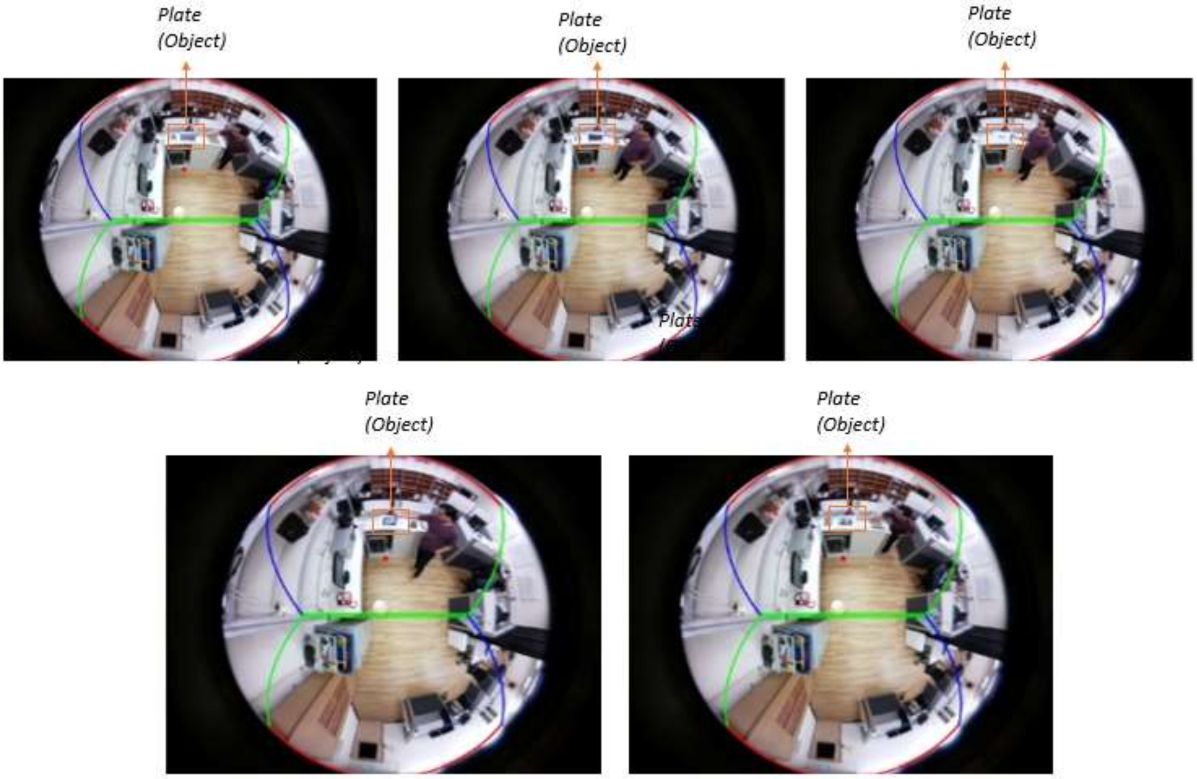
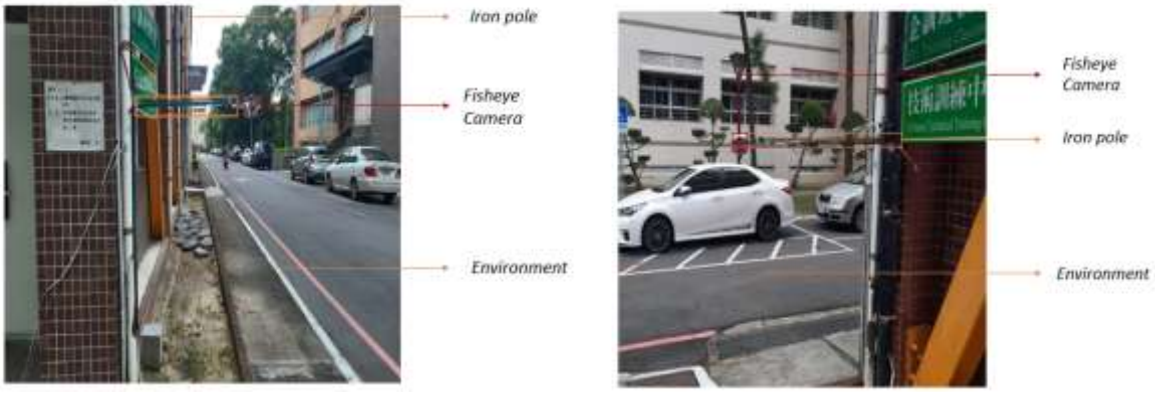


Figure 4. 2 Collect the datasets from in laboratory using fisheye camera

As shown in the results of the dataset in Figure 4.2, which uses a fake vehicle plate to be simulated as a test in an environment with stable light conditions and a distance that is not too far from 1-2 meters in the implementation of this monitoring system.

4.1.2 Installation camera outside the laboratory

The second process is carried out beside the laboratory by taking the entrance area behind the laboratory and the exit area leading to the mechanical engineering building. This process is carried out with the same installation and components of the camera installation at the height of 1.7 meters, with the process in a laboratory environment, only this process is carried out with a longer range and to determine the position of the camera and the environment around vehicle traffic, as shown in Figure 4. 3. a fisheye camera installed outside the environment.



A. From the right of the camera B. From the left of the camera

Figure 4. 3 (A) and (B) the position of fisheye camera use outside laboratory

As shown in Figure 4.3, the fisheye camera was installed outside the environment of the Ming-Chi Omnidirectional Imaging Laboratory. This is done to determine when the camera position is used when taking the dataset, and Figure 4.4 shows the results of taking the dataset in the obtained environment.

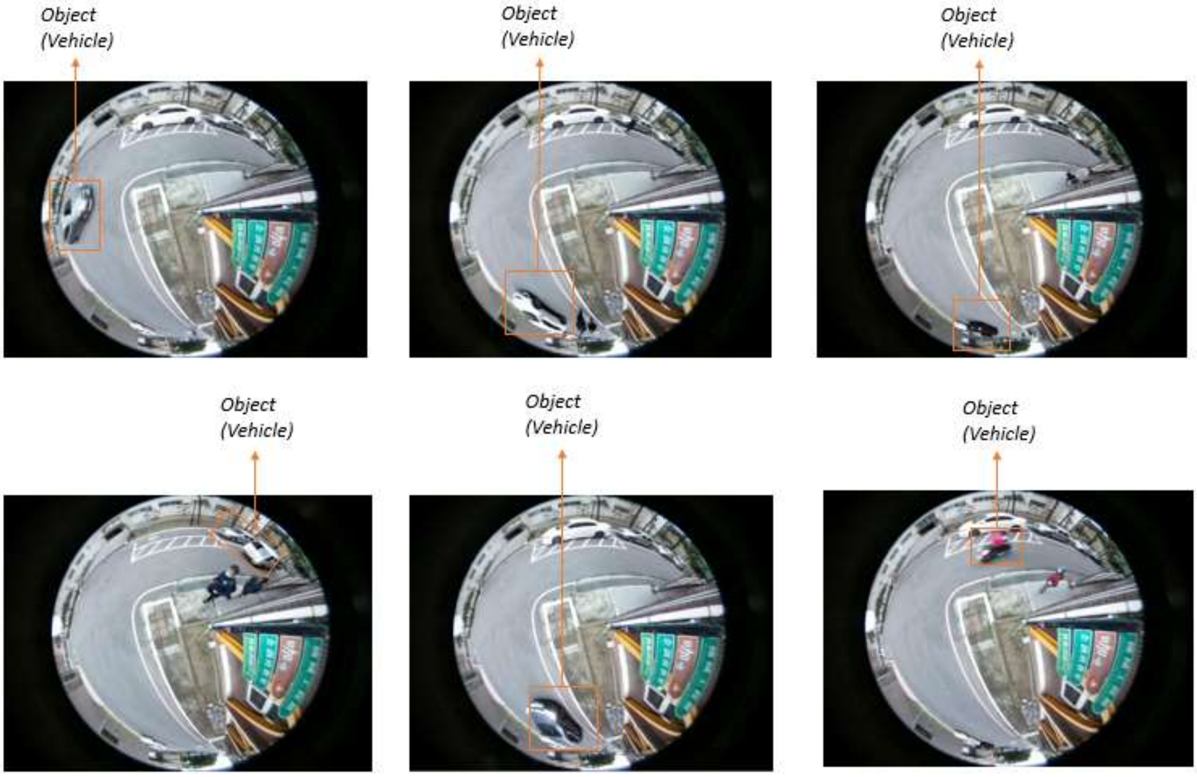


Figure 4. 4 Collect the datasets outside laboratory using fisheye camera

The recording results will be processed to proceed to the next step, such as dividing the fisheye view into two rectilinear views called front (inside) and back (outside) views, which automatically detect the vehicle plate object when it passes the camera. Then, from the plate detection processing on the vehicle, the application will automatically process it further by taking the original image, where this process will facilitate the application in vehicle character detection. Can be seen below for more details on each step process carried out by the application.

This dataset is the result of applying fisheye camera images, which will later be processed into two different views, namely the front view (inside) and the back view (outside), where the original image is the result of a system algorithm that saves automatically. When the system detects an object, both in the front view area (inside) and rear view area (outside), as shown in the image of the section 4.5 A Rectilinear View below.

4.2 A Rectilinear Views

The purpose of the monitoring system is to check the entry and exit of vehicles in the required areas, such as the research conducted in front of the Ming-Chi omnidirectional imaging laboratory, where many vehicles pass into the parking lot toward mechanical engineering. A reliable application is needed to detect vehicles passing through the area where the camera is installed. Various methods for detecting vehicles have been proposed [reff]. However, the standard has not been set. It is still a very interesting topic in this research because it is useful for supporting several technological advances in the future, such as large projects of the internet of things, smart city, and smart parking.

Propose a method that utilizes advanced image processing, requiring only one wide camera to cover the required area by dividing the coverage area with the method used to obtain a rectilinear view. It displays images with a rectilinear view that can minimize the number of cameras installed at each angle to get a different view over a large area. Use a camera with a wide-angle field of view of 180-220 degrees to improve the view. This method is done to determine each desired display size. As described in section 3.4, a rectilinear view is obtained by opening a cylindrical projection into a field of view for general perception. Performing this processing using the Moildev library and source-code can be seen below:

- *Create anypoint map for inside view;*

```
def anypoint_inside(self):
    """
    This function is implement anypoint inside map fisheye camera and make the object
     position become to center which will be show on the user interface.
    """
    self.anypoint_in = MoilUtils.remap(self.image, self.maps_x_in, self.maps_y_in)
    if self.anypoint_in is not None:
        self.num_in = self.num_in+1
    else:
        self.wind_detected_in.setText("No Detected")
        self.wind_detected_in.setText("No Image")

    MoilUtils.showImageToLabel(self.wind_inside_image, self.anypoint_in, 700)
```

- *Create anypoint map for inside view;*

```
def anypoint_outside(self):
    """
    This function is implement anypoint inside map in area fisheye camera
    """
    self.anypoint_out = MoilUtils.remap(self.image, self.maps_x_out, self.maps_y_out)
    if self.anypoint_out is not None:
        self.num_out = self.num_out+1
    else:
        self.wind_detected_out.setText("No Detected")
        self.wind_detected_out.setText("No Image")
    MoilUtils.showImageToLabel(self.wind_outsid_image, self.anypoint_out, 700)
```

As stated above, the Python programming source-code to create anypoint inside and anypoint outside functions in implementing the algorithm used to get two different views of a fisheye image, as shown in the image below;

4.2.1 Result of inside the laboratory

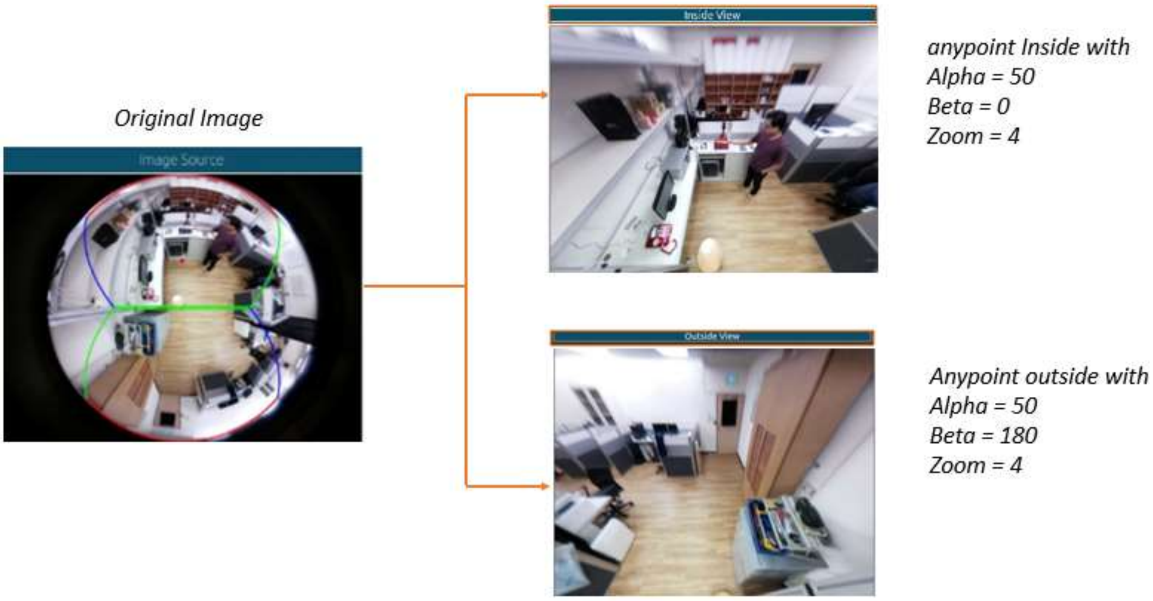


Figure 4. 5 The result of implement the algorithm (Inside Laboratory)

At this stage, the implementation of the monitoring system application will automatically save the vehicle plate image detected by the previous processing. With this, the system will easily recognize of an object vehicle plate, such as a fake vehicle plate used for the simulation process of detecting numbers and letters on vehicles, by creating a box divider on the vehicle as shown below:

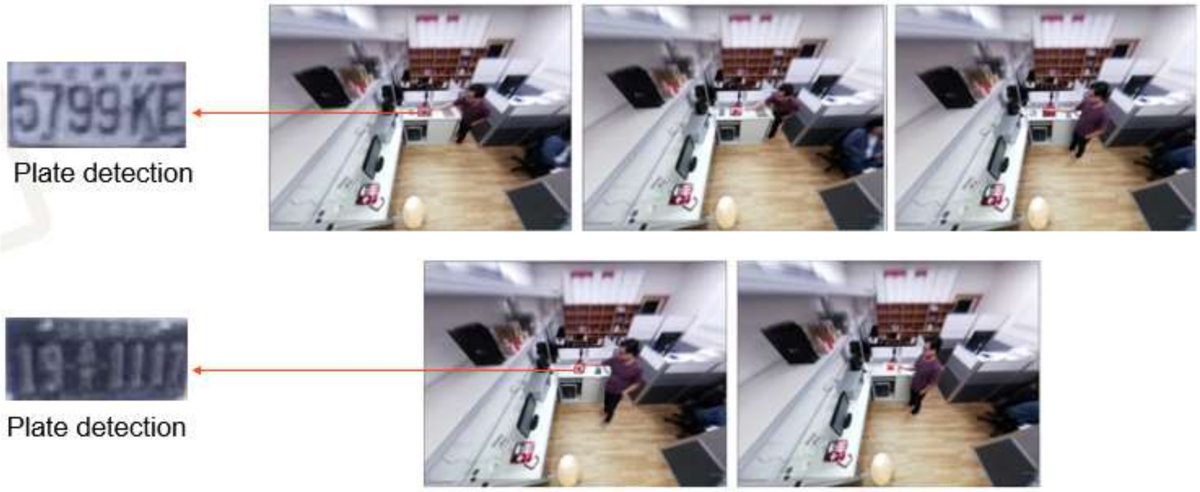


Figure 4. 6 The automatic plate Detection

As seen in Figure 4.6, the results of the detection using a vehicle plate simulation were carried out in a laboratory environment to test the YOLO 4 method in experiments with close and stable light conditions; when conducting experiments, several components need to be considered during this experiment; such as accurate predictions made in the monitoring system application using the YOLO 4 method when detecting vehicle plates automatically. The following can be seen in table 4.1, the detection prediction results are below:

Table 4. 1. The front view (Inside) | Result in laboratory

No	Plate Number	Pixel Image	Width	Height	Prediction (correct)
1.	5799KE	1295	1368	657	87.68%
2.	191117	800	1139	629	84.27%
3.	B6703WJF	340	654	287	80.41%
4.	AAA6385	720	773	340	76.56%
5.	BP1309GD	1265	1234	650	83.38%

Total vehicle plate used : 5
Total plate detection: 5

Table 4. 2 The back view (Outside) | Result in Laboratory

No	Plate Number	Pixel Image	Width	Height	Prediction (correct
1.	5799KE	1295	1368	657	87.35%
2.	191117	800	1139	629	81.03%
3.	B6703WJF	340	654	287	79.24%
4.	AAA6385	720	773	340	78.34%
5.	BP1309GD	1265	1234	650	80.15%

Total vehicle plate used: 5
Total plate detection: 5

Tables 4.1 and 4.2 show that of the 5 fake vehicle plates, which are referred to as the simulation process of testing methods from the inside view and outside view, the results of the prediction of each vehicle plate detected from the implementation of the YOLO 4 method are 81.46 percent for inside view and 80.35 percent for the outside view. This can be proven by the experiments that have been carried out.

4.2.2 Result of outside the laboratory

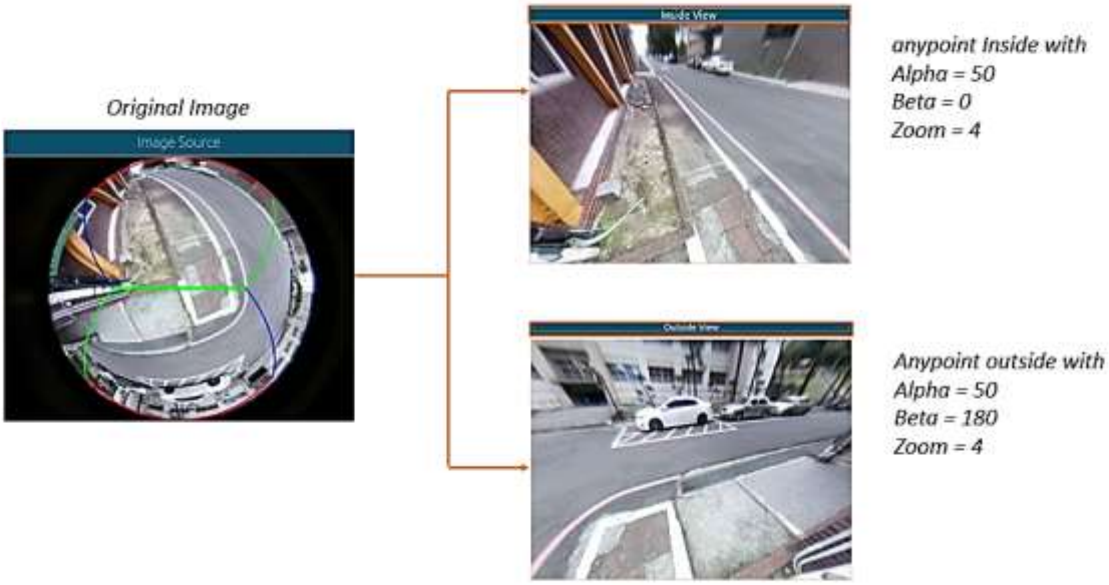


Figure 4. 7 The result of imlemenet the algorithm (Outside Laboratory)

Figure 4. 7 is the result of implementing alpha = 50, beta = 0, zoom 4 for the front (inside) rectilinear view and alpha = 50, beta = 180, zoom 4 for the back (outside) rectilinear view. To ensure that our suggested image processing method can be considered for calculating the number plates of identified vehicles, this technique is tested on images including objects of unknown size. An illustration of a fisheye camera capturing images with various fields of vision is shown in Figure 4.5. The camera is positioned in the center with a fixed viewing angle. As mentioned earlier, an image with a fixed optical axis position will result in a substandard rectilinear appearance. Further processing is needed to get a better rectilinear view of images like this one.

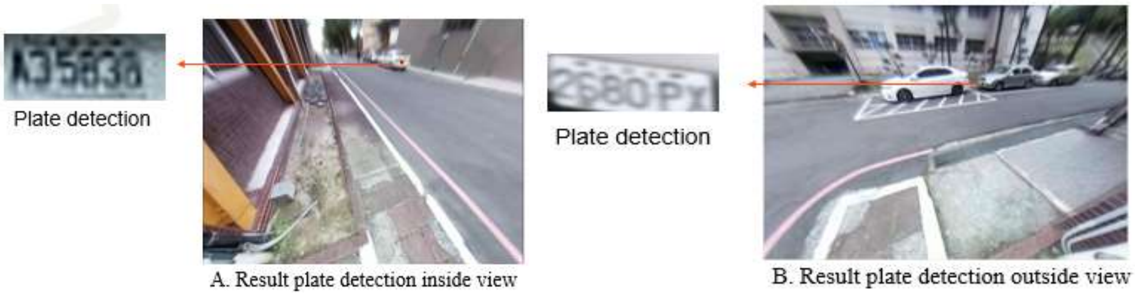


Plate detection
Plate detection
A. Result plate detection inside view
B. Result plate detection outside view

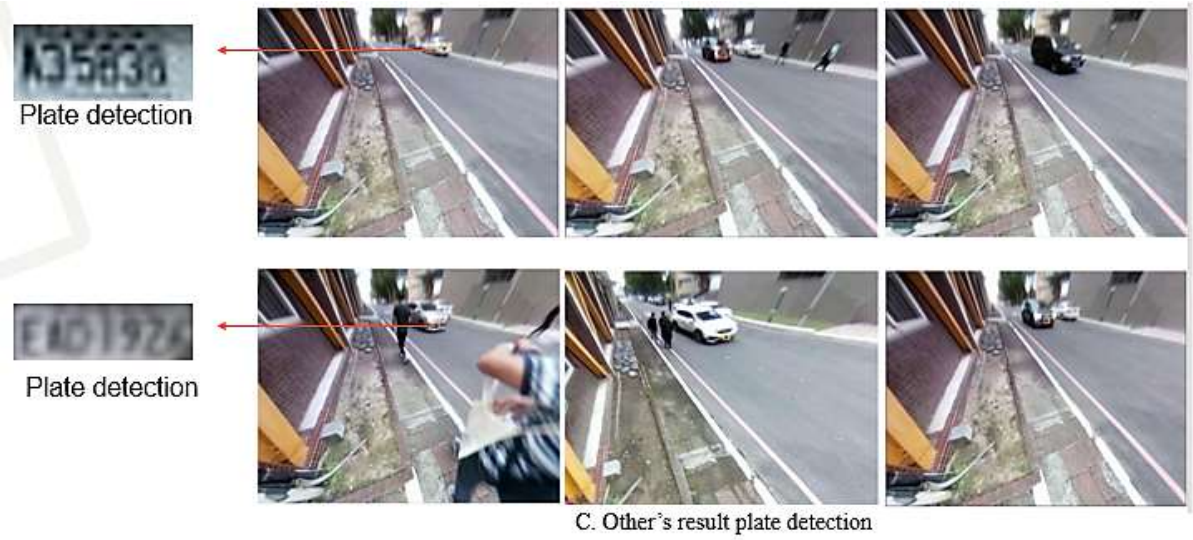


Figure 4. 8 (A), (B), (C). The result automatic plate detection

Figure 4.8 shows the current state of vehicle traffic; the experimental dataset retrieval was carried out for 31 minutes, during which the system recorded events as described in Chapter III, which discusses the performance flow of the system monitoring application when a vehicle object enters an area within the camera's range, which will automatically detect vehicle plates; This process is carried out by the system. by implementing the YOLO 4 method which is used as a vehicle control that predicts vehicle plates. As can be seen in the table of prediction results made by the system below:

Table 4. 3 The front view (outside) | Result outside laboratory

No	Plate Number	Pixel Image	Width	Height	Prediction (Correct)
1.	A35838	1295	417	657	87.69 %
2.	EAD1926	1295	550	564	84.50 %
3.	8F5701	760	460	635	83.31 %
4.	1532J6	620	476	649	80.03 %
5.	ATT87II	1265	439	480	85.47 %
6.	1451KD	637	494	615	87.15 %

Total vehicle entry: 9
Total plate detection: 6

Table 4. 4 The front view (outside) | Result outside laboratory

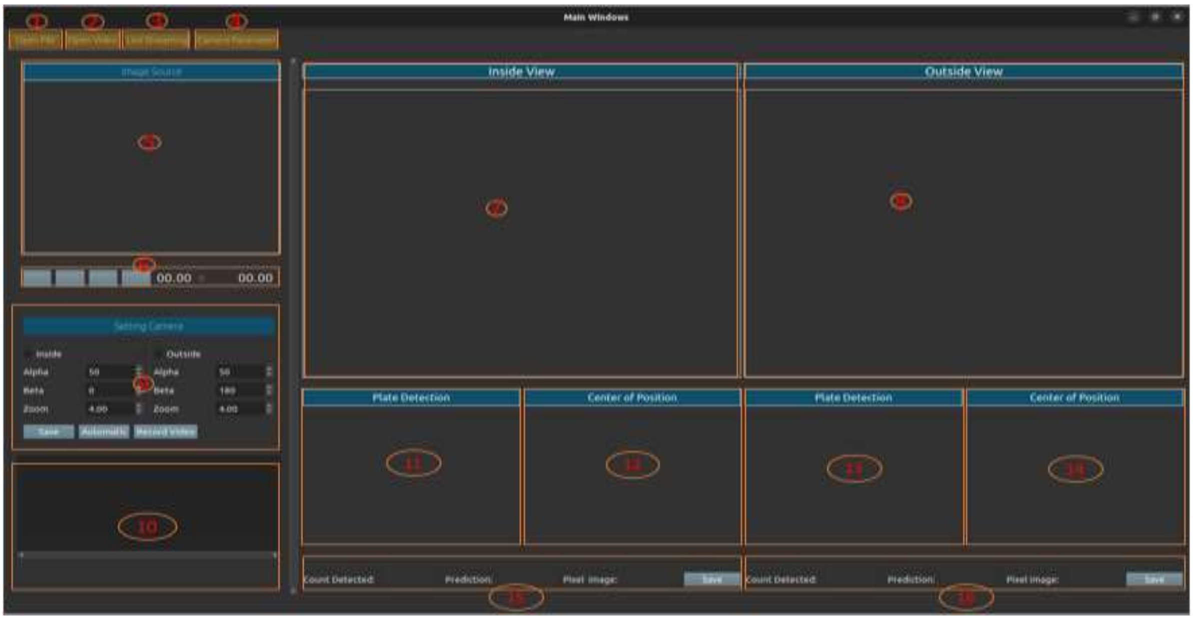
No	Plate Number	Pixel Image	Width	Height	Prediction (Correct)
1.	2680PX	1295	317	657	86.24 %
2.	BLA6058	1295	455	564	81.47 %
3.	7987RJ	1295	476	535	83.11 %

Total vehicle exit: 4
Total plate detection: 3

Tables 4.3 and 4.4 show the results of the prediction of each vehicle plate detected by the implementation of the YOLO 4 method, which successfully detects each traffic vehicle plate with predictions, as shown in the table above. However, in this study, which was conducted outside the environment, the YOLO 4 method could not read the vehicle plate character. The contrast of light caused this during the experiment, and the distance was too far from the vehicle, so the system carried out the image quality produced during the cropping process. This can be proven from the process and experimental results attached in the appendix below this thesis.

4.3 User Interface Design

In this study, we designed a prototype application for system monitoring. This application utilizes advanced image processing to produce a better perspective of the two views of the rectilinear shape. The prototype user interfaces for this application are shown in Fig.



○ Inside
Alpha 50
Beta 0
Zoom 4.00

○ Outside
Alpha 50
Beta 180
Zoom 4.00

Figure 4. 9 Design of the user interface application

As shown in Figure 4.9, a user interface design must be known by the user when using the monitoring system application. To access more in-depth information about each component of the application's user interface, making it easier for users to use each component and every important function. The user interface for the vehicle plate monitoring system application is shown in the following table.

Table 4. 5 Explanation of user interface application

1.	Open File	9.	Setting Camera
2.	Video	10.	Result Views
3.	Streaming	11.	Object Detection Inside View
4.	Cam Parameter	12.	Center of Position Inside View
5.	Image Source	13.	Oject Detection Inside View
6.	Button Play Video	14.	Center of Position Outside View
7.	Anypoint Inside View	15.	Fitur for Result Inside View
8.	Anyoint Outside View	16.	Fitur for Result Outside View

Explanation;

- Open File

  This application has an "open file" button in the first menubar that may be used to load an image file and display the source and original image.
- Video

  The video button in this application is used to load the video file that will be played as input for image processing carried out by the system, as seen in the second menubar display.
- Streaming

  The streaming button in the third menu bar display starts streaming video from the camera, which will then process the image directly.
- Cam Parameter

  The cam parameter button is located in the last menu bar display and is used to add, modify, and remove the camera parameters used.

- Image Source

  When a picture or video is loaded, the program displays the original fisheye version of it in the image source area.

- Button Play Video

  The play button can play, pause, preview, fast-forward, and slide as the seconds begin and end when a video is loading.

- Inside View

  This image was taken from a fisheye image that uses alpha 50, beta 0, and zoom 4. The inside view is a place to present an image or video that has been made rectilinear.

- Outside View

  The outside view is a place to display images or videos that have been made rectilinear, this image is taken from a fisheye image that implements alpha 50, beta 180, and zoom 4.

- Setting Camera

  To adjust the display of the resulting image or video, in the settings menu, there are several configurations such as alpha, beta, zoom for inside view and alpha, beta zoom for outside view, where the configuration is equipped with saving, automatic, and record buttons to save the video at the moment. Dataset retrieval takes place.

- Scroll Area

  In the result view, the system will display the results of application processing, such as the number of vehicles detected, image descriptions, and the accuracy carried out at the time of vehicle plate recognition.

- Plate Detection Inside

  In this section, object detection inside will display an image of a plate that has been cut according to the shape of the vehicle plate automatically during processing. The aim is to facilitate plate recognition.

- Plate Detection Outside

  In this section, object detection outside will only display an image of the back plate of the vehicle plate automatically during processing. The aim is to re-assure the plate is detected from object detection inside and that it is easy to identify the plate.

- Center of Position Inside

  In this section, we will display the numbers and letters read from the vehicle plate detection in the inside view with the accuracy processed in the application. We will also display the pixels on the detection form in real-time from the method used.

- Center of Position Outside

  The recognition outside view will also display numbers and letters that are read from vehicle plate detection with accuracy processed in the application and will display pixels on the detection form in real-time from the method used.

- Fitur for Result Inside View

  This section will display the results of the processing that has been carried out by the application based on the inside view.

- Fitur for Result Outside View

  The outcomes of the application's processing, as seen from the outside view, will be shown in this section.

The results of all the processing on the monitoring system application can be seen in the screenshot of the user interface design, as shown in Figure 4.10 below.

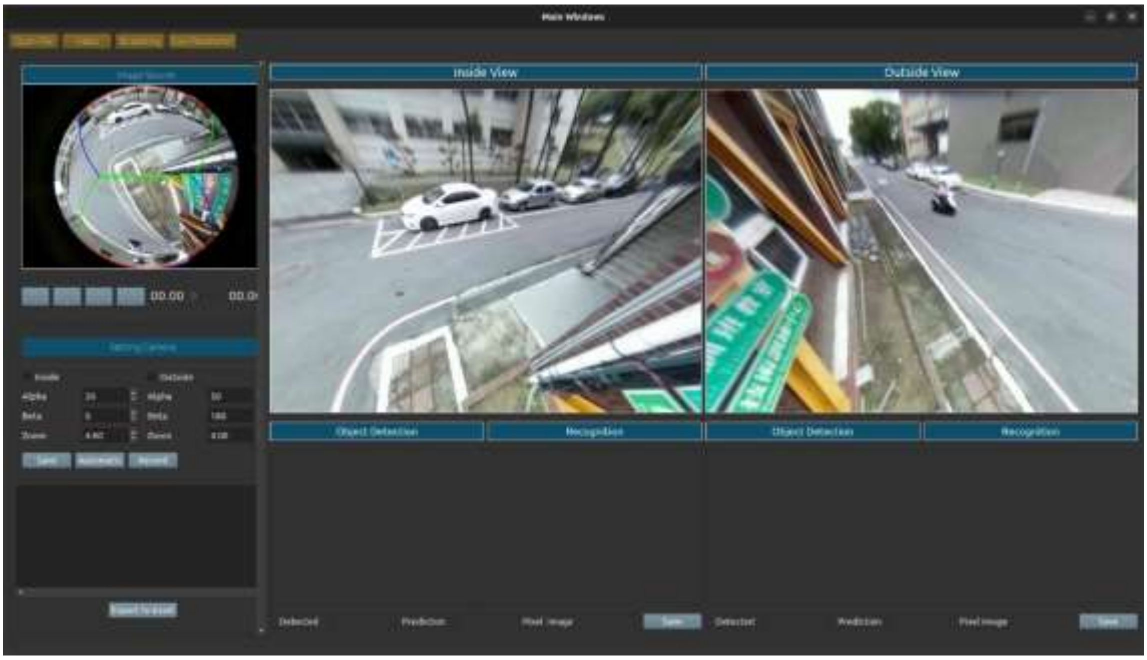


Figure 4. 10 Demonstration of monitoring system application

Chapter V Conclusion and Future Work

5.1 Conclusion

This study answers the problem of using many cameras, which is relatively expensive for ordering each camera in the form of a rectilinear with a not too wide Field of View (FoV) that is used in every corner of vehicle traffic by proposing a monitoring system equipped with a vehicle plate detection feature, which is proposed as an algorithm development. From Ming-Chi, Omnidirectional Imaging and Laboratory (MOIL) can use a fisheye camera known as a 180–220 degree wide angle that can cover every area. Moildev library was introduced, which will create a rectilinear image display according to the needs of the target, as described in section 3.4. A rectilinear view explains the original fisheye image into multi-views with different front (inside) and back (outside) views to minimize costs incurred to obtain each target angle by adopting the YOLO4 method as a trusted algorithm to be able to detect vehicle plates automatically as a feature applied to this application. In this experiment, the process is carried out in 2 stages. The first process is carried out in a laboratory environment with a short distance of 1-3 meters using 5 fake vehicle plates as a vehicle plate object detection test. The second experiment is carried out outside the laboratory at a distance of 1–30 meters on a vehicle object in a large area in a running vehicle state.

Based on the experiments conducted in the research, experiments in a laboratory environment with the parameters of detecting fake vehicle plates from the front view (inside), the predictive value was 76.46%, and the detection of fake vehicle plates from the back view (outside) was 80.35%. While in experiments carried out outside the front view (inside) and back (outside) views, the vehicle plate cannot be detected; this is because the resulting image quality is not good when outside this environment due to several parameters that affect image quality, such as light, speed of the vehicle, and the size of the vehicle plate, which is too small, causing the adopted YOLO4 algorithm to be unable to detect it. Besides that, the advantage of this research is that our proposed algorithm can see various sides of the angle without requiring many cameras to be installed in each corner.

5.2 Future Work

Our algorithm provides a monitoring system with multi-views taken from real fisheye images that can be used as a future solution for dealing with multiple cameras installed at every angle. As future work with the implementation of the proposed algorithm, we can add several complete features in addition to the advantages described in this study, such as automatic ticketing integrated with direct notification to the driver via the email address or other identity of the vehicle owner, recognition of entry plates, exits on toll roads with the working system of character recognition on the vehicle plate, as well as the security of parked vehicles equipped with artificial intelligence technology that reads every movement of objects detected on the camera, so that the standardization in the implementation of this research can be used in other locations to expand the development of findings that minimize low costs.

References

[1] M. A. S. S. J. H. A. K. I. K. M. G. A. A.-F. Ammar Gharaibeh, "Smart Cities: A Survey on Data Management, Security, and Enabling Technologies," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS,,* vol. 19, pp. 2456-2501, 2017.

[2] Myounggyu, "Intelligent Traffic Monitoring Systems for Vehicle Classification: A Survey," *IEEE Access,* vol. 8, pp. 73340-73357, Februari 2020.

[3] T. R. P. F. Remigiusz Baran, "A smart camera for the surveillance of vehicles in intelligent transportation systems," *Multimed Tools Appl,* p. 10471–10493, 2016.

[4] I. E. A. I. D. P. V. L. E. K. Christos-Nikolaos E. Anagnostopoulos, "License Plate Recognition From Still Images and Video Sequences: A Survey," *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS,* vol. 9, pp. 377-391, Sept 2008.

[5] T. W. a. B. Rinner, "Security and Privacy Protection in Visual Sensor Networks: A Survey," *ACM Computing Surveys,* vol. 47, pp. 2:1-2:42, April, 2014.

[6] L.-C. C. D.-Y. C. Jun-Wei Hsieh, "Symmetrical SURF and Its Applications to Vehicle Detection and Vehicle Make and Model Recognition," *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS,* vol. 15, pp. 6-20, Feb 2014.

[7] C.-J. C. GWO-Jen Jan, "Method for Exploring ViewPoint and Focal Length of Camera". Taiwan Patent US 6,985,183 B2, 10 January 2006.

[8] Y. L. L.-M. D. a. H. M. Hanxiang Wang, "Robust Korean License Plate Recognition Based on Deep Neural Network," *MDPI - Publisher of Open Access Journals,* pp. 1-18, 2021.

[9] C.-Y. W. H.-Y. M. L. Alexey Bochkovskiy, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *Semantic Scholar (arXiv),* pp. 1-17, 2020.

[10] S.-K. K. S.-H. L. T.-M. L. a. J. L. Jung-Hwan Kim, "License Plate Detection and Recognition Algorithm for Vehicle Black Box," *Proceedings of the 2017 International Automatic Control Conference (CACS),* pp. 1-6, Nov 2017.

[11] J.-Y. S. S.-h. P. Jeong-ah Kim, "Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition," *In Proceedings of the 2020 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia) ,* pp. 1-3, Nov 2020.

[12] L.-S. C. Y.-C. C. a. S.-W. C. Shyang-Lih Chang, "Automatic License Plate Recognition," *IEEE Transactions on Intelligent Transporttation System,* pp. 42-53, Upload by Sei-Wang Chen on September 2014.

[13] H. Z. a. M. K. Michael Hodlmoser, "An Evaluation of Camera Calibration Methods Using Digital Low Cost Cameras," in *Computer Vision Winter Workshop 2010, Libor Spa 'cek, Vojt' ech Franc*, Austria, 2010.

[14] E. Ozcakir, "Camera Calibration With OpenCV," *Access on April 27, 2022, Available on: https://medium.com/@elifozcakiir/camera-calibration-with-opencv-9fb104fdf879,* Sep 18, 2020.

[15] A. M. R. S. Davide Scaramuzza, "A Toolbox for Easify Calibrating Omnidirectional," in *Iros*, Beijing, China, 2006.

[16] C. Mei, "Central Catadioptric System," Inria, [Online]. Available: http://www-sop.inria.fr/icare/personnel/Christopher.Mei/ChristopherMeiPhDStudentToolbox.html. [Accessed 22 May 2021].

[17] OpenCV, "Camera Calibration," [Online]. Available: https://docs.opencv.org/master/dc/dbb/tutorial_py_calibration.html. [Accessed 22 May 2021].

[18] C.-J. C. GWO-Jen Jan, "Method for Exploring ViewPoint and Ffocal Length of Camera". Taiwan Patent US 6,985,183 B2, 10 January 2006.

[19] B. L. a. M. P. Lionel Heng, "CamOdoCal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. pp. 1793-1800, 2013.

[20] V. Tyagi, Understanding Digital Image Processing, Guna (MPI), India: Department of Computer Science and Engineering Jaypee University of Engineering and Technology Raghogarh, 2018.

[21] G. L. Loay Alzubaidi, "Real time License Saudi Plate Recognition Using Raspberry Pi," *International Journal of Advanced Trends in Computer Science and Engineering,* vol. 8, pp. 42-47, 2019.

[22] G. S. Kothman, "U.S. Patent and Trademark Office," no. U.S. Patent No. D163,328, 1951.

[23] S. A. L. P. R. System, "Muhammad Sarfraz, Mohammed Jameel Ahmed, and Syed A. Ghazi," *Proceedings of the 2003 International Conference on Geometric Modeling and Graphics (GMAG'03) IEEE,* pp. 1-6, 2003.

[24] H. S. W. Z. L. L. Lin Luo, "An Efficient Method of License Plate Location," *International Conference on Information Science and Engineering (ICISE),* pp. 770-773, 2009.

[25] B. A. H. R. S. B. Reza Azad, "Real-Time and Efficient Method for Accuracy Enhancement of Edge," *International Conference on computer, Information Technology and Digital Media (CITADIM Proceeding-Scientific),* pp. 146-155, 2013.

[26] "Towards End-to-End License Plate Detection and Recognition: A Large Dataset and Baseline," *Springer, Computer Vision Foundation (ECCV),* pp. 1-17, 2018.

[27] P. W. n. C. S. Hui Li, "Towards End-to-End Car License Plates Detection and Recognition with Deep Neural Networks," *IEEE Transactions on Intelligent Transportation System ,* pp. 1-9, Dec 2017.

[28] D. A. D. E. C. S. S. R. C.-Y. F. A. C. B. Wei Liu, "SSD: Single Shot MultiBox Detector," *arXiv:1512.02325v5 [csCV],* pp. 1-17, 2016.

[29] K. H. R. G. a. J. S. Shaoqing Ren, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* vol. 60, pp. 1137-1149, June 2017.

[30] K. S. A. V. A. Z. Max Jaderberg, "Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition," *arXiv:1406.2227v4 [cs.CV] 9 Dec 2014,* pp. 1-10, Dec 2014.

[31] L. Z. X. L. X. W. D. T. Fangfang Wang, "Geometry-Aware Scene Text Detection with Instance Transformation Network," *IEEE/CVF Conference on Computer Vision and Pattern Recognition,* pp. 1381-1389, 2018.

[32] "Raspberry Pi," *Access on April 24, 2022, Available:https://en.wikipedia.org/wiki/Raspberry_Pi,* June 24, 2019.

[33] G. Halfacree, "Raspberry Pi Beginner's Guide," *Publish in 2020 by Raspberry Pi Trading Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS,* no. ISBN: 978-1-912047-73-4, pp. 1-248, 2020.

[34] L. Hattersley, "How to set up Raspberry Pi 4," *Access on: April 27 2022, Available on : https://magpi.raspberrypi.com/articles/set-up-raspberry-pi-4,* 2019.

[35] "Raspberry Pi Documentation," *Access on April 24, 2021 Availabel: https://www.raspberrypi.com/documentation/accessories/camera.html.*

[36] G. k. surabaya, "Fisheye Lens Alias Artistic Fisheye Lens," *Access on April 26, 2022 Available: https://www.gudangkamera.net/blogs/21-lensa_fisheye_alias_lensa_mata_ikan_yang_artistik.html,* 2017.

[37] OpenCV, "About OpenCV Library," [Online]. Available: https://opencv.org/about/. [Accessed 15 June 2021].

[38] Haryanto, "Moildev, Python Package Index (PyPI)," Perseverance Technology, Ming Chi University of Technology, Taiwan, [Online]. Available: https://pypi.org/project/Moildev/. [Accessed 16 June 2021].

[39] A. F. Jeseph Redmon, "YOLOv3: An Incremental Improvement," *arXiv.1804.02767,* pp. 1-6, 2018.

[40] J. G. Z. H. a. J. W. Jing Li, "Application Rresearch of Improved YOLO V3 Algorithm in PCB Electronic Component Detection," *Applied Sciences MDPI journal,* pp. 1-21, 2019.

[41] Y. J. Z. W. Z. C. T. H. Jiahui Yu, "UnitBox: An Advanced Object Detection Networkk," *arXiv:160801471v1,* pp. 1-5, 2016.

[42] N. T. J. Y. G. A. S. I. R. S. S. Hamid Rezatofighi, "Generalized Intersection Over Union: Matric and A Loss for Bounding Box Regression," *arXiv:1902.09630v2,* pp. 1-9, 2019.

[43] J. Redmon∗, "You Only Look Once Unified, Real-Time Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition ,* pp. 779-788, June 2016.

[44] A. F. Joseph Redmon, "YOLO9000: Better, Faster, Stronger," *IEEE Conference on Computer Vision and Pattern Recognition,* pp. 6517-6525, July 2017.

[45] D. A. D. E. C. S. Wei Liu, "SSD: Single Shot MultiBox Detector," *In Proceedings of the European Conference on Computer Vision (Springer International Publishing: Cham, Switzerland),* pp. 11-14, Oct 2016.

[46] H. P. D. M. a. C. P. Jithmi Shashirangana, "Automated License Plate Recognition: A Survey on Methods and Techniques," *IEEE Access ,* vol. 9, p. 11203–11225, Dec 2020.

Appendix