Automated License Plate System Recognition for Campus Gate System

BY

LIEW VOON CHOON

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) COMPUTER ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Automated License Plate System Recognition for Campus Gate System

Academic Session: 2023/2024

I LIEW VOON CHOON

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature) (Supervisor's signature)

Address:
NO 40, HALA KLEDANG 4,
TAMAN SEPAKAT, 31450
MENGLEMBU.PERAK

Ts. Dr. Vikneswary Jayapal
Supervisor's name

Date: 23/4/2024

Date: 24/04/24

FACULTY/INSTITUTE* OF Information and Communication Technology

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 23/4/2024

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ***LIEW VOON CHOON*** (ID No: ***21ACB04376***) has completed this final year project/ dissertation/ thesis* entitled "*Automated License Plate System Recognition for Campus Gate System*" under the supervision of Ts Dr Vikneswary a/p Jayapal (Supervisor) from the Department of Computer and Communication Technology Faculty/Institute* of Information and Communication Technology

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

LIEW VOON CHOON

DECLARATION OF ORIGINALITY

I declare that this report entitled "**AUTOMATED LICENSE PLATE SYSTEM RECOGNITION FOR CAMPUS GATE SYSTEM**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ______________________

Name : LIEW VOON CHOON

Date : 23/4/2024

ACKNOWLEDGEMENTS

I extend my sincere gratitude and appreciation to my supervisor, Dr. Vikneswary a/p Jayapal, for providing me with the opportunity to work on the Automated License Plate Recognition System for the Campus Gate System. This project marks my initial foray into the field of object detection and recognition, setting the foundation for my future career endeavors. I am immensely grateful to you.

Additionally, I would like to express my heartfelt thanks to my parents and family for their unwavering love, support, and encouragement throughout this journey.

ABSTRACT

This project involves an license plate detection and character recognition system developed for the campus gate system. Commonly referred to as the Automated License Plate Recognition System (ALPR), it proves valuable in various scenarios, including monitoring vehicle entry and exit on university campuses. Following the implementation of this project on campus, it successfully reduced the need for manpower, streamlined traffic flow control during peak hours, and prevented unregistered or uncategorized vehicles from entering the university premises. An ALPR system typically involves three fundamental actions: license plate detection, image processing, and license plate character’s recognition.

Primary objective of this project was to enhance recognition accuracy, angle checking, and handling of unconstrained scenes, oblique views, and more. This was achieved through a comprehensive evaluation of the developed ALPR system's performance. Given the remarkable advancements in machine learning today, this project proposes the utilization of YOLO, GAN methods and other tools for object detection, along with EasyOCR for optical character recognition. These technologies collectively extract characters and numbers from license plates to yield the final result.

The project primarily focuses on Malaysia's regional license plates, which adhere standards license plate format set by the Malaysia Road Transport Department (JPJ). However, some special cases may arise, such as logos positioned beside the license plate, or other factors that could potentially influence the accuracy rate of license plate detection and the availability of license plate character recognition. To facilitate this research, a dataset containing 1200 images of Malaysia's license plates was employed. This dataset encompasses challenging license plate images from various areas and acquisition states.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

%	percentage

LIST OF ABBREVIATIONS

ALPR	Automated License Plate Recognition
LP	License Plate
JPJ	Malaysia Road Transport Department
LPD	License Plate Detection
ANN	Artificial Neural Networks
Seg	Segmentation
CR	Character Recognition
OCR	Optical Character Recognition
CNN	Convolutional Neural Network
Fig	Figure
CR	Character Recognition
YOLO	You Only Look Once
UTAR	University Tunku Abdul Rahman
IOU	Intersection Over Union
mAP	mean Average Precision
FPS	Frames Per Second
LR	Low Resolution
HR	High Resolution
PNSR	peak signal-to-noise ratio
SSIM	Structural Similarity Index
MSE	Mean Square Error

Chapter 1

Introduction

An Automated License Plate Recognition (ALPR) system constitutes an image processing methods that employs cameras plus specialized software to automatically detect, capture, process, recognize, and analyse the license plates of vehicles. ALPR systems have found extensive applications, including automatic toll collection, traffic management, automated gate systems, and parking management. They have become increasingly crucial due to rising traffic congestion, incidents of traffic violations, and unauthorized vehicle access. While a range of ALPR systems has been created to tackle these obstacles, in particular traffic surveillance systems, vehicle tracking systems, and vehicle speed detection, ALPR systems remain a pivotal topic in the development of intelligent infrastructure.

Typically, an ALPR system comprises three main fundamental phases. The first phase involves image pre-processing, where images are detected, captured, and then subjected to color space conversion, image resolution resizing, and noise reduction. In the second phase, license plate localization occurs, discovering the region of interest upon the vehicle where this license plate is located, using characteristic and feature-based methods. Indeed, the final phase of the process involves optical character recognition (OCR), where the letters or characters on the processed license plate are read and identified. OCR is widely regarded as the most crucial component related to Automatic License Plate Recognition (ALPR) system.

The title associated with the project is “Automated License Plate Recognition System for Campus Gate System” focusing on enhancing security within the university campus area. According to the UTAR Vehicle Parking Rules, only vehicles with a valid UTAR sticker are allowed to enter the university compound. Despite the stipulated requirement for a UTAR sticker to access the campus area, security guards are still required to manually inspect each vehicle's sticker, a process that consumes significant time, particularly during peak traffic hours such as 8:00 am morning classes. Furthermore, due to oversights, vehicles without the required vehicle sticker sometimes gain access to the campus area. Such occurrences can lead to parking violations, compromised campus safety, unauthorized access, and even vehicle theft. Moreover, counterfeit stickers can be designed to resemble legitimate campus vehicle stickers. Since security guards allow vehicles with valid stickers to enter the campus area, they may inadvertently permit vehicles with counterfeit stickers to enter as well. From a safety perspective, security guards are required to conduct spot checks of vehicle stickers in the university campus area. If this situation arises, the process of checking every vehicle in the campus area one by one and distinguishing between valid and fake stickers can be time-consuming and inefficient.

By considering the paramount importance of safety and security, the implementation of the Automated License Plate Recognition (ALPR) system in conjunction with the gate system serves as a robust and proactive measure. The ALPR system's integration with the gate system results in a sophisticated solution that enhances the overall safety framework of the university campus. This integration involves strategically positioning high-resolution cameras equipped with advanced image processing capabilities at key entry and exit points.

In cases where a vehicle lacks the required UTAR sticker or has a suspicious license plate, the ALPR system triggers an immediate alert to the security personnel. This enables timely intervention by the security team, ensuring that only authorized vehicles gain access to the campus premises. The automated nature of the ALPR system significantly reduces the need for manual inspection of each vehicle's sticker, thus optimizing the security guard's efficiency and effectiveness during peak traffic hours.

Moreover, the ALPR system's continuous monitoring and data logging contribute to a comprehensive security record. In the event of an incident or security breach, the system provides a valuable resource for retrospective analysis and investigation. This historical data can aid authorities in identifying any patterns of unauthorized access, enabling them to take appropriate measures to prevent future occurrences.

Ultimately, the ALPR system's integration with the gate system aligns with the university's commitment to safeguarding its campus community. The technology not only streamlines the vehicle access process but also fortifies the campus against potential security threats. It is a testament to the institution's dedication to granting protected and unharmed surroundings for students, faculty, staff, and guests alike.

1.1 Problem Statement and Motivation

In contemporary times, numerous educational institutions dealing with daunting challenge of effectively managing parking and traffic flow within their campus areas. These challenges culminate in issues such as congestion, safety hazards, and inconveniences experienced by students, staff, and visitors. Conventional parking systems, notably those reliant on parking attendants and vehicle stickers, often prove to be inadequate and financially burdensome, thereby placing operational and fiscal pressures on the student community.

A key aspect in ensuring the seamless operation of the ALPR system lies in the standardization of vehicle plate formats. Consequently, vehicle plates have been streamlined into two distinct configurations, with the alphabet or letters arranged either in a single or double row. This standardization effort, particularly in compliance with the regulations set forth by the JPJ, is crucial for maintaining the efficiency of the ALPR system. Illustrative examples of these standardized vehicle plate formats, as endorsed by the JPJ, can be found in Figure 1.1.1.

By aligning with these standardized formats, the ALPR system is optimally equipped to tackle the intricacies of vehicle plate recognition and verification. In doing so, the system can seamlessly contribute to the enhancement of campus gate management, offering a technologically advanced and efficient solution that ensures not only convenience but also heightened security for all stakeholders involved.

Figure 1.1.1 The sample form of vehicle plate allowed in Malaysia.[22]

Acknowledging the importance lies in recognizing that the configurations or designs of license plates may differ based on the geographical area of vehicle registration. The popularity of bespoke license plate characters introduces an added layer of complexity to the operation of the ALPR system. These custom license plate numbers frequently deviate from the standardized format, often incorporating fewer characters and unique spacing arrangements. To better visualize this, please refer to Figure 1.1.2, which highlights instances of custom vehicle plates that substantially differ from the established standardized license plate configuration. Notably, the performance of character segmentation is adversely impacted by the considerable spacing between characters on the LP.

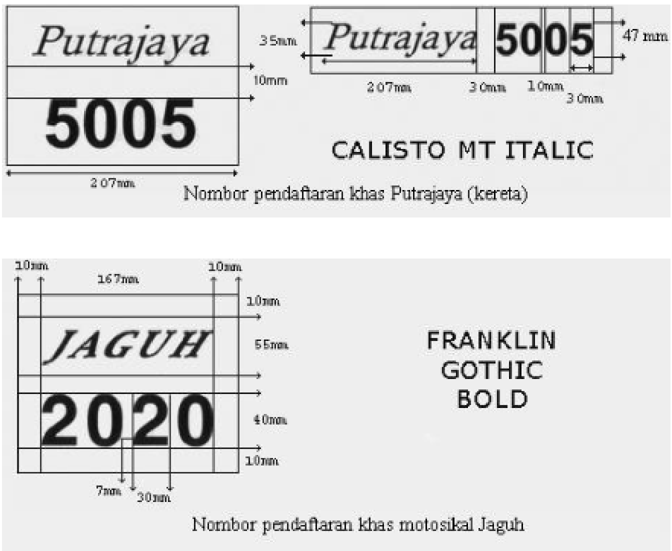


Figure 1.1.2 Example of custom vehicle plates [13]

1.1.1 Motivations

In response to the challenges, the proposition of an ALPR system for campus gate system emerges as an innovative and practical solution. This automated system presents a heightened level of reliability, efficiency, and security, constituting a viable alternative to traditional gate management approaches. By seamlessly integrating automated processes encompassing detection, capture, processing, recognition, and analysis of vehicle license plate information, the ALPR system inherently possesses the capability to rigorously verify vehicle identities and pertinent details. This verification mechanism serves as a robust deterrent against unauthorized access, significantly elevating the overarching security and safety measures within the campus environment.

1.2 Research Objectives

In this project, three objectives have been identified.

1. To investigate license plates in terms of character segmentation stage

In Malaysia, the JPJ has standardized the form and style of license plates, but there are instances where some vehicle owners do not adhere to these regulations. License plates can deviate from the standardized format in various ways. For instance, if the registration number on the license plate is written too far apart, it can lead to erroneous recognition by the Optical Character Recognition (OCR) system. On the flip side, performance during the character segmentation stage might suffer if characters are positioned too closely together. This is exemplified in Figure 1.1.2, which displays samples of custom license plates.

2. To develop an ALPR system that can enhance the performance of character segmentation stage.

Creating an ALPR system with an improved character segmentation stage demands a thorough strategy that covers various critical elements to guarantee precise extraction of alphanumeric characters from license plate images. Character segmentation stands as a pivotal phase, acting as the foundation for precise character recognition. To achieve an advanced character segmentation process, a range of strategic steps and considerations can be employed:

1. **Image Preprocessing:** Prior to character segmentation, effective image preprocessing techniques must be applied. These techniques encompass procedures such as noise reduction, contrast enhancement, and resizing. Addressing these factors helps to mitigate the impact of varying lighting conditions, thereby enhancing the overall quality and clarity of license plate images.
2. **Integration of Object Detection Algorithms:** The incorporation of object detection algorithms like YOLO, as employed in this project, can significantly aid in initial character localization. These algorithms detect and create bounding boxes around license plate characters, providing a solid foundation for subsequent segmentation processes. Following this, contour analysis techniques can be implemented to segment characters based on identified contours within the license plate region.

3. **Dataset Augmentation:** Enhancing the training dataset through augmentation techniques is valuable. This includes incorporating variations such as rotations, scaling, and mirroring to add diversity to the dataset. Augmentation not only enriches the training data but also strengthens the model's proficiency to generalize over diverse license plate images.
4. **Additional of super-resolution methods:** Super-resolution methods such as SRGAN and ESRGAN will be deployed to enhance the license plate quality when detected by object detection and passing to OCR for recognition.

In conclusion, optimizing the character segmentation stage of an ALPR system demands a multifaceted approach that encompasses image preprocessing, advanced segmentation techniques, object detection algorithms, and thorough validation. By systematically addressing each facet, the system can achieve heightened accuracy and precision in character recognition, thus elevating the comprehensive effectiveness of the ALPR system.

3. To evaluate the performance of the developed ALPR system

The object detection algorithms applied in this project are YOLOv5, YOLOv7 and YOLOv8. All these models can be trained using a custom dataset for license plate detection. After pre-training both models, their performances will be evaluated to achieve the desired results. However, different methods of GANs will added to compare the effectiveness and accuracy for different models.

The performance tests to be conducted include:

a. Diverse shooting angles
b. Varying illumination conditions
c. Complex surroundings
d. Different weather conditions
e. Image distortion
f. Degraded image quality
g. Varied lighting conditions

These performance tests are essential to evaluate how well these models perform in different conditions and scenarios when detecting license plates. Any occurrence of the mentioned issues can potentially impact the accuracy of the ALPR system, leading to traffic congestion within the university campus.

1.3 Project Scope and Direction

This project aim to create ALPR system for the UTAR campus gate system with the goal of mitigating traffic congestion during morning hours and reducing the need for manual vehicle checks at the guardhouse. By comparing pre-trained models, YOLOv5, YOLOv7 and YOLOv8, this project seeks to evaluate the efficiency of these models in object detection. The comparison between all these models is facilitated by utilizing a custom dataset. Additionally, the project aims to address the objectives outlined in Chapter 1.2, aiming to enhance and develop a high-performance ALPR system that can be implemented at UTAR to enhance overall security and safety.

Additionally, GAN methods will be added to the final chosen YOLO model to compare it with different models with and without GAN methods. This comparison will provide comprehensive results for further analysis.

Moreover, the scope and direction of this project encompass the development of an advanced ALPR system, utilizing the latest generation of algorithms and incorporating compelling prototypes to elevate the system's intelligence and capabilities in object detection and recognition. Nevertheless, it's critical to be aware that this project will not extensively focus on the Optical Character Recognition (OCR) tool used for character recognition. This decision is rooted in the comparison of performance metrics. Notably, EasyOCR showcases an overall performance surpassing a 95% accuracy rate in predicting license plate numbers, whereas Tesseract OCR demonstrates an accuracy rate of 90%. This substantiates the conclusion that EasyOCR excels in real-time license plate recognition, as illustrated in Fig 1.3.1, which displays the testing results of using EasyOCR and Tesseract OCR [2].

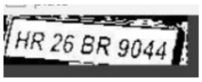
Parameter	Test Data 1	Test Data 2
Original Image		
Character Recognition using EasyOCR	'HR.26 BR.9044', 0.5167916087717548)]	'POLBCAF5030', 0.30806908725519727)]
Character Recognition using Tesseract		

Fig 1.3.1 Testing of EasyOCR and Tesseract OCR [2]

1.4 Contributions

The project titled "Automated License Plate System Recognition for Campus Gate System" primarily aims to benefit universities and other educational institutions. The implementation of this project holds the potential to alleviate several challenges, including reducing manpower requirements, enhancing operational efficiency by minimizing manual labor, elevating safety and security levels within the campus area, and mitigating traffic congestion. These issues are prevalent across educational institutions, particularly during periods of peak traffic congestion.

The fundamental process of the ALPR system involves license plate detection, image processing, and license plate recognition. Through this project, we are capable of evaluating and substituting the models utilized for license plate detection to identify the most precise and high-performance model. This evaluation process is instrumental in optimizing the system's accuracy and overall effectiveness.

The ALPR system is an object detection model that holds versatile applicability. Its usage extends beyond the campus environment, encompassing various other applications such as parking management, traffic management, toll collection, law enforcement, and security. This implies that successful implementation of this project can yield benefits across diverse scenarios where the ALPR system is employed.

At its core, this prototype of the ALPR system lays the groundwork for its adaptation to distinct situations, contributing to the realization of an intelligent world. Its potential spans a wide spectrum of use cases, underscoring its adaptability and relevance across different contexts.

1.5 Report Organization

The report is formatted to five chapters. Chapter 1, Introduction, covers the project's problem statement, background, motivation, scope, objectives, contribution, highlights of achievements, and organization of the report. In Chapter 2, Literature Review, various existing ALPR system methods, YOLO architecture, and GAN methods are examined. Chapter 3, System Design, elaborates on the overall design of the project. The evaluation and experiments conducted are detailed in Chapter 4, System Evaluation and Discussion. Finally, Chapter 5, Conclusion, offers a summary of the project's findings and conclusions. This format facilitates a thorough exploration of the project's background, methodologies, results, and implications.

Chapter 2

Literature Review

2.1 Automated License Plate System Recognition for Campus Gate System

The utilization of ALPR systems for campus gate systems has gained significant traction within the university landscape. This surge in adoption can be attributed to their multifaceted advantages, encompassing efficiency, accuracy, and cost-effectiveness. These systems offer swift and automated access control, substantially curtailing waiting times even during periods of heightened vehicular traffic. Furthermore, they enhance the overall efficacy of gate systems by expediting checks and traffic control procedures.

Beyond the realm of efficiency, ALPR systems tailored for campus gate applications yield profound improvements in security. By establishing a dependable and automated mechanism for overseeing and regulating vehicle access, these systems bolster campus security measures. Upon entry or exit, the ALPR system promptly captures and records the vehicle's license plate, cross-referencing it with a database. This process effectively discerns suspicious and unauthorized vehicles, preventing their intrusion into the campus area.

Numerous techniques have been advanced in the pursuit of elevated accuracy and reliability in license plate recognition. This demonstrates the collective endeavor to refine and optimize these systems, ensuring their precision aligns with the security demands of modern university campuses.

2.1.1 Campus Vehicle Sticker

In order to control the number of vehicles entering the university campus, the management usually distributes campus vehicle stickers to manage parking spaces. Following the procedure of University Tunku Abdul Rahman, students are required to submit legal documents and information through the provided website. The staff then issues the campus vehicle stickers to students, considering several factors. Students are subsequently notified to collect their stickers at specific locations. Upon receiving the campus vehicle sticker, students must affix it to their vehicle's mirror, allowing security guards to easily conduct vehicle checks [25].

There are several terms and conditions that must be followed by the sticker holder for the campus vehicle sticker, aimed at reducing potential problems caused by students on campus. Firstly, students are not permitted to purchase campus vehicle stickers from sources other than UTAR. Additionally, campus vehicle stickers are non-transferable, including those that have expired [25].

However, the weakness of the campus vehicle sticker system lies in its cost to students. Campus vehicle stickers must be purchased, with fees varying based on the length of the semester, resulting in different prices. These additional fees can be burdensome for students already facing financial strain. Furthermore, if the number of vehicles on campus exceeds available parking spaces, even with the vehicle car sticker system in place, it may lead to frustration and competition for parking spots.

Nevertheless, the strength of the vehicle car sticker system lies in its ability to allow security guards to easily determine the legality of vehicles by checking the sticker on the mirror.

2.1.2 Parking Access Card

To acquire an access card, students and staff must fill out and provide the access card application document. The requests are typically handled within one working day [26]. These parking access cards grant entry and exit from designated gated facilities, allowing for Night or Weekend permissions. Parking access cards are automatically provided to faculty/staff and student permits assigned to a gated facility. Any other access cards are distributed upon inquiry [26].

The strength of using parking access cards is that it reduces the workload of security guards. They no longer need to check each vehicle for permits to enter the campus area. Security guards can now focus on controlling the crowd or traffic in the campus area. By using the parking access card, students can quickly enter and exit parking areas by holding the card near the reader. The barrier will open automatically.

However, the weakness of the parking access card system is that it requires a high implementation cost. To implement this system, the university needs to purchase necessary hardware and software such as barrier gates, access card readers, and the system or software to read the access cards.

2.1.3 Parking Credential

The parking credential is another effective method to regulate vehicle access to the campus area. It aims to provide fairness and convenience to all individuals on campus when obtaining parking. There are several types of parking credentials available:

1. **Visitor Parking Credential:** This type of credential is designated for visitors to the campus. It allows them temporary access to park their vehicles in specified areas.

2. **Vendor Parking Credential:** Vendor parking credentials are issued to vendors and suppliers who frequently visit the campus for deliveries or services. This credential grants them access to designated parking zones.

3. **Pick Up or Drop Off Parking Credential:** This specific credential is designed for individuals who need short-term parking for pick-ups or drop-offs. It allows quick and convenient access to designated areas for these purposes.

Each type of parking credential serves a distinct purpose, ensuring efficient and organized parking management on campus. However, the weakness of this method is that the management needs to monitor and control parking, as anyone can easily obtain the parking credential and enter the campus without returning the credential to the security guard. Additionally, this method can lead to traffic congestion because in order to obtain the parking credential, drivers need to provide their information such as car plate number, driving license, and identity card to the security guard. This is done to ensure the safety of the campus area when entering.

2.2 Review of the Current System / Applications

2.2.1 License plate detection

The author provides a clear explanation of YOLO (You Only Look Once) in [1]. One of the best examples of a one-stage object detector is YOLO, which is incredibly effective at identifying things in pictures or videos. Unlike its multi-stage counterparts, YOLO operates by swiftly detecting objects in a singular pass, resulting in enhanced speed and efficiency.

The essence of the YOLO algorithm lies in its ingenious division of an image within a grid. Within separate grid cell, YOLO dynamically forecasts bounding boxes and the chances of associated classes. This architectural choice equips YOLO with the unique capacity to detect multiple objects within a single image, all accomplished within a solitary forward pass of the neural network. This streamlined approach inherently accelerates the object detection process, effectively contributing to the expeditious and efficient nature of YOLO.

In reference to [3], the author underscores the distinctive prowess of the proposed method utilizing YOLO, deeming it the sole algorithm capable of accurately recognizing a minimum of six characters across all license plates (LPs). This assertion substantiates the method's exceptional performance and underscores its superiority in character recognition.

The author further advances their argument by subjecting the proposed ALPR system, integrated with YOLO, to a comparative evaluation against two commercial ALPR systems, serving as baselines, within the context of their dataset. The findings yielded compelling insights: both commercial ALPR systems yielded recognition rates below the 70% threshold. In stark contrast, the proposed ALPR system, seamlessly interwoven with YOLO, outshone its counterparts with a remarkable recognition rate of 78.333%. This outcome emphasizes the tangible benefits and efficacy of the proposed approach, firmly establishing it as a frontrunner in the realm of automated license plate recognition. Figure 2.2.1 shows all the result obtained in the custom data set [3].

ALPR	≥ 6 characters	All correct (vehicles)
Sighthound [7]	62.50%	47.39%
OpenALPR [28]	54.72%	50.94%
Proposed	87.33%	64.89%
Sighthound (with redundancy)	76.67%	56.67% (34/60)
OpenALPR (with redundancy)	73.33%	70.00% (42/60)
Proposed (with redundancy)	**88.33%**	**78.33% (47/60)**

Figure 2.2.1 A comparison of detection accuracy between the ALPR system proposed in this study and commercial ALPR system [3]

While in [27], the author presents a immediate Bhutanese license plate positioning using YOLO technology. Vehicle detection was executed before positioning to exterminate false positives. A single CNN achieved a general mean average exactness of 98.6% together with a development loss of 0.0231 for LP [27]. ALPR system playing a vital role in reducing congestion and increasing processing time. As road users increase, Bhutan faces the challenge of managing traffic efficiently. The country lacks single traffic lights, which are tedious and require extra workforce. To address this issue, a real-time system to identify single traffic lights (LP) is needed for developing an ALPR system for Bhutan. The conventional ALPR system consists of three phases: LP localization, character segmentation, and character recognition. Accurate localization can hinder character segmentation and recognition. This research aims to construct an application that identifies Bhutanese LP after vehicle detection using a single convolutional neural network. This will help improve traffic management in Bhutan and reduce the need for human resources.

The proposed system by the author in Figure 2.2.1.1 uses a video frame as input and a license plate (LP) output to localize a vehicle. Vehicle detection is performed to eliminate false positives, and license plates are only extracted if they are enclosed within a vehicle's bounding box. The YOLO single convolutional neural network is utilized to accomplish both localization of the automotive and the license plate simultaneously.

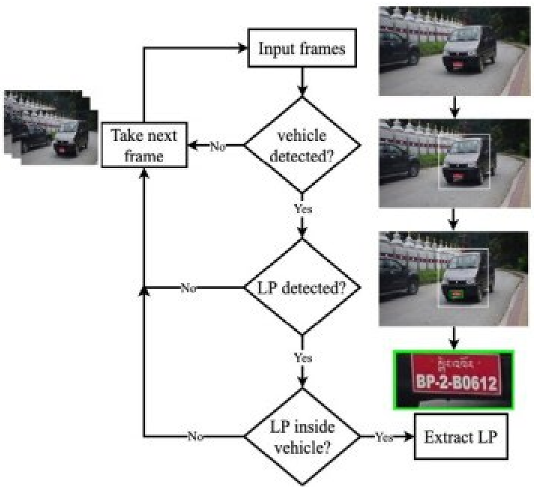


Figure 2.2.1.1 Author proposed system overview [27]

During the experimental results, the author used a dataset containing 1014 license plate images from video frames found in Bhutan. The YOLOv2 Darknet [28], based on Alexey's deployment, was used in author's system. The YOLO model was conditioned in Google Colaboratory for 7000th epochs alongside batch size of 64 plus image detection of 608 x 608.

Figure 2.2.1.3 illustrate the average Intersection Of Union and precision for plates plus vehicles generated for per epoch.

Tabulation of Average Precision (AP) and average IOU for each epoch.

Epoch	Plate (%)	Vehicle (%)	Avg IOU
1000	98.25	99.01	81.75
2000	98.51	98.95	83.48
3000	98.04	98.79	83.61
4000	98.52	98.83	83.92
5000	**98.52**	**98.82**	**83.99**
6000	98.52	98.82	83.94
7000	98.52	98.73	83.93

Figure 2.2.1.2 The Average Precision and average Intersection over Union (IOU) scores for each epoch are reported in [27]

After completing the 7000th epochs, the epoch with the peak average Intersection Of Union was designated to be applied into the author's offered real-time system. The 5000th epoch was chosen as it had the highest average IOU at 83.99%. The overall mAP came to be 98.6% alongside a training loss of 0.0231 for a batch size of 64 together 8 subdivisions [27]. Based on the documentation of Darknet [29], the model considered to be executing effectively if the average training loss is less than 0.05. Figures 2.2.1.3 and 2.2.1.4 show the results obtained from the author's proposed system.

Figure 2.2.1.3 False positive generated by signboard [27]

Figure 2.2.1.4 (a) LP detection (b) No LP was identified [27]

The study by the author aimed to address the issue of Bhutanese license plate localization using a sole convolutional neural network. This method achieved a 98.6% mean average precision (mAP) with a training loss of 0.0231, rendering it suitable towards real-time applications. Future research should consider utilizing larger datasets for training.

Automatic License Plate Recognition (ALPR) has taken place as popular research study since the early 1990s, leading in the creation of a variety of systems and commercial solutions adapted to a wide range of applications. However, many previous methods lack robustness in real-world settings due to limitations such as reliance on specialized cameras, narrow viewing angles, simple backgrounds, perfect lighting conditions, fixed region search, and certain vehicle kinds. Despite the availability of extensive annotated datasets and powerful hardware like GPUs capable of processing vast amounts of data, there persists a need toward achievable datasets encompassing vehicle and license plate labels.

The SSIG-SegPlate dataset, a famous community dataset of Brazilian license plates for ALPR, comprises under 800 training examples plus is subject to various restrictions. In response, this research proposes a more extensive benchmark dataset, UFPR-ALPR, focusing on tough yet common actual scenarios [30].

The UFPR-ALPR dataset was utilized by capture 4,500 images encompassing various vehicle types amidst intricate backgrounds plus varying lighting situations. These vehicles were positioned at different angles plus distances from the camera, and in certain instances, the vehicle was partially obscured within the image. Notably, there is a lack of publicly available datasets for ALPR containing labels for cars, motorcycles, license plates (LPs), and their associated symbols.

ALPR application should recognize diverse license plate layouts while functioning quickly sufficient to meet the expectations of Intelligent Transportation Systems (ITS). However, many academics underestimate the need for real-time processing in ALPR, instead proposing computationally intensive techniques that struggle to handle frames in real time. To solve this, the researchers developed specific YOLO-based ALPR models, using a variety of data augmentation approaches and network adjustments to accomplish the optimal stability of speed and accuracy at separate phase [30].
By presenting a unique, effective, and layout-independent ALPR system based on YOLO-based Convolutional Neural Networks (CNNs), the study significantly advances the ALPR field. Additionally, it presents a publicly accessible dataset for ALPR, which includes annotations detailing the positions of vehicles, license plates (LPs), and symbols. Moreover, this research conducts a comprehensive comparative evaluation of the suggested methodology against past methodologies plus two commercial applications across eight openly available datasets. Notably, this marks the first instance where an end-to-end ALPR system has been assessed on such a wide array of publicly accessible datasets, as highlighted in the current work [30].

The research assesses a newly suggested end-to-end applications across eight openly datasets, among which is UFPR-ALPR, employing images depicting diverse range of vehicles and license plate layouts. Most solutions work for specific layouts [31] and only evaluate in three datasets (e.g. [32], [33] and [34]). Even though motorcycles are a common mode of mobility in cities, ALPR system evaluations frequently ignore motorcycle photos.

The majority of methods lack execution time and are unable to recognize LPs in real-time, which makes it challenging to evaluate their suitability and trade-off between speed and accuracy. To tackle this issue, the author's proposed system assesses different YOLO models with diverse modifications, refining and amalgamating them to achieve optimal performance.

The research primarily concentrates on object detection and layout classification tailored for autonomous vehicle recognition (ALPR) systems. It adopts YOLO due to its remarkable speed/accuracy trade-off, enabling it to process more than double the frames per second (FPS) compared to other detectors while maintaining competitive performance. Addressing the bottleneck of license plate (LP) recognition, which currently impedes ALPR systems, the researchers suggest an integrated methodology encompassing LP detection plus with layout identification. This approach integrates heuristic rules and employs data enhancement techniques to enhance recognition outcomes. Furthermore, they devise and implement data enhancement methods mimic LPs of different layouts and also produce LP images featuring limited instances in the training set [30].

The dataset used for LP identification is diverse, as seen in Figure 2.2.1.5. It comprises 4,500 photos taken with three distinct camera types and a fixed PNG format size of 1,920 × 1,080 pixels. This study used three cameras, GoPro Hero4 Silver, Huawei P9 Lite, and iPhone 7 Plus, to capture images in PNG format. The images differ in quality due to camera specifications and camera position variations. The dataset was divided into training, testing, and validation, with 40% for training, 40% for testing, and 20% for validation. The photographs were annotated in text files with information such as the camera, vehicle position, type, maker, model, and year, as well as the LP's identification and position and character placement. The goal of the study was to increase the number of samples available for statistical analysis as well as for statistically significant analysis [30].

Figure 2.2.1.5 Sample images of the UFPR-ALPR dataset with various conditions [30]

Traffic images pose challenges for LP detection approaches due to textual blocks and small portions of LPs. In response to this challenge, the authors suggest a methodology that involves initially identifying vehicles within the input image and subsequently detecting their corresponding license plates (LPs) within vehicle patches. Then, instead of segmenting and classifying each character separately, they use a method where all characters are detected and recognized simultaneously by feeding the entire LP patch into the network.

The authors [30] offer an ALPR system that combines layout identification and LP detection tasks by training an object detection network. This enables the application of layout-specific LP recognition techniques, which kick in when the confidence value of the expected LP and its layout surpasses a predetermined level. This is noteworthy because it is the first time that a layout classification step is recommended to improve recognition results. The workflow of the authors' suggested ALPR system is shown in Figure 2.2.1.6.

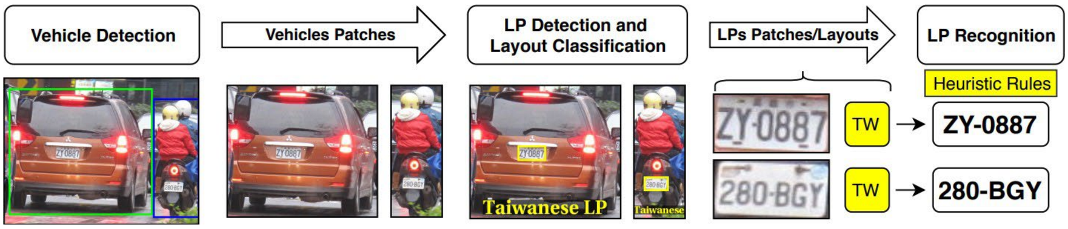


Figure 2.2.1.6 The proposed ALPR system's workflow by the authors [30]

The authors specialize object detection using YOLO-inspired models for ALPR, using specific models for each stage to improve performance. They adapt models like:

1. YOLOv2 for automative detection and recognition
2. Fast-YOLOv2 for LP detection and layout identification
3. CR-NET for LP recognition

They assess data augmentation techniques and adjust each network to attain the optimal balance between speed and accuracy at every stage. The authors explain how their models have been altered, with YOLOv2 digesting less distorted images and operating faster with a 25% smaller input size. Evaluations of speed and accuracy using several input sizes served as a guide for choosing the new dimensions. The k-means clustering algorithm was used to recalculate anchor boxes for the new input size and datasets. The number of filters in the final convolutional layer was decreased to produce two classes: cars and motorcycles, as the performance improved significantly with this configuration.

The Fast-YOLOv2 network's performance was enhanced by constructing a 3×3 convolutional layer with twice the filters of the preceding layer and altering the kernel size of the penultimate convolutional layer from 3×3 to 1×1. Better results were obtained from these changes without requiring more floating-point computations. To discover and classify additional number plate (LP) layouts, the network recalibrated anchor boxes and increased the number of filters in the final convolutional layer. To account for the average aspect ratio of Brazilian LPs discovered in the experimental datasets, the CR-NET measurements were modified from 240 × 80 to 352 × 128 pixels. The network is trained with the LP patch, character class, and coordinates to forecast 35 classes. Additionally, heuristic methods are developed to modify the CR-NET's output according to the anticipated class, guaranteeing accurate LP layout classification.

Using publicly available datasets, this study determines the minimum and maximum number of characters to be considered in number plates (LPs) of each design. Taiwanese, American, and European LPs lack a character count, although Brazilian and Chinese LPs do. Consideration is given to characters predicted with a confidence value higher than a predetermined threshold.

Redundant detections are eliminated via the non-maximum suppression (NMS) technique. When necessary, characters predicted with lower confidence values are either added or eliminated. LPs with low confidence value layouts usually have four to eight characters on them. On Brazilian and Chinese LPs, numerals and letters are swapped to reduce errors in often misclassified characters. The maximum and minimum number of symbols that must be included in LPs for each layout class are shown in Figure 2.2.1.7 [35].

Characters	American	Brazilian	Chinese	European	Taiwanese
Minimum	4	7	6	5	5
Maximum	7	7	6	8	6

Figure 2.2.1.7 The maximum and minimum character thresholds to be accounted for license plates within each layout category [30]

Research was done on eight publicly accessible datasets: AOLP [40], OpenALPR-EU [41], SSIG-SegPlate [42], ChineseLP [39], UCSD-Stills [38], EnglishLP [37], UFPR-ALPR, and Caltech Cars [36]. The researchers meticulously annotated the positions of vehicles, license plates (LPs), and characters in all images within these datasets. In the vehicle detection, LP detection, and layout classification stages, the system's F-measure rates exceeded 99%, demonstrating its robustness—a critical component in obtaining outstanding recognition results.

In order to compare the suggested ALPR system against cutting-edge methods and commercial systems like Sighthound and OpenALPR, which are frequently used as baselines in ALPR literature, the researchers followed the same evaluation procedure. The obtained results are readily available for more examination and contrast.

The results of the suggested system, earlier research, and commercial systems are displayed in Figure 2.2.1.8 for all datasets. The proposed method detected 96.9% of licence plates (LPs) correctly during five runs, outperforming Sighthound and OpenALPR by 9.1% and 6.2%, respectively. Moreover, the proposed strategy outperformed commercial solutions and earlier methods on four datasets.

Dataset \ Approach	[43]	[25]	[44]	[29]	[13]	Sighthound	OpenALPR	No Layout Classification†	Proposed
Caltech Cars	–	–	–	–	–	95.7 ± 2.7	**99.1 ± 1.2**	96.1 ± 1.8	98.7 ± 1.2
EnglishLP	**97.0**	–	–	–	–	92.5 ± 3.7	78.6 ± 3.6	95.5 ± 2.4	95.7 ± 2.3
UCSD-Stills	–	–	–	–	–	**98.3**	**98.3**	97.3 ± 1.9	98.0 ± 1.4
ChineseLP	–	–	–	–	–	90.4 ± 2.4	92.6 ± 1.9	95.4 ± 1.1	**97.5 ± 0.9**
AOLP	–	**99.8**‡	–	–	–	87.1 ± 0.8	–	98.4 ± 0.7	99.2 ± 0.4
OpenALPR-EU	–	–	93.5	85.2	–	93.5	91.7	96.7 ± 1.9	**97.8 ± 0.5**
SSIG-SegPlate	–	–	88.6	89.2	85.5	82.8	92.0	96.9 ± 0.5	**98.2 ± 0.5**
UFPR-ALPR	–	–	–	–	64.9	62.3	82.2	82.5 ± 1.1	**90.0 ± 0.7**
Average	–	–	–	–	–	87.8 ± 2.4	90.7 ± 2.3	94.8 ± 1.4	**96.9 ± 1.0**

Figure 2.2.1.8 Recognition rates (%) attained by the author's proposed system, prior research, and commercial systems across all datasets utilized in the experiments [30]

In the Caltech Cars dataset, the suggested method achieved results similar to OpenALPR, with a recognition rate of 98.7% as opposed to 99.1%. Notably, the system does not necessitate prior knowledge and relies on user input for the correct LP layout. Furthermore, in two of the five runs on the EnglishLP dataset, the system beat the optimal baseline. In the UCSD-Stills dataset, the recommended approach yielded an average recognition rate of 98%, whereas the two commercial systems produced an average rate of 98.3%. These analyses highlight how important it is to run the procedure five times and average the outcomes. Furthermore, the approach yielded similar results in the AOLP dataset, albeit with cropped LP patches and less precise detections. Figure 2.2.1.9 showcases the proposed approach's ability to accurately recognize LPs in various layouts, even under challenging conditions like shadows and high exposure, demonstrating its generalization capabilities.

Figure 2.2.1.10 illustrates the processing time required for each network in the author's proposed system to analyze an input image. For vehicle detection, the suggested system uses a deep CNN model, yet on a top-tier GPU, it processes data at 73 frames per second (FPS). Notably, it maintains a processing rate of over 30 FPS even in scenarios with four vehicles present, which represents a significant advancement compared to most approaches that are limited to real-time frame processing.

Figure 2.2.1.9 Samples of license plates accurately detected by the suggested ALPR system [30]

ALPR Stage	Adapted Model	Time (ms)	FPS
Vehicle Detection	YOLOv2	8.5382	117
LP Detection and Layout Classification	Fast-YOLOv2	3.0854	324
LP Recognition	CR-NET	1.9935	502
End-to-end	**-**	**13.6171**	**73**

Figure 2.2.1.10 Results for different networks analyzing through input images. [30]

This study provides an efficient and layout-independent ALPR system that uses a single technique for both license plate (LP) detection and layout classification, based on the YOLO detector. Utilizing post-processing rules at the recognition phase solves a significant flaw in current ALPR systems, which mostly rely on layout. Across all datasets except UFPR-ALPR, the system achieved recognition rates exceeding 95%, outperforming the best baseline by 7.8%. Notably, the system can process images in real-time, even when faced with scenes featuring four vehicles. Furthermore, 4,500 completely annotated photos from 150 vehicles in real-world circumstances are provided as part of a public dataset for ALPR. This dataset offers more than twice the number of images and encompasses a broader variety in various aspects. The researchers meticulously labeled the positions of vehicles, LPs, and characters in all datasets, which are made publicly available to the research community. The authors intend to recreate real-world circumstances without requiring the retraining of current systems by undertaking extensive experiments on cross-dataset scenarios using all available datasets, with the exception of one.

2.2.2 Character Segmentation

Throughout the literature review, the research has revealed various techniques for character segmentation. For instance, Figure 2.2.2.1 visually depicts the four distinct techniques of character segmentation.

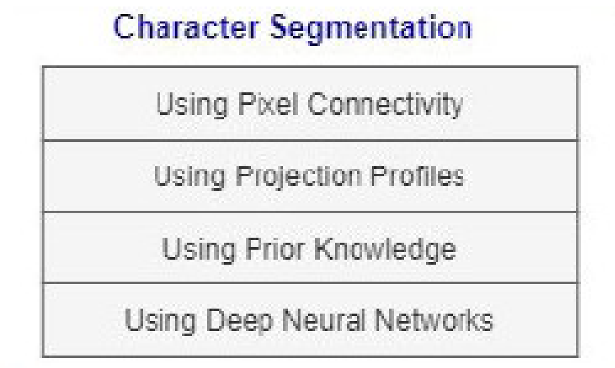


Figure 2.2.2.1 Techniques of character segmentation [4]

Additionally, in [4], the authors meticulously construct Table 2, which provides a comprehensive comparison of the diverse techniques employed in character segmentation. This tabulated representation facilitates an organized assessment of the distinctions and similarities between the different methods.

Techniques	Advantages	Limitations	References
Pixel connectivity	Simple, robust to rotation	Not suitable to be applied with joined or broken characters	[6], [7]
Projection profiles	Robust to rotation, independent of character positions	Sensitive to noise and font changes, number of characters in the license plate should be known	[8]
Using prior knowledges about the license plate	Simple	Specific to the region where they were designed to operate	[9], [10]
Deep neural networks	Reduce the number of parameters and the computation lost		[11], [12]

Table 2 Comparison of character segmentation methods [4]

2.2.3 Character Recognition

In an ALPR system, character recognition pertains to a technique or process that autonomously identifies and extracts the alphanumeric characters (comprising both letters and numbers) from a pre-processed license plate image. The fundamental objective of character recognition within an ALPR system revolves around the transformation of the visual elements present on the license plate into text that is machine-readable. This extracted text can then be subjected to subsequent processing and employed for diverse applications.

As highlighted in [4], the author classifies various classification techniques that necessitate fixed-size inputs for the learning model. Since the output from the segmentation stage varies in size, the input segments undergo re-scaling before classification. Given that the number of characters on a license plate, their relative positions, and possible values are generally known, each segment is classified to correspond to one of the possible values. This classification process unfolds across three scenarios:

1. A direct comparison is made between the pixel values of the raw image data and predefined templates.
2. Different image processing and machine learning techniques are employed to extract relevant features before segment classification.
3. Deep learning techniques are harnessed for segment classification.

2.2.3.1 Template and Patter Matching Techniques

In Malaysia, license plates adhere to a standardized font and character size, meticulously regulated by the JPJ. This standardized design presents a promising avenue for employing template matching techniques to facilitate character classification on license plates. Template matching, a widely recognized image processing approach, entails the identification of regions within an image that correspond to a designated template image. Notably, the author uses this technique in [13] by determining the correlation coefficient between the input and template photos; the closest match is indicated by the highest correlation coefficient value.

For further insight into the utilization of the template matching technique, Table 3 encapsulates a compilation of research papers that harness this approach. An overview of several factors is given in the table, including the techniques used for segmenting and detecting license plates, the speed at which recognitions occur, and performance measures including the success rates of character recognition (CR), segmentation (Seg.), and license plate detection (LPD), in addition to overall accuracy. This compilation serves as a comprehensive resource, shedding light on the varied applications and outcomes derived from the template matching technique within the context of license plate analysis.

Ref.	License Plate Detection (LPD) method	Segmentation (S) method	Time (sec)	LPD success rate (%)	Seg success rate (%)	CR success rate (%)	Overall success rate (%)
[14]	Horizontal and vertical projection segmentation	Bounding box	1.25	100	99.6	91.5	91.1
[15]	Gradient analysis	n/a	1.1s	n/a	n/a	n/a	91
[16]	Vertical edge	Vertical	n/a	96.22	94.04	95.24	n/a

	detection, seed-filling algorithm	projection & count number of pixels					
[17]	Edge detection & Smearing algorithm	Smearing, filtering, morphological	n/a	97.6	96	98.8	n/a
[18]	Morphological operation	Histogram and intensity projections	n/a	n/a	n/a	n/a	n/a
[19]	Canny method	Horizontal and vertical projection	n/a	n/a	n/a	n/a	91
[20]	Smearing method	Row and column segmentation	n/a	97.4	96.0	76.0	n/a
[21]	Connected component analysis	Bounding box	0.298	80	87.5	97	88.16

Table 3 showcases a collection of research papers that have employed template matching techniques. [13]

However, employing this technique can become challenging when dealing with numerous potential templates based on typography variations. To address this issue, the requirement to store additional templates arises, consequently leading to an escalation in both computation time and processing memory usage.

2.2.3.2 Character Recognition using feature extractor

Character recognition using feature extractors involves the extraction of distinct characteristics or features from characters to differentiate and classify them. These extracted features subsequently serve as input for classification algorithms, facilitating the determination of character identities. Feature extraction techniques play a pivotal role in pattern recognition tasks, such as optical character recognition (OCR). Several prevalent feature extraction techniques employed for character recognition are outlined below [4]:

1. **Gabor filter:** Class of linear filters used for feature extraction in image processing. They are especially effective for capturing texture information in images.
2. **Krish edge detection:** An edge detection algorithm that aims to identify significant intensity changes in an image.
3. **Eigenvector transformation:** A dimensionality reduction technique used to transform high-dimensional data into a lower-dimensional space while retaining the most important information.

These feature extraction techniques facilitate the transformation of raw pixel data into compact and meaningful representations, optimizing the process of character classification. The selection of a suitable technique hinges on factors such as the inherent characteristics of the character data, the complexity of the recognition task, and the performance requisites of the OCR system.

2.2.3.3 Character recognition using deep learning

Character recognition using deep learning entails the utilization of deep neural networks to autonomously identify and classify characters present within images. The advent of deep learning has brought about a paradigm shift in various domains, including computer vision, by empowering models to discern intricate features and patterns directly from data. Within the realm of character recognition, deep learning models exhibit the capacity to achieve exceptional levels of accuracy and robustness.

As highlighted in [4], recent studies emphasize the adoption of CNN, which have showcased significant promise across numerous computer vision tasks. Although deep learning-based approaches typically demand more computational resources compared to alternatives such as template matching and statistical feature extraction, they exhibit superior accuracy in general. This inherent trade-off between computational complexity and performance underscores the potency of deep learning techniques in enhancing character recognition outcomes.

2.3 YOLO version

Object detection algorithms are categorized into two groups based on the frequency of the same input image being passed through a network.

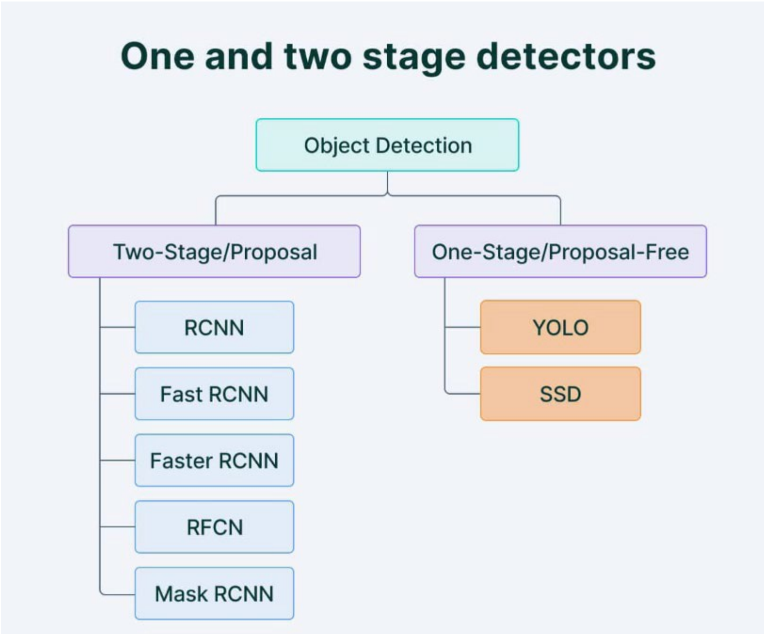


Figure 2.3.1 One and two stage detectors [44]

The YOLO models are well known for their quickness, precision, and capacity to identify objects in images with high accuracy and speed [43]. The backbone, neck, and head of an object detection model must be understood in order to gain further insight into the object detection problem. The architecture of contemporary object detectors is divided into the head, neck, and backbone in Figure 2.3.2 [43].

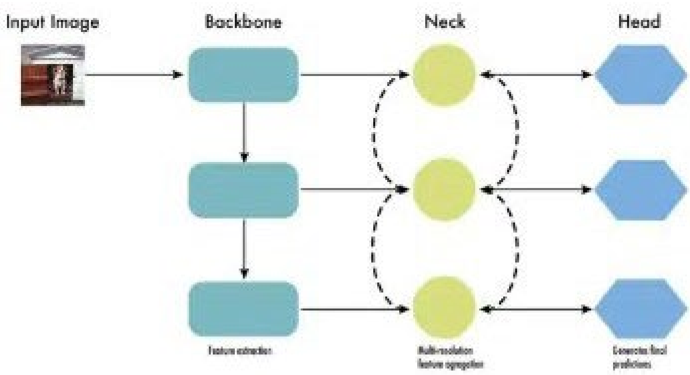


Figure 2.3.2 Architecture of modern object detectors [43].

The head, neck, and backbone make up the three components of the object detector architecture. The backbone uses a convolutional neural network to extract useful features at different scales from input images, capturing hierarchical features. Higher-level features are eliminated by the neck, which joins the head and backbone [43].

To measure the performance of object detection, mean Average Precision (mAP) is used to assess object detection, which offers a single number for comparison across all categories. The mAP is calculated using the COCO Evaluator with the following equation:

$$mAP = \frac{1}{|classes|} \sum_{ceclasses} \frac{|TP_c|}{|FP_c| + |TP_c|}$$

Equation 2.2.3.4 COCO Evaluator [34]

The mAP metric handles multiple object categories, defines positive predictions using Intersection over Union (IoU), and employs precision-recall metrics. Precision and recall metrics are used to assess the accuracy of the model; non-maximum suppression filters out redundant parameters, mAP provides a balanced evaluation, and IoU measures the quality of the bounding boxes.

Precision and recall are crucial aspects of a model's performance, with precision indicating accurate predictions and recall indicating the proportion of positive cases identified. The mean Average Precision (mAP) metric provides a balanced assessment by plotting precision against recall for different confidence thresholds. By comparing intersection and union areas, the Intersection over Union (IoU) metric assesses the accuracy of projected bounding boxes.

The overlap between the predicted and ground truth bounding boxes is calculated using the intersection area to union area ratio, or IoU. It is essential for assessing how accurate and useful a model is for the use case for which it was designed. Many times, object detection algorithms provide a number of bounding boxes with different confidence levels.

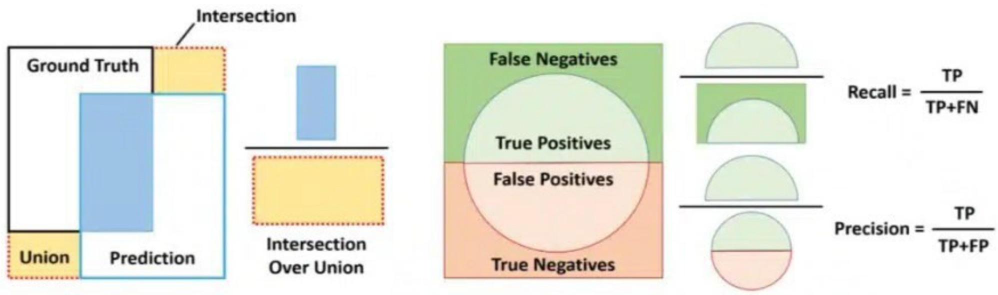


Figure 2.3.3 Evaluating the accuracy of a model [34]

In June 2016, Joseph Redmon presented YOLO, a groundbreaking end-to-end approach to real-time object detection at the CVPR Conference in Las Vegas. YOLO is an efficient object detection architecture that requires only one network pass, eliminating the need for multiple runs or a two-step process. The YOLO model achieved an impressive mAP of 63.4 on the PASCAL VOC2007 dataset [43].

By splitting an input image into a grid and projecting B bounding boxes with confidence scores for C classes per grid element, the YOLO object detection method finds bounding boxes. Pc, which represents the accuracy and confidence of the model, is included in each prediction. A tensor of SxSx(Bx 5 + C) is the result, and it can be suppressed to get rid of duplicate detections.

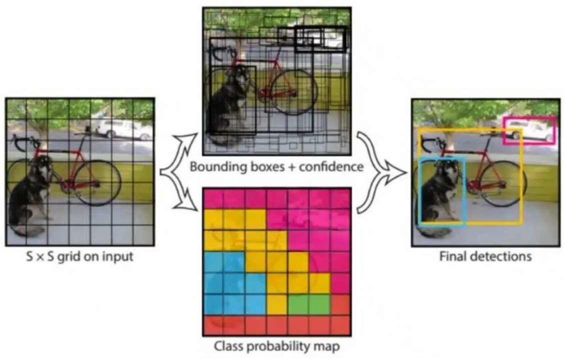


Figure 2.3.4 How the object is detected [43]

2.3.1 YOLO Architecture

The YOLO algorithm uses a deep convolutional neural network to detect objects in an image, with the CNN model as its backbone. YOLO architecture is similar to GoogleNet [47]. As seen in Figure 2.2.2.4 [47], it has a total of 24 convolutional layers, four max-pooling layers, and two fully connected layers.

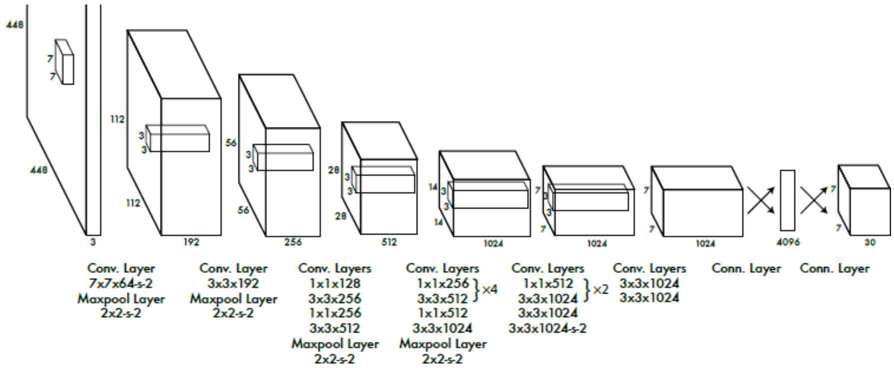


Figure 2.3.1.1 YOLO Architecture [48]

The YOLO (You Only Look Once) model is a deep learning approach that leverages pre-trained convolution layers from ImageNet. It transforms this model into a detection model by integrating convolutional and connected layers to enhance its performance. The final fully connected layer of the model predicts class probabilities and bounding box coordinates. YOLO divides an input image into an $SS \times SS$ grid, with each grid cell predicting BB bounding boxes and confidence scores. Each object in the image is assigned to one predictor based on the highest Intersection over Union (IOU) with the ground truth, resulting in specialization between predictors. This specialization enhances the overall recall score [44].

One key technique used in the YOLO models is non-maximum suppression (NMS), which improves object detection accuracy and efficiency by identifying and removing redundant or incorrect bounding boxes and outputs a single bounding box for each object in the image [44].

2.4 Analysis of YOLO version 5 to 8

2.4.1 YOLOv5 analysis

You Only Look Once (YOLO) is a well-known object detection system, with its fifth iteration, YOLOv5, introduced in 2020. Developed by Ultralytics, YOLOv5 enhances its speed, accuracy, and user-friendliness. It can achieve real-time object detection at up to 140 FPS on a single GPU using "SPP" feature. YOLOv5 also achieves state-of-the-art performance on popular benchmarks like COCO, PASCAL VOC, and OIDv4. It also introduces a new backbone architecture called "CSPNet" to enhance feature extraction. YOLOv5 comes in five versions: x, l, m, s, and n as shown in the Figure 2.4.1.1 [45].

Model	size (pixels)	mAP^{val} 50-95	mAP^{val} 50	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)	params (M)	FLOPs @640 (B)
YOLOv5n	640	28.0	45.7	**45**	**6.3**	**0.6**	**1.9**	**4.5**
YOLOv5s	640	37.4	56.8	98	6.4	0.9	7.2	16.5
YOLOv5m	640	45.4	64.1	224	8.2	1.7	21.2	49.0
YOLOv5l	640	49.0	67.3	430	10.1	2.7	46.5	109.1
YOLOv5x	640	50.7	68.9	766	12.1	4.8	86.7	205.7

Figure 2.4.1.1 Performance metrics of YOLO v5 [45]

YOLOv5 is the first YOLO version that implemented in Pytorch, rather than Darknet [47]. The primary enhancement of YOLOv5 is the incorporation of the Focus layer, which is a single layer generated by swapping out the YOLOv3 initial layer [47]. Without significantly affecting the mAP, this integration has decreased the number of layers and parameters while simultaneously increasing forward and reverse speed [47].

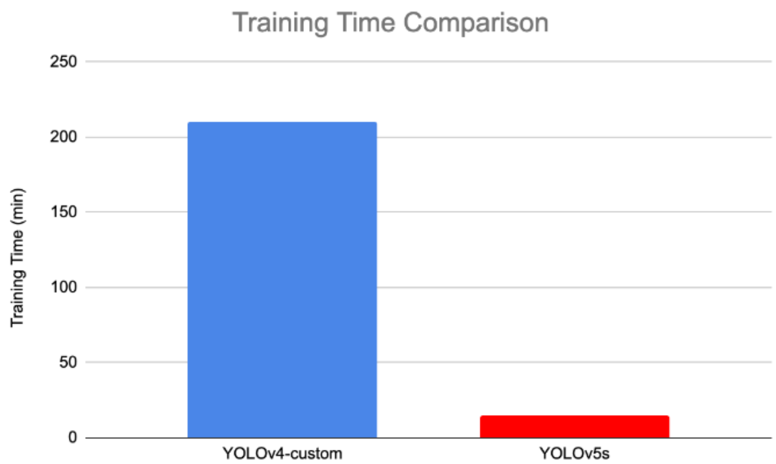


Figure 2.4.1.2 Training time comparison between YOLOv5 and YOLOv4 [49]

2.4.2 YOLOv7 analysis

YOLOv7 is a real-time object detection model that uses "bag-of-freebies" techniques like AutoAugment, CutMix, and DropBlock to enhance its robustness and generalization. Based on a single-shot detector architecture, it outperforms previous models on benchmarks like COCO and PASCAL VOC. It also performs efficiently on various hardware platforms, including desktop CPUs, mobile devices, and embedded systems. Figure 2.4.2.1 shows the performance of YOLO v7 in terms of speed, accuracy, efficiency, computation cost etc.

Model	Test Size	AP^{test}	AP_{50}^{test}	AP_{75}^{test}	batch 1 fps	batch 32 average time
YOLOv7	640	51.4%	69.7%	55.9%	161 *fps*	2.8 *ms*
YOLOv7-X	640	53.1%	71.2%	57.8%	114 *fps*	4.3 *ms*
YOLOv7-W6	1280	54.9%	72.6%	60.1%	84 *fps*	7.6 *ms*
YOLOv7-E6	1280	56.0%	73.5%	61.2%	56 *fps*	12.3 *ms*
YOLOv7-D6	1280	56.6%	74.0%	61.8%	44 *fps*	15.0 *ms*
YOLOv7-E6E	1280	56.8%	74.4%	62.1%	36 *fps*	18.7 *ms*

Figure 2.4.2.1 Performance benchmark for YOLO v7 [45]

A significant shift in architecture and at the Trainable bag-of-freebies level has been made by YOLOv7 [47]:

1. **Architecture level:** Extend Efficient Layer Aggregation Network (E-ELAN) integration has changed the YOLOv7 architectural level and enabled the model to learn more varied features for improved learning [47]. Additionally, YOLOv7 modifies its design by using components from YOLOv4, Scaled YOLOv4, and YOLO-R. This allows the model to accommodate different inference speed requirements [47].

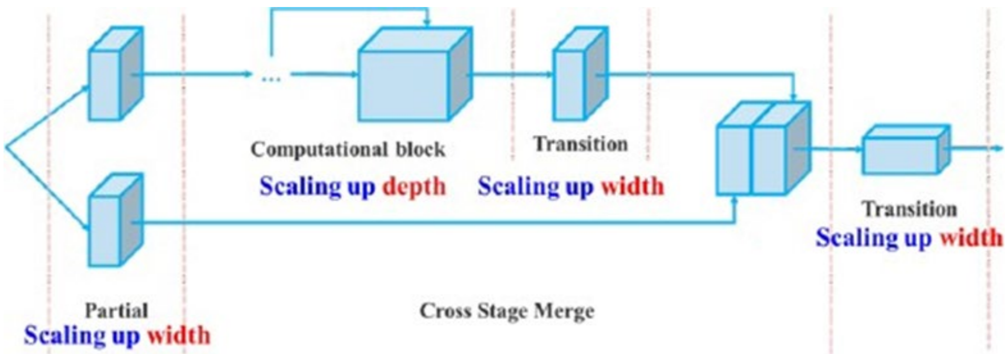


Figure 2.4.2.2 The process of compound scaling, which entails augmenting both the depth and width for a concatenation-based model [50]

2. **Trainable bag-of-freebies:** The 'bag-of-freebies' refers to enhancements made to the model's accuracy without raising the training costs. In summary, YOLOv7 has not only improved its inference speed but also its detection accuracy [47].

2.4.3 YOLOv8 analysis

YOLOv8, developed by Ultralytics, is an improved object detection model that builds on the success of previous versions of YOLO.

Features of YOLO v8:

- Better speed and accuracy than earlier iterations.
- A new backbone network for high-level feature capture that is based on EfficientNet.
- A brand-new feature fusion module that integrates features at various scales.
- Improved methods for augmenting data, such as MixUp and CutMix.

Ultralytics' YOLOv8 model is a state-of-the-art model that enhances performance and versatility, making it an excellent choice for object recognition, picture segmentation, and image classification tasks due to its quick, precise, and simple design. Figure 2.4.3.1 shows the performance of YOLO v8.

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Figure 2.4.3.1 Performance benchmark for YOLO v8 [45]

2.5 Strength of YOLO

There are several reasons why YOLO is popular and leading in object detections:

1. **Speed:** YOLO is incredibly quick since it doesn't deal with intricate pipelines. YOLO can also handle photos at 45 frames per second. The speed of YOLO in relation to other cutting-edge object detectors is shown in Figure 2.5.1 [47].

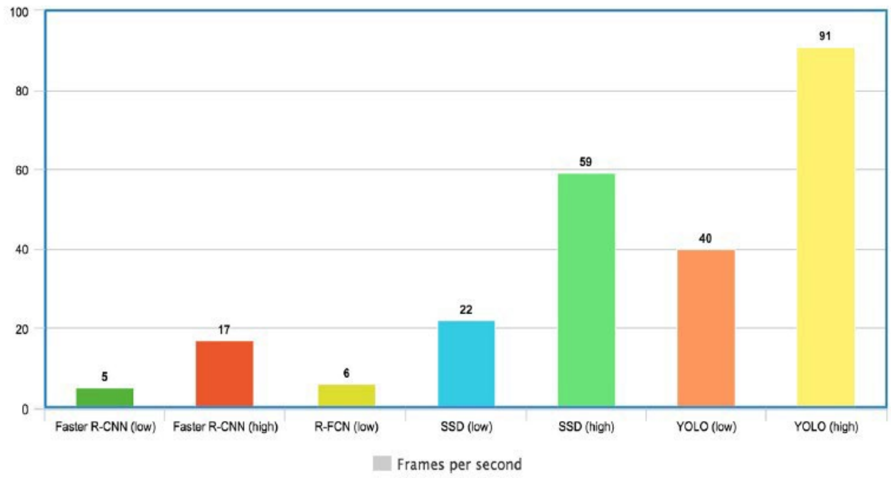


Figure 2.5.1 YOLO Speed compared to others object detectors [46]

2. **High detection accuracy:** With very few background mistakes, YOLO's accuracy greatly exceeds that of other cutting-edge models [47].

3. **Better generalization:** By offering a greater generalization for new domain, YOLO went a bit further, making it ideal for applications that depend on quick and reliable object detections [47].

2.6 Weakness of YOLO

YOLO, a fast object detector, has limitations due to its limited localization error, which is more significant than Fast R-CNN. This is due to its ability to detect two objects of the same class in a grid cell, its difficulty in predicting objects with non-existent aspect ratios, and its learning from coarse object features [43].

2.7 GAN (Generative Adversarial Network)

An estimating generative model via an adversarial process that run two models simultaneously:

1. G: Generative model captures the data distribution.
2. D: Discriminative model, estimates the probability that a sample came from the training data rather than G.

The training procedure of G is to maximize the probability of D making a mistake.

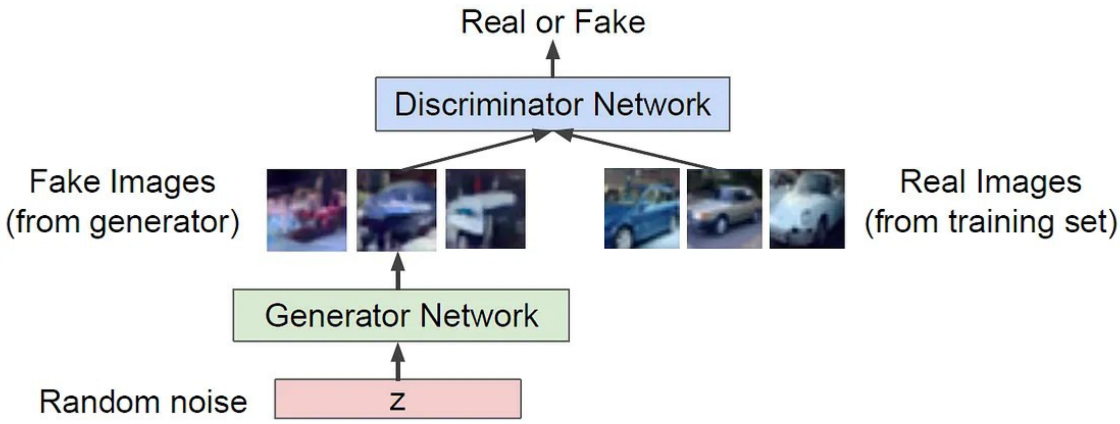


Figure 2.7.1 The functions and responsibilities of the generator and the discriminator [53]

In this proposed GAN, the generative model is pitted against an adversary: a discriminative model that learns to determine whether a sample is from the model distributions or the data distribution. The generative model can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency. Competition in this game drives both teams to improve their methods until the counterfeits are indistinguishable from the genuine articles. [54]

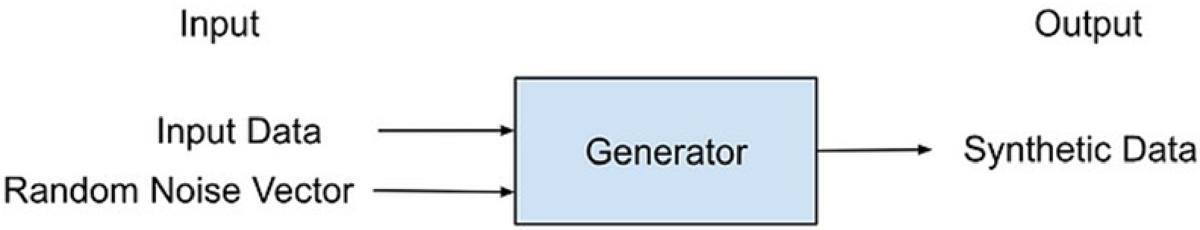


Figure 2.7.2 Generator [54]

The generator receives input data and a random noise vector and generates synthetic data that is passed into the discriminator.

Figure 2.7.3 Discriminator [54]

The discriminator is trained to discern between actual and generated data.

2.7.1 How does GAN work?

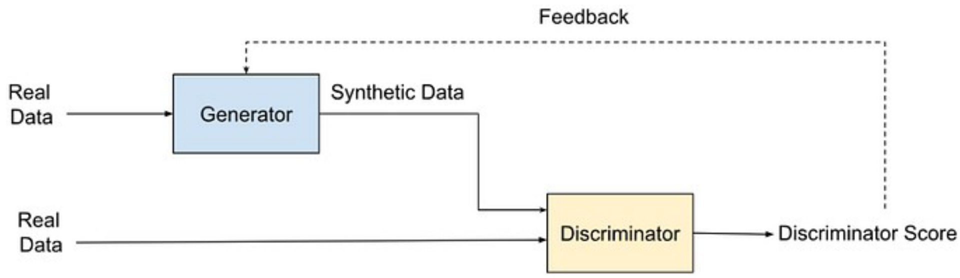


Figure 2.7.1.1 show how GAN work [55]

During training, generator and discriminator are trained alternately [55]. When the generator is working, the discriminator stops, and vice versa [55].

The generator model will try its best to generate synthetic data that is indistinguishable from the real data to fool the discriminator, while the discriminator will try its best to catch and identify the synthetic data created by the generator compared to the real data [55].

If the discriminator successfully detects synthetic data provided by the generator, it will provide feedback to the generator, allowing it to enhance the quality of data generated [55]. A point of equilibrium will be reached when the discriminator receives discriminating scores of 0.5 (50%) for the input to be either false or real [55].

2.8 Adversarial nets

D and G play the following two-player minimax game with value function V(G, D):

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{\text{data}}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log\left(1 - D_{\theta_d}\left(G_{\theta_g}(z)\right)\right)\right]$$

Figure 2.8.1 show equation of minimax objective function [53]

θ_g represents G's parameters, while θ_d represents D's parameters. D calculates the likelihood of a true image in the interval [0, 1]:

- D(x) equals 1 (or near to 1) when x is considered real data.
- D(x) equals 0 (or near to 0) if the data is considered fraudulent (e.g., generated).

This demonstrates that, at equilibrium, D outputs 1/2 everywhere because D has no notion how to distinguish between fraudulently created data and actual data.

2.9 Advantages and disadvantages

The disadvantages are primarily that there is no explicit representation of pg(x), and that D must be synchronized well with G during training (in particular, G must not be trained too much without updating D, in order to avoid "the Helvetica scenario" in which G collapses too many values of z to the same value of x to have enough diversity to model pdata), much as the negative chains of a Boltzmann machine must be kept up to date between learning steps [54].

The advantages are that Markov chains are never needed, only backprop is used to obtain gradients, no inference is needed during learning, and a wide variety of functions can be incorporated into the model [54].

The advantages are primarily computational [54]. Adversarial models may also gain some statistical advantage from the generator network not being updated directly with data examples, but only with gradients flowing through the discriminator [54]. This means that components of the input are not copied directly into the generator's parameters [54]. Another advantage of adversarial networks is that they can represent very sharp, even degenerate distributions, while methods based on Markov chains require that the distribution be somewhat blurry in order for the chains to be able to mix between modes [54].

2.10 Various types of GAN models

Depending on the mathematical formulas employed and the distinct manner in which the generator and discriminator interact, there are several types of GAN models.

- **Vanilla GAN:** With little to no feedback from the discriminator framework, the basic GAN model produces changes in the data. The majority of real-world use cases usually ask for improvements above a vanilla GAN.

- **Conditional GAN:** Targeted data generation is made possible by the conditional GAN (cGAN), which adds the notion of conditionally. Additional data is fed to the discriminator and generator, usually in the form of condition data or class labels. When creating photos, for example, the condition might be a label describing the content of the image. The generator can generate data that satisfies certain requirements by conditioning it.
- **Deep convolutional GAN:** The CNN architecture is integrated into GAN by Deep Convolutional GAN (DCGAN). In DCGAN, the discriminator also employs convolutional layers for data classification, while the generator utilises transposed convolutions to improve data distribution. In order to improve training stability, the DCGAN also introduces architecture guidelines.
- **Super-resolution GAN:** Upscaling low-quality images to high resolution is the main goal of super-resolution GANS (SRGANs). The objective is to preserve image quality and details while enhancing photographs to a greater resolution.
- **Enhanced super resolution** GAN: The concept of Generative Adversarial Networks (GANs), utilizing a generator network to increase the resolution of low-quality images while a discriminator network helps in ensuring that the generated images maintain realistic features.
- **Least Squares GAN:** A type of GAN that adopts the least squares loss function for the discriminator.
- **Wasserstein GAN:** WGAN is a type of GAN that minimizes an approximation of the Earth-Mover's distance (EM). This GAN model lead to more stable training than original GANs with less evidence of model collapse, as well as meaningful curves that can be used for debugging and searching hyperparameters.

2.11 Utilizing YOLOv4 for sophisticated traffic sign recognition, augmented with synthetic training data produced by diverse Generative Adversarial Networks (GANs)

According to the author of this paper, given enough annotated training data, a Convolutional Neural Network (CNN) can identify traffic signs perfectly [56]. Nevertheless, there aren't many databases for traffic signs from most countries in the world [56]. In this instance, the author suggested using GAN to generate a variety of more realistic training images that complement the actual image layout [56]. The author's goal is to explain how DCGAN, LSGAN, and WGAN produce synthetic images and how that quality is assessed [56]. YOLOv3 and YOLOv4 will be utilised in the testing phase to identify the photos [56].

In Figure 2.11.1, it shown author's proposed system's synthetic data generation methodologies for enhanced traffic sign identification utilizing DCGAN, LSGAN and WGAN [56].

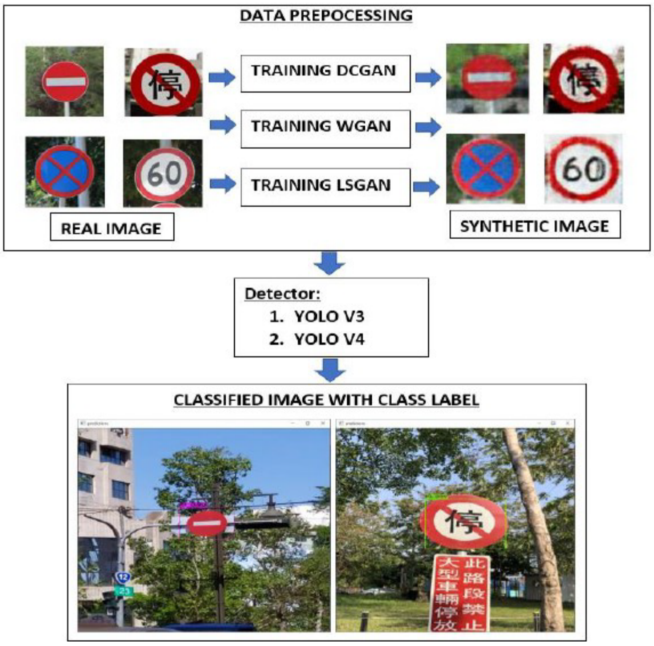


Figure 2.11.1 Overview of the system [56]

In the data preparation step, the author uses multiple GANs to generate synthetic prohibitory sign images [56]. Furthermore, the dataset was separated into four groups, each containing a different form of synthetic image generated by a GAN [56]. The whole dataset combination is presented in the figure below.

Group	Original Image	Synthetic Image		
		DCGAN	LSGAN	WGAN
1	✓			
2	✓	✓		
3	✓		✓	
4	✓			✓

Figure 2.11.2 Dataset combination [56]

In this research, images are categorised based on the total number of photos utilised for training [56]. 200 images with 64x64 and 32x32 pixel dimensions make up the first batch

[56]. Next, 1000 photos will be generated for every combination of the same size, and so on for other sets [56]. The picture below depicts the GAN experimental settings:

No	Total Image	Image size (px)/ Generate Image (px)	Total Generate Image
1	200	64x64	1000
2	200	32x32	1000
3	100	64x64	1000
4	100	32x32	1000
5	50	64x64	1000
6	50	32x32	1000

Figure 2.11.4 GANs experiment setting [56]

Every GAN uses the same training set. The performance of the traffic sign identification system will be improved by integrating real images with synthetic photos generated by various GANs methodologies for training [56].

Figure 2.11.5 Synthetic images generated by DCGAN with size (a)32x32 and (b)64x64 [56]

No	Total Image	Image size (px)	DCGAN MSE	DCGAN SSIM	LSGAN MSE	LSGAN SSIM	WGAN MSE	WGAN SSIM
				P1				
1	200	64x64	9.342	0.483	8.156	0.497	7.485	0.509
2	200	32x32	3.502	0.558	4.019	0.529	4.009	0.533
3	100	64x64	8.285	0.502	8.569	0.475	7.653	0.504
4	100	32x32	5.126	0.449	4.102	0.557	4.081	0.531
5	50	64x64	9.776	0.366	9.93	0.336	9.39	0.41
6	50	32x32	3.924	0.562	8.619	0.26	4.877	0.453
				P2				
1	200	64x64	8.969	0.385	8.96	0.436	8.385	0.475
2	200	32x32	4.724	0.383	4.213	0.423	4.639	0.432
3	100	64x64	8.907	0.391	8.943	0.362	8.094	0.449
4	100	32x32	7.999	0.123	4.408	0.402	4.425	0.393
5	50	64x64	9.549	0.394	9.313	0.272	9.013	0.377
6	50	32x32	4.29	0.375	4.999	0.243	4.601	0.336
				P3				
1	200	64x64	9.966	0.452	9.222	0.461	8.977	0.469
2	200	32x32	5.239	0.478	4.644	0.504	4.865	0.459
3	100	64x64	9.895	0.392	9.941	0.38	8.321	0.478
4	100	32x32	4.651	0.494	4.571	0.469	4.579	0.47
5	50	64x64	10.503	0.339	12.537	0.233	9.936	0.377
6	50	32x32	5.484	0.434	5.874	0.363	5.503	0.391
				P4				
1	200	64x64	10.055	0.463	11.888	0.362	9.649	0.48
2	200	32x32	5.698	0.453	4.934	0.535	4.89	0.504
3	100	64x64	11.181	0.431	11.255	0.39	10.834	0.459
4	100	32x32	7.358	0.459	5.762	0.469	5.527	0.47
5	50	64x64	16.311	0.326	13.637	0.313	13.035	0.405
6	50	32x32	6.428	0.456	6.399	0.362	6.102	0.445
				Average				
1	200	64x64	9.583	0.446	9.557	0.439	8.624	0.483
2	200	32x32	4.791	0.468	4.453	0.498	4.601	0.482
3	100	64x64	9.567	0.429	9.677	0.402	8.726	0.473
4	100	32x32	6.284	0.381	4.711	0.474	4.653	0.466
5	50	64x64	11.535	0.356	11.354	0.289	10.344	0.392
6	50	32x32	5.032	0.457	6.473	0.307	5.271	0.406

Figure 2.11.6 Performance evaluation of various GAN with 2000 epochs [56]

According to the picture above, LSGAN achieves the best performance in synthetic image creation with a total of 200 photos as input, dimensions 32x32, and 2000 epoch [56]. These groups show the highest SSIM (Structural Similarity Index) values of 0.498 and the lowest MSE (Mean Square Error) values of 4.453 [56].

The experiment with the Taiwan prohibitory sign was done out using generated photographs, synthesized images, and a combination of the two. The dataset consists of 70% training and 30% testing. 200 generated images from various GANs will be mixed with real photos [56]. Figure [56] provides a detailed explanation of the Taiwan prohibitory signage.

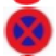
Class ID	Class Name	Sign	Original Image	Synthetic Image DCGAN, LSGAN, WGAN
P1	No entry		235	200
P2	No stopping		250	200
P3	No parking		230	200
P4	Speed Limit		185	200

Figure 2.11.7 Taiwan Prohibitory Sign [56]

Model	Dataset	Loss Value	Name	AP (%)	TP	FN	FP	Recall	Precision	IoU (%)	F1-score	mAP (%)
Yolo V3	Group 1 (Original Image)	0.0162	P1	97.5	299	2	3	0.99	0.99	90.93	0.99	99.08
			P2	100								
			P3	99.8								
			P4	99.04								
Yolo V4	Group 1 (Original Image)	0.1441	P1	98.75	299	2	3	0.99	0.99	92.4	0.99	99.55
			P2	100								
			P3	100								
			P4	99.46								
Yolo V3	Group 2 (Original Image, DCGAN)	0.0269	P1	94.9	558	10	12	0.98	0.98	86.45	0.98	98.44
			P2	100								
			P3	99.97								
			P4	98.88								
Yolo V4	Group 2 (Original Image, DCGAN)	0.3601	P1	97.25	563	5	6	0.99	0.99	88.33	0.99	99.07
			P2	100								
			P3	100								
			P4	99.05								
Yolo V3	Group 3 (Original Image, LSGAN)	0.0217	P1	99.99	564	4	7	1	0.99	89.48	1	99.83
			P2	100								
			P3	98.23								
			P4	99.99								
Yolo V4	**Group 3 (Original Image, LSGAN)**	0.2173	P1	100	566	2	5	1	0.99	90.35	0.99	**99.98**
			P2	99.33								
			P3	100								
			P4	100								
Yolo V3	Group 4 (Original Image, WGAN)	0.025	P1	98.63	565	2	5	1	0.99	89.51	0.99	99.51
			P2	100								
			P3	99.98								
			P4	99.43								
Yolo V4	Group 4 (Original Image, WGAN)	0.2239	P1	98.63	565	2	4	1	0.99	90.4	0.99	99.45
			P2	100								
			P3	99.99								
			P4	99.2								

Figure 2.11.8 Training result [56]

Dataset	Accuracy (%)		Not detect	
	Yolo V3	Yolo V4	Yolo V3	Yolo V4
Group 1 (Original Image)	84.31	69.7	7	8
Group 2 (Original Image, DCGAN)	74.34	82.5	7	7
Group 3 (Original Image, LSGAN)	**84.9**	**89.33**	**7**	**2**

Figure 2.11.9 Testing accuracy result performance [56]

As can be seen from the following table, YOLOv4 is typically more accurate than previous iterations [56]. Group 3 (Original Image, LSGAN) had the best accuracy, with YOLOv4 model accuracy at 89.33% and YOLOv3 accuracy at 84.9% [56]. When utilising YOLOv4, there are just two detection mistakes in Group 3 as opposed to seven on YOLOv3 [56].

2.12 Enhancing License Plate Image Analysis Through the Use of Generative Adversarial Networks (GANs)

With the goal of learning probability distributions from input data, deep neural networks for generative models are trained using the Generative Adversarial Network (GAN) approach [57]. Originally designed to produce more realistic-looking false images, GANs have been shown in previous studies to be able to produce intricate training algorithms [57]. Applications range from discriminative tasks like human pose estimation to generative tasks like super-resolution, style transfer, and natural language processing [57].

Car plate Super-Resolution (SR) aims to improve and precisely identify plate images so that computer and human identification can be done more precisely [57]. The efficacy of the Automatic License Plate Recognition (ALPR) system hinges on the accuracy of the car plate identification algorithm and the quality of acquisition [57]. Surveillance systems commonly use LP pictures or videos [57]. However, text on vehicle plates may become illegible due to factors such as distance, lighting conditions, and perspective distortion in low-resolution surveillance setups [57]. Hence, a balance between heightened accuracy and minimal processing time is crucial for successful implementation [57].

The author reports that only minor modifications were made to the structure of SRGAN and that a figure was provided [57] that showed the overall arrangement of the proposed ALPR model based on Super-Resolution GAN. Following training of the network, the super-resolution images are fed into the OCR framework (Optical Character Recognition), which utilizes YOLOv5 in the last step to identify characters on license plates [57].

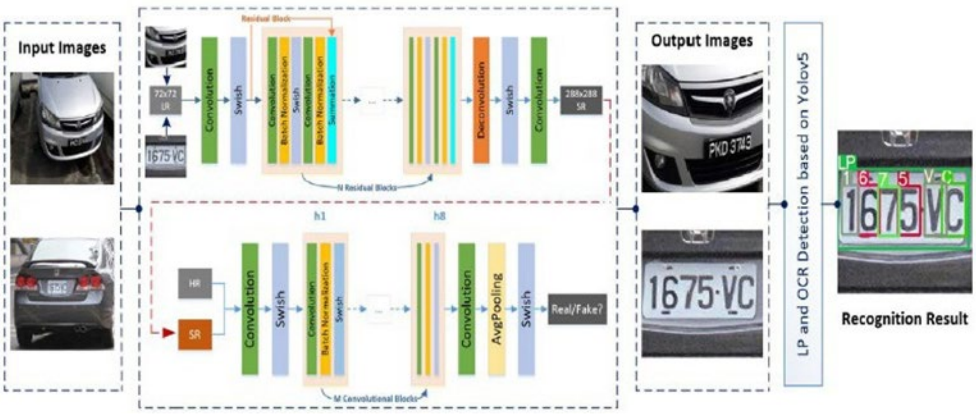


Figure 2.12.1 Overview of the proposed system integrating super-resolution techniques for license plates and a recognition system based on YOLO [57]

Two datasets were used in the studies. The vehicle plate dataset was first divided into two parts: testing images (102) and training images (1060). Metrics including the Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR) were used to evaluate the output image quality using the testing subset [57]. Later, the recognition model (YOLOv5) was evaluated [57], and the suggested model was tested on a different dataset, which is the "Application-Oriented Licence Plate (AOLP)" [58].Based on the results of the current investigation, the author plans to compare the suggested model with alternative methods, namely Deep Image Prior (DIP) and SRCNN [57].

The Deep Image Prior technique promotes the use of Deep Neural Networks (DNNs) that have not yet been trained as prior models for pictures. The "deep image prior" model is credited with originating this idea [57].

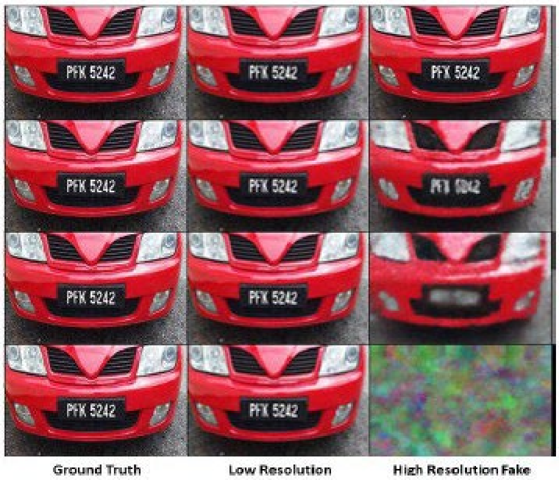


Figure 2.12.2 Example of Deep Image Prior [57]

Deep learning methods for Single Image Super-Resolution (SISR) are introduced by SRCNN [59]. These methods train an end-to-end mapping from Low-Resolution (LR) to High-Resolution (HR) pictures [57]. This mapping is realized through a Deep Convolutional Neural Network (DCNN) that takes an LR image as input and outputs a corresponding HR image [57]. The Super-Resolution Convolutional Neural Network (SRCNN), as shown in Figure 2.12.3, is a basic network design that the authors suggest. It consists of three convolutional layers for feature extraction, nonlinear mapping, and image reconstruction. When compared to earlier machine-learning-based SR techniques, SRCNN performs better [57].

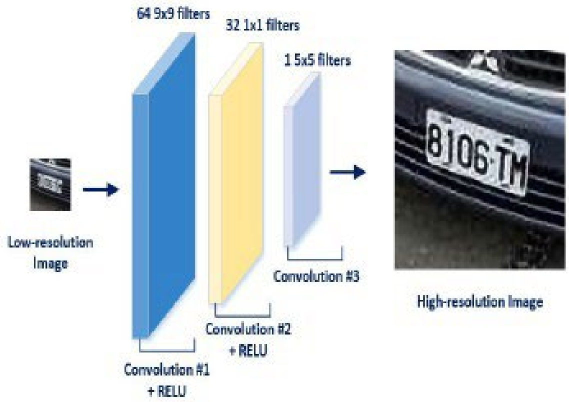


Figure 2.12.3 SRCNN architecture [57]

A comparison is conducted among the following Super-Resolution (SR) methods on the proposed dataset to illustrate the efficacy of the proposed approaches [57]. A visualization of the performance comparison between the SRGAN architecture and alternative techniques is presented in Figure 2.12.4 [57].

Figure 2.12.4 Example of comparative analysis with current methodologies incorporating visual elements [57]

As shown in figure 2.12.5, the results of SRGAN are better than the other methods.

Figure 2.12.5 PSNR comparison of the proposed techniques [57]

Additionally, Figure 1 offers a succinct summary of the quantitative performance comparisons with earlier methods [57]. The test dataset has 102 photos that have the same distribution as the training datasets, and these results are assessed across these images [57]. Furthermore, the author claims that SRGAN performs better in terms of Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM), as shown in the figure below [57].

Model	PSNR	SSIM	Execution Time
Deep Prior	23.756	0.771	157.58 sec
SRCNN	23.494	0.767	0.11 sec
SRGAN	26.449	0.833	0.036 sec
SRGAN-TV	26.621	0.837	0.03 sec

Figure 2.12.6 Quantitative comparison with existing methods [57]

Model	PSNR	SSIM
SRGAN (MSE)	24.8686	0.7213
SRGAN (MSE and perceptual loss)	23.0307	0.6760
Proposed SRGAN (Model I)	26.449	0.833
Proposed SRGAN-TV (Model II)	26.621	0.837

Figure 2.12.7 Quantitative comparison with method [57]

The main goal of the author's suggested approach, in addition to super-resolution, is to improve License Plate (LP) identification performance by testing the recommended model on a different dataset (ALOP) [57]. T The procedure entails YOLO identifying the little license plate from the pictures, then the SR model enhancing the detected license plate's resolution. Lastly, the letters on the license plate are recognized using the Optical Character Recognition (OCR) technique [57].

The figure below showcases the high-resolution images generated by the SRGAN model for the AC, LE, and RP subsets [57]. The proposed method achieves superior accuracy compared to previous methods, highlighting the capability of super-resolution to enhance recognition [57].

Model	LR Images	DIP	SRCNN	SRGAN	SRGAN-TV
Accuracy	18.8%	43%	93.5%	95.4%	93.2%

Figure 2.12.8 Comparative assessment of the SRGAN technique against other cutting- edge methods using the ALOP dataset [57]

Figure 2.12.9 PNSR comparison of the proposed techniques [57]

Figure 2.12.10 The results generated by the Optical Character Recognition (OCR) model for low-resolution (LR) images [57]

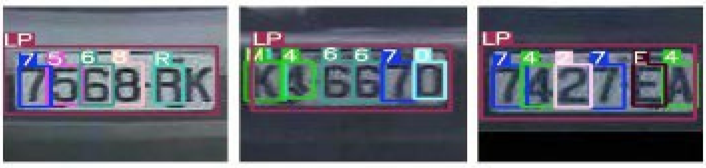


Figure 2.12.11 The predictions produced by the SRCNN model [57]

Figure 2.12.12 The results obtained from the Optical Character Recognition (OCR) model for the SRGAN model [57]

Figure 2.12.13 The results yielded by the Optical Character Recognition (OCR) model for the SRGAN-TV model [57]

2.13 ESRGAN refers to Enhanced Super-Resolution Generative Adversarial Networks

A revolutionary development that produces realistic textures in single picture super-resolution is the Super-Resolution Generative Adversarial Networks (SRGAN) [51]. To elevate the visual quality further, the authors have meticulously studied three crucial components of the SRGAN network architecture: adversarial loss, perceptual loss, and enhanced each component to derive an improved version of SRGAN [51].

A fundamental low-level vision challenge, single image super-resolution (SISR) has attracted increasing attention from AI firms as well as the research community [51]. The primary objective of SISR is to enhance a low-resolution image to yield a high-resolution counterpart [51].

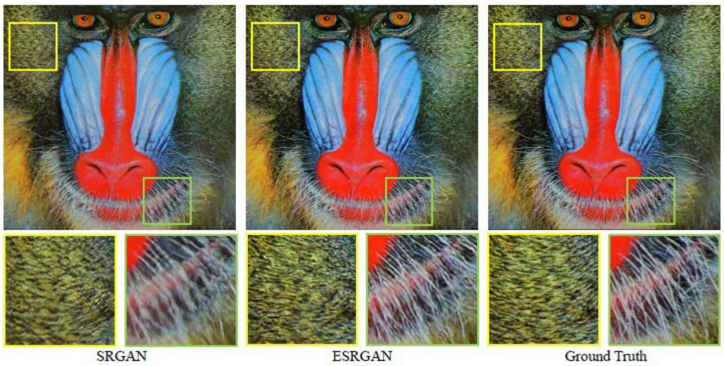


Figure 2.13.1 The super-resolution results of SRGAN, the proposed ESRGAN by the authors, and the ground-truth ESRGAN are depicted [51]

The preceding figure shows that, as the authors have mentioned, there is still a discernible difference between the SRGAN results and the ground-truth (GT) photos. In response, the authors revisit the key components of SRGAN and enhance the model in three aspects [51]. Firstly, they enhance the network structure by introducing the Residual-in-Residual Dense Block (RDDB), which boasts higher capacity and is easier to train [51]. Next, authors removed Batch Normalization (BN) layers by using residual scaling and smaller initialization to facilitate training a very deep network [51].

Besides that, authors improve the discriminator using “whether one image is more realistic than the other” rather than “whether one images is real or fake” [51]. Furthermore, in this paper, authors propose an improved perceptual loss by using the VGG features before activation instead of activation as in SRGAN.

The aim of authors is to improve the overall perceptual quality of SR. In figure below show a basic architecture of SRResNet [51].

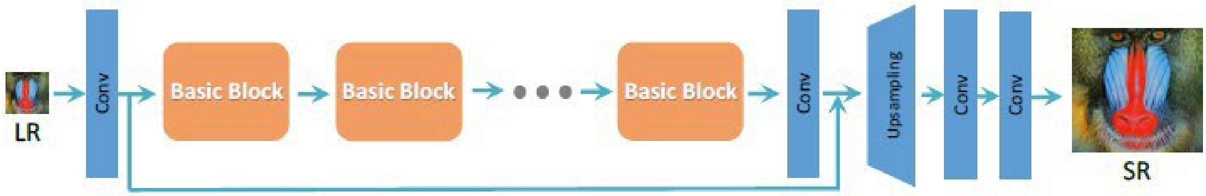


Figure 2.13.2 show the basic architecture of SRResNet [51]

In order to improve the recovered image quality of SRGAN, authors mainly make two modifications to the structure of generator by remove all BN layers and replace the original basic block with the proposed Residual-in-Residual Dense Block (RRDB), which combines multi-level residual network and dense connections as depicted in Figure below.

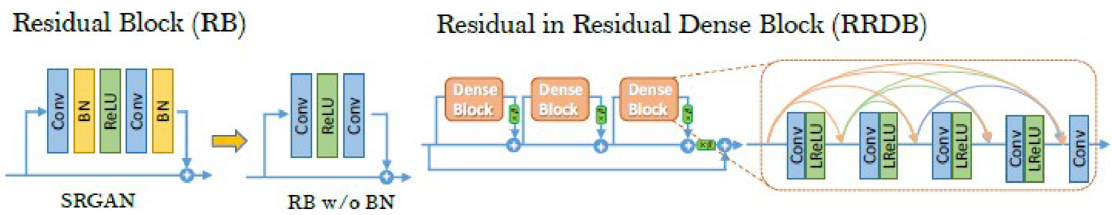


Figure 2.13.3 show the proposed Residual-in-Residual Dense Block (RRDN) [51]

It has been shown that removing the Batch Normalisation (BN) layers improves performance and decreases computing cost for a number of Peak Signal-to-Noise Ratio (PSNR)-focused tasks, such as deblurring and Super-Resolution (SR) [51]. BN layers use the mean and variance of a batch to normalise features during training, and during testing, they use the estimated mean and variance of the complete training dataset [51]. The authors preserve the overall architectural layout of SRGAN while presenting a brand-new fundamental block known as the Residual-in-Residual Dense Block (RRDB) [51]. By comparison with the original residual block in SRGAN, the proposed RRDB incorporates a more intricate and profound structure [51]. In addition to the improved architecture, authors also exploit several techniques to facilitate training a very deep network:

1. **Residual scaling:** Scaling down the residual by multiplying a constant between 0 and 1 before adding them to the main path to prevent instability [51].
2. **Smaller initialization:** Authors empirically find the residual architecture is easier to train when the initial parameter variance become smaller [51].

Beside the improved structure of generator, authors also enhance the discriminator based on the realistic GAN [52]. Different from the standard discriminator D in SRGAN, which estimates the probability that one input image xx is real and natural, a relativistic discriminator tries to predict the probability that a real image xx_{rr} is relatively more realistic than a fake on x_f as shown in figure below.

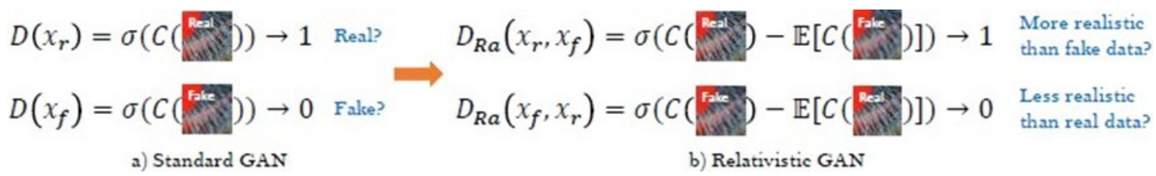


Figure 2.13.4 Difference between standard discriminator and relativistic discriminator [51]

Moreover, the authors constrained features prior to activation, as opposed to after activation, as is frequently done in SRGAN, to provide a more effective perceptual loss L_{percep} [51]. The authors then evaluated their final models on several public benchmark datasets, comparing them with state-of-the-art PSNR-oriented methods such as SRCNN, EDSR, and RCAN, as well as perceptual-driven approaches including SRGAN and EnhanceNet.

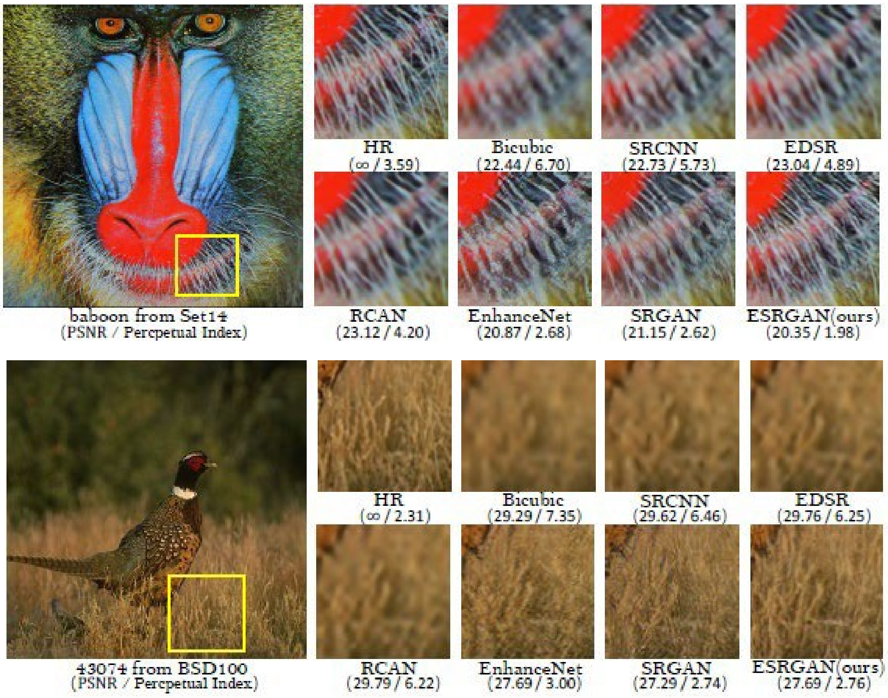


Figure 2.13.5 The qualitative results of ESRGAN demonstrate its ability to generate images with more natural textures compared to other methods [51]

It is clear that the authors' suggested ESRGAN outperforms earlier methods in terms of both granularity and detail [51]. For example, ESRGAN produces grass textures and baboon whiskers that are crisper and more realistic than PSNR-oriented techniques, which result in textures that are artificial and noisy [51].

The authors systematically alter the baseline SRGAN model and compare their inconsistencies in order to thoroughly analyse the effects of each component in the proposed ESRGAN [51]. Figure [51] shows the overall visual comparison. Each column shows a model with its configuration information at the top [51].

To guarantee steady and reliable performance free from artefacts, the authors first remove all BN layers throughout the BN removal method [51]. This modification successfully conserves computational resources and lowers memory consumption without compromising performance [51]. Additionally, the authors note that in deeper and more complex networks, models with BN layers are more prone to introducing undesirable artifacts [51]. Examples illustrating this phenomenon are available in the supplementary material [51].

The authors show that using characteristics before to activation produces more accurate brightness in reconstructed images, which mitigates the perceived loss [51]. Furthermore, the RaGAN employs an improved relativistic discriminator, which enhances the learning of sharper edges and more detailed textures [51].

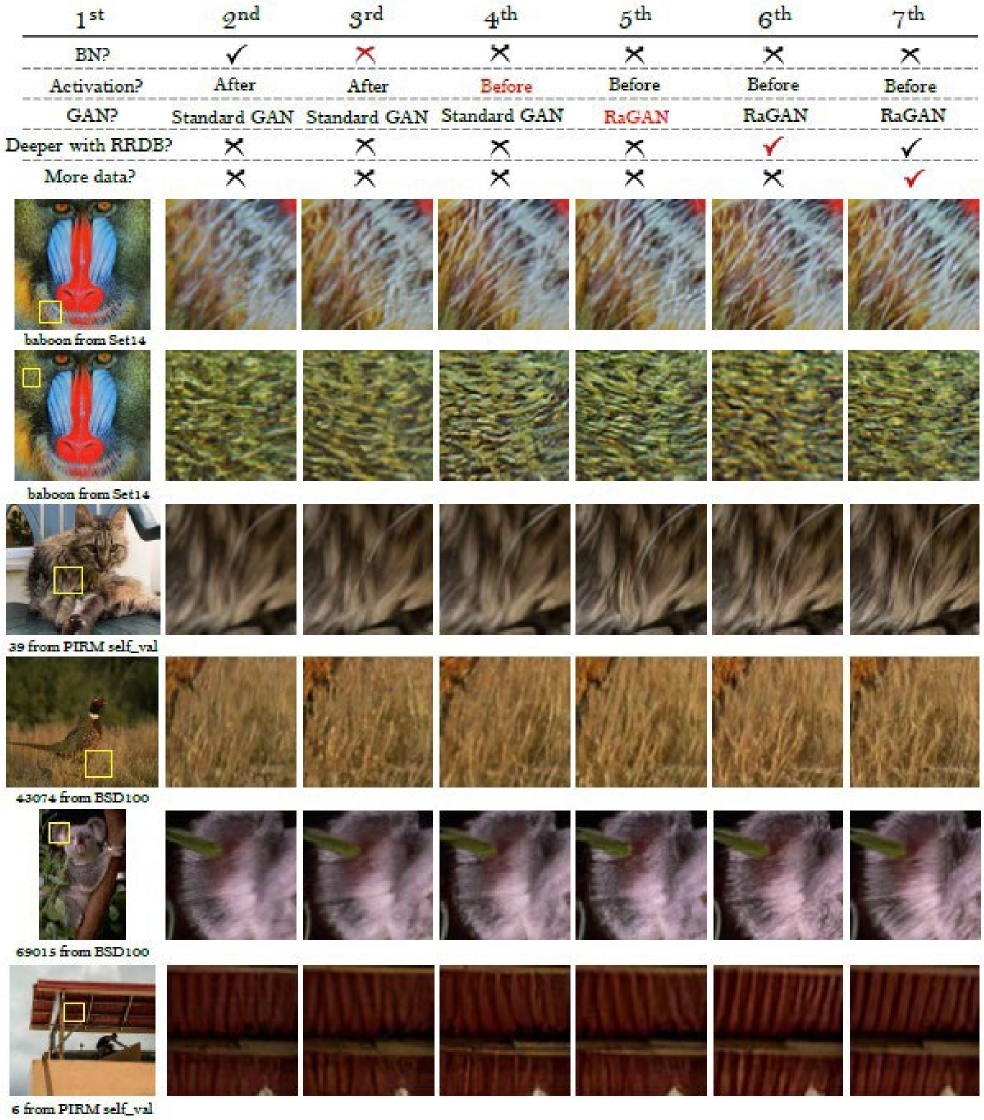


Figure 2.13.6 An overall visual comparison effectively demonstrates the effects of each component in ESRGAN [51]

2.14 Chapter Summary

This chapter further discusses the existing Automatic License Plate Recognition (ALPR) systems and applications developed by others. Additionally, it outlines the strengths and weaknesses of the reviewed systems. Different techniques used for ALPR systems are listed to improve the proposed system and enhance its performance.

Chapter 3

System Methodology and Proposed System

3.1 Methods / Technology involve

The project's path was divided into different parts to make things easier. First, planned the project and chose how would do it. Then, gathered a lot of license plate images to work with. After that, got these images ready for the computer to understand by making them cleaner and clearer. This project used these prepared images to teach the computer models what license plates look like. Finally, made the proposed system even better by fixing any issues and making it work as best as it could. The progress can conclude through development process as below:

1. Project Pre-development
2. Data Collection
3. Data Pre-processing
4. Model Training
5. Enhancement of developed system

This way of working step by step helped us create a strong and effective Automated License Plate System Recognition (ALPR).

3.2 System Requirement

3.2.1 Hardware

The hardware employed in this project prominently features a laptop, assuming a pivotal role in facilitating a multitude of steps and processes across the entire development trajectory. The laptop constitutes the cornerstone of executing pivotal activities, encompassing the implementation of the ALPR system, the training of machine learning models, and the exhaustive testing regimen aimed at gauging the efficacy of the developed system. It functions as the nucleus, unifying and spearheading all undertakings associated with the inception and refinement of the ALPR system. The particulars of the laptop employed during the development are meticulously laid out in Table 3.2.1, encapsulating its specifications and capabilities.

Table 3.2.1 Specifications of laptop

Description	Specifications
Model	Asus TUF Gaming FX505GD
Processor	Intel Core i5-8300H
Operating System	Windows 11
Graphic	NVIDIA® GeForce® GTX 1050, 4GB GDDR5
Memory	20GB DDR4 RAM
Storage	1TB SATA HDD

Since the laptop's webcam is not providing clear vision, an additional webcam has been acquired to improve the image quality. The specifications of the webcam used are listed in Table 3.2.2 below.

Table 3.2.2 Specification of webcam

Description	Specifications
Logitech C310 HD webcam	• Resolution: 720p • Frame Rate: 30 fps • Connection: USB 2.0

This webcam was chosen for its high resolution and frame rate, ensuring clear and smooth video for meetings and visioning sessions.

3.2.2 Software / Tools

To achieve the successful completion of the project, it's important to recognize that the efforts extend beyond hardware components. Software components are equally critical for achieving seamless integration and realizing our project objectives. Software serves as the backbone of our project, enabling the coordination and interaction of various elements.

Software takes on a pivotal role in the development process. It acts as the brain that orchestrates the functioning of the hardware components, making them work together harmoniously. Just as hardware provides the physical foundation, software provides the intelligence and instructions that guide the system's behavior.

In context, software is responsible for various tasks, such as implementing the algorithms that power our Automated License Plate Recognition (ALPR) system, managing data processing and storage, conducting real-time analysis, and presenting the results in a comprehensible format. Table 3.2.3 presents an overview of the tools and software utilized throughout the project to achieve its successful execution. This table provides insight into the essential components that contribute to the realization of the project's objectives.

Table 3.2.3 Specifications of software / tools

Software/Tools	**Logo**	**Specifications**
Google Colab		A free cloud-based platform provided by Google that allows users to write and execute Python code collaboratively. This tool will be utilized for machine learning and model training.
Visual Studio Code		A popular and widely used source-code editor developed by Microsoft. It's known for its versatility, efficiency, and extensibility.
Roboflow		A platform designed to simplify and accelerate the process of creating and deploying computer vision models for tasks like object detection, image classification, and more.

3.3 Methodologies

The way we're making this project happen follows clear steps. First, gather a bunch of different pictures of Malaysia license plates from the internet. Then, work on these pictures a lot, making them better for the needs and create data set. The data set need to be in different angle captured, light condition and quality. After that, annotate up the pictures with labels to teach the model. This labelled dataset helps the computer learn what to look for.

Next, the project involves the utilization of cutting-edge deep learning techniques, particularly CNNs, which have demonstrated remarkable proficiency in image-related tasks. The custom dataset will serve as the foundation for training the model. Continuous iterations of training and fine-tuning will take place to optimize the model's performance. The ultimate objective is to achieve enhanced accuracy, robustness, and real-time capabilities in license plate recognition.

The implementation of this project entails the use of various software and tools that synergistically contribute to the project's execution. The deep learning frameworks YOLOv5, YOLOv7 and YOLOv8 will be harnessed for object detection and model training. Besides that, SRGAN and ESRGAN methods will be implemented into the process to enhance the detected license plate images through super-resolution. Additionally, EasyOCR will be employed for optical character recognition (OCR) to extract alphanumeric information from license plates.

In addition, the project's success will be assessed based on specific improvements in key performance metrics. These metrics include accuracy, where the system should demonstrate a high rate of correctly identifying and recognizing license plates, particularly in challenging situations. Timing is also crucial; the system should perform quickly, delivering real-time results to ensure effective traffic flow management. Furthermore, the ALPR system must be robust in capturing and pre-processing images to achieve the highest recognition accuracy.

Last but not least, to verify the efficacy of the developed ALPR system, a comprehensive verification plan is in place. Diverse inputs will be tested, involving license plates captured under various lighting conditions, angles, and qualities. The system will be rigorously assessed based on its ability to accurately detect license plates, effectively segment characters, and recognize alphanumeric content. The verification process will validate the system's robustness across different scenarios, ensuring its reliability and effectiveness.

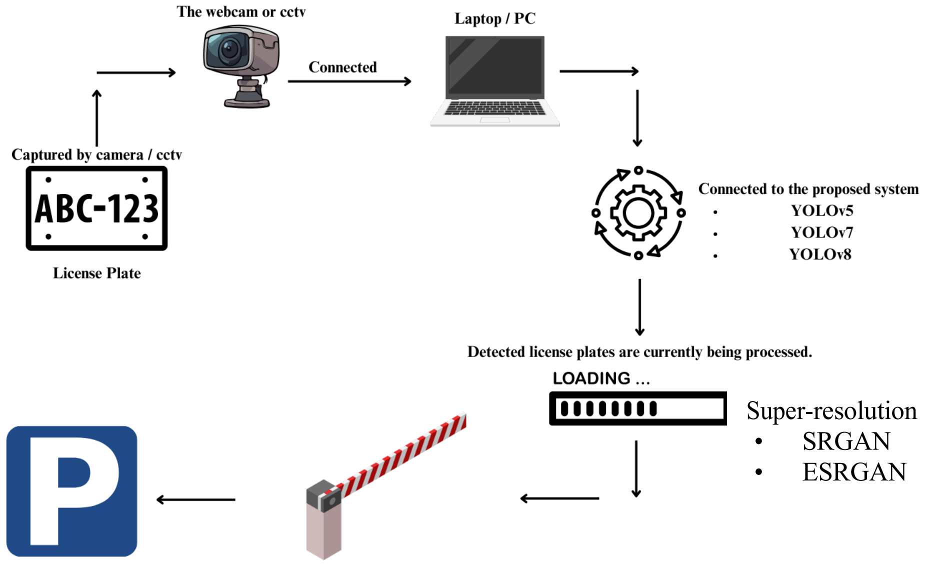
Incorporating these methodologies, tools, performance metrics, and verification procedures, the project aims to deliver an advanced ALPR system capable of significantly enhancing campus security and access control.

3.4 Block Diagram of ALPR system

In the ALPR system, the flow of the process is outlined in Figure 3.4.1, detailing the steps from the start to the end of the process. When a vehicle enters the university's entry area, the installed camera captures the license plate of the vehicle, which is then processed by the

ALPR system software. If the detected license plate number matches one in the database, the barriers in front will open, allowing the authenticated vehicle to enter the campus area for parking.

However, it's important to note that in this proposed ALPR system, the database is not set up and utilized. Instead, it will be established in the guard house to monitor and assess the accuracy of the proposed ALPR system. This setup allows for testing and validation of the system's functionality before full implementation.



Fige 3.4. Overview design of proposed ALPR system

3.5 System Design / Overview

This project involves the integration of hardware, software, and a custom dataset to achieve its objectives. Figure 3.4.1 illustrates the overall process and the system's design. Before beginning the development process, it is essential to install specific environments on the desktop to facilitate the project's execution. Table 3.4 below outlines the environments that need to be installed.

Environment	Version
Python	3.9.13

PyTorch	1.8.1+cu101
OpenCV	4.8.0
EasyOCR	1.7.0

Table 3.4 Environment to be installed

It's important to note that the versions of each environment must be installed correctly. Incompatibilities between different versions of environments can lead to malfunctioning. Now, let's begin by discussing the custom dataset. Despite the existence of various ALPR systems such as R-CNN, Faster R-CNN, and CNN in the market, our ALPR system offers unique advantages. It compiles diverse images from the internet that encompass various situations, angles, and image qualities, which are crucial for machine learning. A total of 1200 images have been carefully selected for the training process. These training images were obtained from Roboflow, a platform designed to simplify the creation, training, and deployment of computer vision models. Once the images are collected, annotations are added to each image, and the label "Number Plate" is applied. This annotated dataset is then used for machine learning purposes. Figure 3.4.2 illustrates the process of adding annotations and labelling license plates.

After finishing the annotation and labelling of the license plates, we will divide the license plates into separate sets for training, validation, and testing. Machine learning models necessitate a training set, validation set, and test set to guarantee their ability to learn from data, generalize effectively to new and unseen data, and be assessed accurately. These three sets serve distinct purposes in the process of constructing and evaluating machine learning models.

The division of data into these three sets is crucial to ensure that the machine learning model learns patterns from the training data, fine-tunes hyperparameters on the validation set, and ultimately demonstrates its true performance on unseen data with the test set. This process helps prevent issues like overfitting (where the model performs well on training data but poorly on new data) and provides a more precise estimation of a model's real-world performance. Figure 3.4.3 illustrates that the custom dataset has been partitioned into 1,100 images for the training set, 50 images for the validation set, and 50 images for the testing set.

Splitting a custom dataset into training, validation, and testing sets is crucial for effectively training YOLO models. The training set, which typically comprises 60-80% of the data, is used to train the model by adjusting its parameters to minimize the loss function. The validation set, typically 10-20% of the data, plays a crucial role in tuning hyperparameters and detecting overfitting. It provides an independent evaluation of the model's performance during training. Finally, the testing set, also 10-20% of the data, serves as an unbiased assessment of the model's generalization to unseen data. This ensures the model's real-world performance is accurately evaluated. By adhering to this split, we can optimize the model's performance, prevent overfitting, and obtain a reliable estimate of its capabilities on new, unseen data.

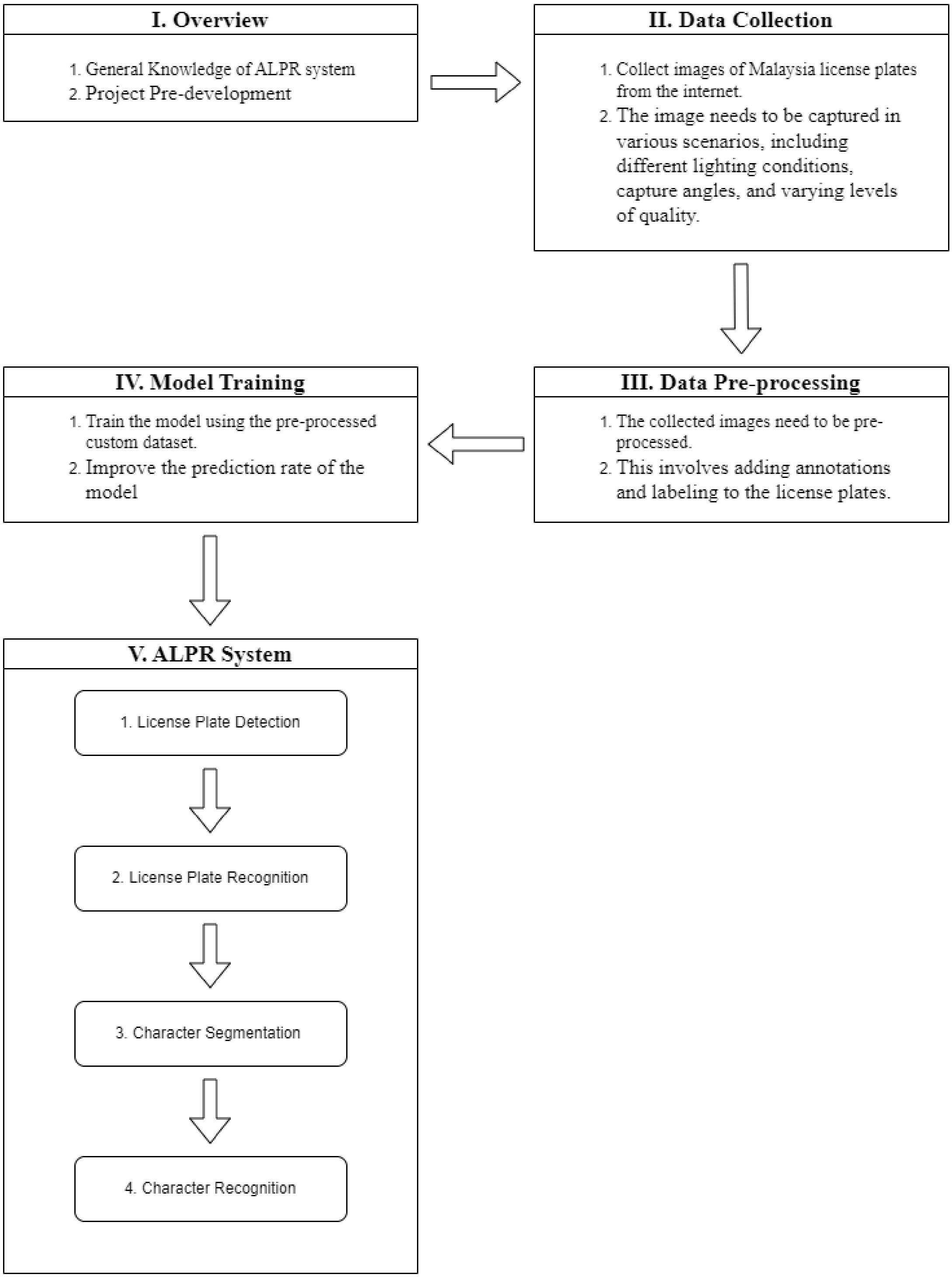


Figure 3.4.1 System Design Process Overview

Figure 3.4.2 The process of adding annotations and labelling license plates

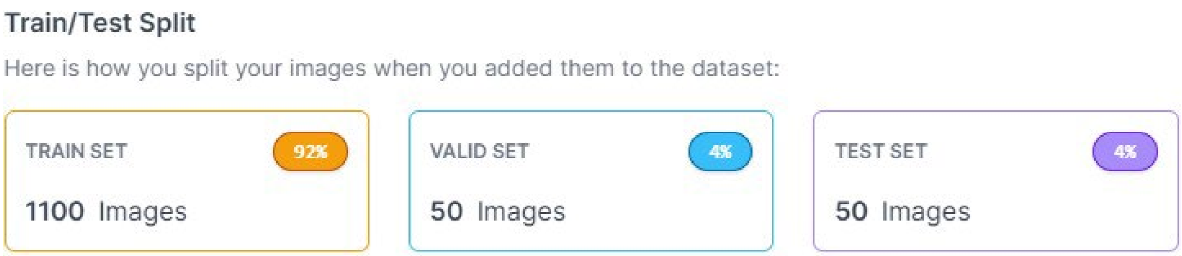


Figure 3.4.3 Splitting of custom data set

After exporting the custom dataset, we will obtain the annotated and labelled set separately. Now, we move to the next step, which involves training the algorithm for YOLO version 5 to YOLO version 8 to obtain pre-trained models for implementation and comparison in our ALPR system later.

3.6 Setting and Configuration

As training the model demands a GPU capable of handling multiple computations simultaneously, it facilitates the distribution of training processes and can significantly accelerate machine learning operations. GPUs allow the aggregation of numerous cores, utilizing fewer resources without compromising efficiency or power. However, the laptop in use has a limited GPU capacity with only 4GB of memory. Hence, we have chosen to utilize Google Colab. The services of Google Colab must be purchased to access the provided GPU for machine learning and model training for all YOLO models. To ensure the machine learning model becomes intelligent, robust, and proficient in license plate detection, it's crucial to run a specific number of epochs to minimize errors in the model as much as possible. For this project, different epochs will be run for the YOLO models.

The number of epochs to be run for all YOLO models is 50, 80, and 100. This will allow us to observe the differences between each epoch and evaluate their efficiency in the model's performance.

To evaluate the performance of the pre-trained model from different epochs, we will analyze the Precision, Recall, and mAP (mean Average Precision), with the inclusion of IoU (Intersection over Union) values set at both 0.5 (50%) and 0.95 (95%) to assess the model's performance during the training phase. Since the custom dataset, comprising 1200 images collected from the internet with various situations and conditions, has been acquired, a statistical analysis and real-time monitoring will be conducted to obtain the final performance measurements for all the models.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$

Equation 3.6 IoU

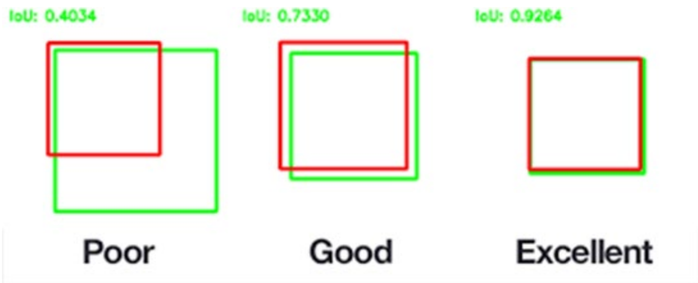


Figure 3.6.1 Intersection over Union

The details of each performance measurement are explained as follows:

- **Precision:** Precision measures the accuracy of the positive predictions.
- **Recall:** Recall measures the ability of the model to find all the relevant cases.

- **mAP_0.5:** Mean Average Precision at IoU threshold of 0.5. It measures the accuracy of object detection at a less strict IoU threshold.
- **mAP_0.5:0.95:** Mean Average Precision across all IoU thresholds from 0.5 to 0.95. This provides a broader measure of the model's accuracy across various IoU thresholds.

Moreover, the confusion matrix is another important consideration for machine learning to measure the performance of a classification model. The confusion matrix is a means of displaying the total number of accurate and inaccurate instances based on the model's predictions. It shows the number of instances produced by the model on the custom dataset of Malaysian vehicle license plates. The confusion matrix can be specified into four different fields:

- **True Positive (TP):** Models accurately predict a positive data point.
- **True Negative (TN):** Models accurately predict a negative data point.
- **False Positive (FP):** Model predicts a positive data point incorrectly.
- **False Negative (FN):** Model predicts a negative data point incorrectly.

Furthermore, to provide a clear understanding of how the confusion matrix can be used in the proposed model, refer to Figure 3.6.2 below:

		Actual	
		License Plate	Not License Plate
Predicted	License Plate	True Positive	False Positive
	Not License Plate	False Negative	True Negative

Figure 3.6.2 Confusion Matrix for License Plate

In literature review, it was mentioned that all the YOLO versions have different YOLO models, such as YOLOv5 having YOLOv5s (small), YOLOv5m (medium), YOLOv5l (large), and YOLOv5x (extra-large) models. For the proposed ALPR system, the medium model will be used, the same for both YOLOv7 and YOLOv8. The reason for choosing the medium model is that it offers a better balance between speed and accuracy compared to other models. All the processes of YOLO model training are depicted in Figure 3.6.3.

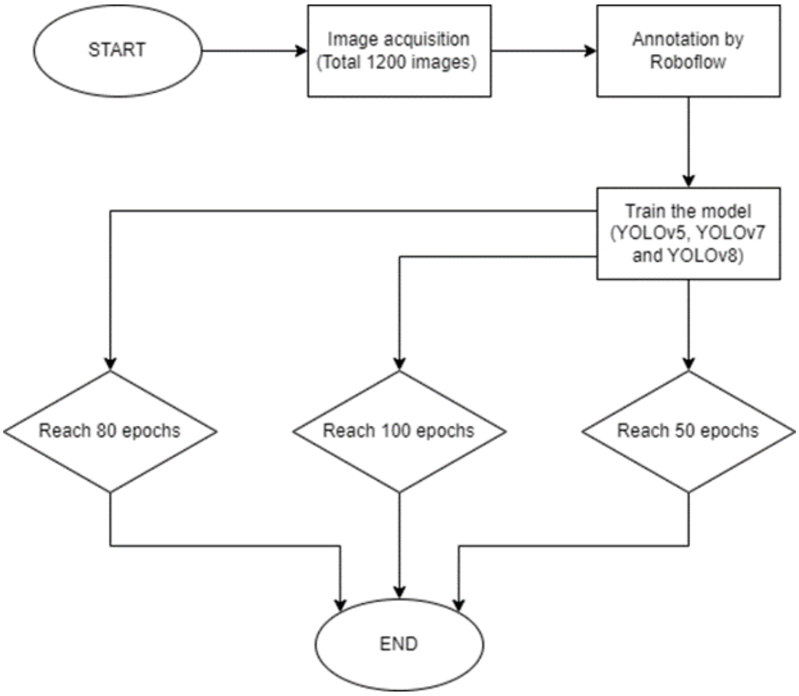


Figure 3.6.3 YOLO model training process

3.6.1 YOLOv5

The model training for YOLOv5m is executed in Google Colab to utilize the provided GPU. The results of the pre-trained model running in different epochs can be viewed in Table 3.6.1 to verify the functionality.

Total Epoch	Precision	Recall	mAP@0.5	mAP@0.5:0.95
50	88.69%	98.25%	98.10%	58.31%
80	90.04%	98.25%	97.92%	58.13%
100	90.28%	97.73%	97.45%	58.03%

Table 3.6.1: Model Training of YOLOv5m with Different Epochs

In general, the model trained with a total of 50 epochs has shown relatively higher metrics compared to the models trained with 80 and 100 epochs. This can be observed from the recall, mAP@0.5, and mAP@0.95, where it exhibits slightly higher percentages than the other two epochs of model training. However, the model trained with 80 epochs can be considered as the median or most stable compared to the other two epochs of training. Finally, when the model is trained with a total of 100 epochs, it shows a lower performance compared to the 50 and 80 epochs. The performance of the 100 epochs model dropped.

Although the training results for the three YOLOv5m models show slight differences, further experimentation needs to be conducted to draw more concrete conclusions about the models' performance.

Up to this point, the development of the ALPR system using the YOLOv5 algorithm is already more than halfway complete. The subsequent step involves implementing the model into our code using Python as the main programming language, chosen for its blend of user-friendliness, versatility, and an extensive collection of libraries and tools. In the programming phase, our focus will be on localizing a detected license plate through the model. Subsequently, the program will undertake pre-processing tasks and leverage the pre-processed images for segmentation. Ultimately, EasyOCR will be utilized for character recognition.

3.6.2 YOLOv7

For the model training of YOLOv7, the methodology closely follows that of YOLOv5, utilizing the same custom license plate dataset for training. The total number of epochs for all versions of the YOLO model is fixed at 50, 80 and 100 epochs to provide a comparison for the accuracy of the models. During the training process, the model's performance metrics such as loss and accuracy are closely monitored to gauge its progress. Table 3.6.2 illustrates the outcome of the pre-trained YOLOv7 model, showcasing its effectiveness in license plate recognition tasks. The pre-trained model provides a solid foundation for further fine-tuning and deployment in ALPR system.

Total Epoch	Precision	Recall	mAP@0.5	mAP@0.5:0.95
50	94.11%	84.21%	91.49%	51.31%
80	82.07%	96.39%	95.72%	57.42%
100	94.54%	91.23%	97.62%	59.66%

Table 3.6.2 Model Training of YOLOv7x with Different Epochs

A clear trend emerges from these results. The model's precision and mAP@0.5:0.95 consistently improved with more training epochs, indicating the model's ability to detect objects with high accuracy and over a broader range of IoU thresholds. Recall initially increased, showcasing the model's ability to capture more relevant objects, but it showed a slight decline at 100 epochs. This trade-off between precision and recall is essential, as it reflects the model's balance between minimizing false positives and false negatives.

In conclusion, the YOLOv7 model demonstrates promising performance in object detection tasks. The improvements in precision and mAP metrics suggest the model's capability to accurately detect objects, especially as the training progresses. The mAP@0.5:0.95 metric provides a comprehensive evaluation of detection quality across various IoU thresholds. The choice between the different epochs (50, 80, or 100) depends on the specific needs of the application. For applications requiring high precision and mAP, the model at 100 epochs might be preferred, while those focusing on a balance between precision and recall might find the model at 80 epochs suitable. Further fine-tuning and optimization could continue to enhance the model's performance for specific use cases.

3.6.3 YOLOv8

For YOLOv8 model training, the methodology follows a similar process as YOLOv5 and YOLOv7, utilizing the same custom license plate dataset. The training process is consistent with 50, 80 and 100 epochs to ensure model stability and accuracy. Table 3.6.3 illustrates the outcomes of the pre-trained YOLOv8 model, highlighting its effectiveness in license plate recognition tasks. The model's performance is evaluated against various metrics to ensure its robustness and generalization to unseen data.

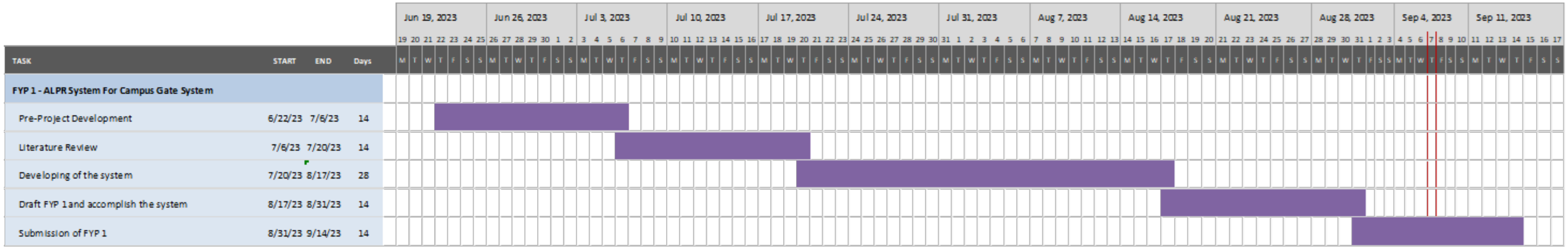
Total Epoch	Precision	Recall	mAP@0.5	mAP@0.5:0.95
50	95.33%	94.74%	98.86%	62.35%
80	96.43%	94.74%	98.5%	62.11%
100	94.82%	96.28%	98.90%	62.73%

Table 3.6.3 Model Training of YOLOv8m with Different Epochs

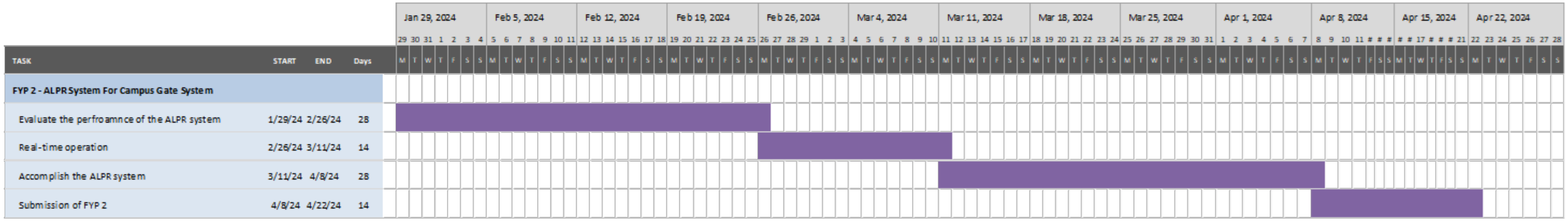
Comparing these results, the differences between the epochs are relatively minor. Precision and recall metrics show little variation, while mAP scores also remain consistently high. The precision scores ranged from 94.82% to 96.43%, recall from 94.74% to 96.28%, mAP@0.5 from 98.5% to 98.90%, and mAP@0.5:0.95 from 62.11% to 62.73%. These findings suggest that the model maintains a strong and stable performance throughout the training epochs, with no significant degradation or improvement in results beyond 80 epochs. Therefore, training the YOLOv8 model for 80 epochs seems to strike a balance between precision, recall, and mAP scores, providing an optimal performance without a substantial increase in computational cost beyond that point. However, the choice between 80 and 100 epochs may also depend on practical considerations such as available computational resources and training time.

3.7 Timeline

3.7.1 FYP 1 timeline



3.7.2 FYP 2 timeline



3.8 Selection of different YOLO model for testing

After conducting experiments with different total numbers of epochs for the YOLOv5, YOLOv7, and YOLOv8 models, which each have their own pre-trained weights, one of these models will be selected for testing in the next chapter.

3.8.1 YOLOv5 model

Based on these results, the model trained for 100 epochs has the highest precision (90.28%) and slightly lower recall (97.73%). The mAP@0.5 is 97.45% and the mAP@0.5:0.95 is 58.03%. These results indicate that the model at 100 epochs achieves a good balance between precision and recall, with a strong performance in mAP@0.5 as well.

However, it's important to consider that the differences between the models at 80 and 100 epochs are relatively small. The model at 80 epochs also performs well with a precision of 90.04%, recall of 98.25%, mAP@0.5 of 97.92%, and mAP@0.5:0.95 of 58.13%.

Therefore, based on these results and considering the minor differences, the model trained for 80 epochs seems to be the most suitable to use. It provides a good balance between precision, recall, and mAP scores, offering strong performance without a significant increase in computational cost compared to the 100-epoch model. Additionally, the 80-epoch model has slightly higher mAP scores, indicating good overall performance in object detection tasks.

3.8.2 YOLOv7 model

Comparing the model training for 80 epochs and 100 epochs, the model trained for 100 epochs has higher precision and mAP@0.5:0.95, indicating that it has better performance in terms of minimizing false positives and capturing a broader range of object detections. However, the model at 80 epochs has a higher recall and slightly lower precision and mAP@0.5:0.95, indicating that it might be more conservative in its detections but maintains a good overall balance.

Therefore, based on these results and considering the trade-offs between precision and recall, the model trained for 80 epochs might be more suitable to use in scenarios where higher recall (96.39%) is desired while still maintaining a decent precision (82.07%) and mAP scores. The 80-epoch model also has a reasonably high mAP@0.5 score (95.72%), indicating good performance at a stricter IoU threshold.

3.8.3 YOLOv8 model

Comparing these three models, all of them exhibit high precision, recall, and mAP@0.5 scores. However, the mAP@0.5:0.95 score, which considers a broader range of IoU thresholds, is also relatively high for all models, ranging from 62.11% to 62.73%.

Considering these results, the model trained for 100 epochs has the highest mAP@0.5:0.95 score (62.73%), indicating its ability to perform well across a wider range of IoU thresholds. Additionally, it has a good balance of precision (94.82%) and recall (96.28%).

Therefore, based on these results and considering the trade-offs between precision, recall, and mAP scores, the model trained for 100 epochs is likely the most suitable to use. It provides a high mAP@0.5:0.95 score, indicating good performance across different IoU thresholds, while maintaining strong precision and recall values. This model is well-rounded and should perform reliably for object detection tasks, capturing a wide range of objects with high accuracy.

3.9 Evaluation

In conclusion, the models that will be used for YOLOv5, YOLOv7, and YOLOv8 have been selected. The total number of epochs and their performances are shown in Table 3.9.

YOLO model	Total Epoch	Precision	Recall	mAP@0.5	mAP@0.5:0.95
YOLOv5	80	90.04%	98.25%	97.92%	58.13%
YOLOv7	80	82.07%	96.39%	95.72%	57.42%
YOLOv8	100	94.82%	96.28%	98.90%	62.73%

Table 3.9 Selected YOLO model

These three models will be further used in the next section for implementation and evaluation of performances based on static results and select the best model to use in next chapter.

3.10 Final Evaluation of YOLO models

To evaluate the performance of each YOLO model from section 3.9, a total of 10 number plates with various conditions are selected to validate the performance of each model. The conditions of each number plate include:

- A normal angle of the number plate (90 degrees).
- A fixed angle of the number plate (30 degrees).
- Day and night conditions of the number plate.

From the various types of number plate conditions, this testing can observe a total of 40 images to validate the performance of the model. In addition, the OCR method is added to all the models to evaluate the detection and recognition performances of each model.

To ensure consistency among all the models, the same methods will be applied, such as converting the detected number plate to grayscale and applying a Gaussian filter. The final performances of each model will be evaluated by their accuracy in detecting and recognizing the characters and numbers on the number plate. As illustrated in the figure below are the various conditionsof a number plate.

Figure 3.10.1 shows a number plate under daytime conditions and from different angles

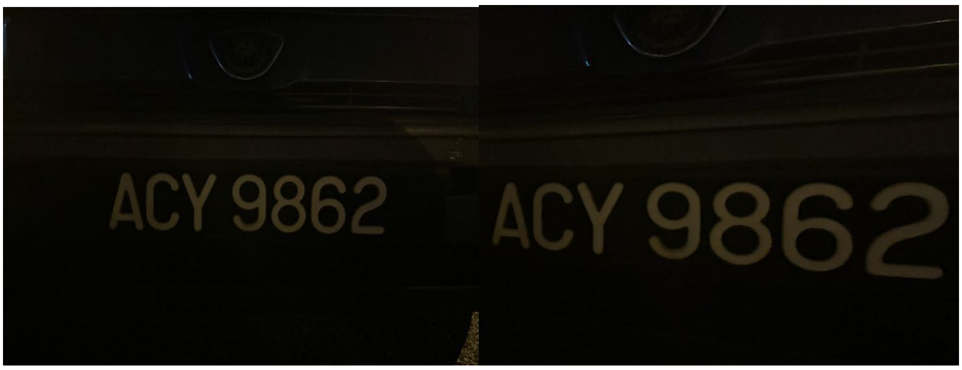


Figure 3.10.2 shows a number plate under nighttime conditions and from different angles

The evaluation of the performance of each model is listed in the table below:

YOLO model	Number Plate recognition	Error Rate	Success Rate
YOLOv5	13/40	67.5%	32.5%
YOLOv7	21/40	47.5%	52.5%
YOLOv8	25/40	37.5%	62.5%

Table 3.8 The performance of each model during number plate recognition

From the table above, it is evident that YOLOv5 exhibits the worst performance, whereas YOLOv8 demonstrates the best performance among the models. Even though the static experiment involved displaying number plates from a phone, the YOLOv8 model still demonstrated remarkable speed and accuracy in detecting and recognizing the number plates. The figure below illustrates the various conditions that affect the performance of detection and recognition.

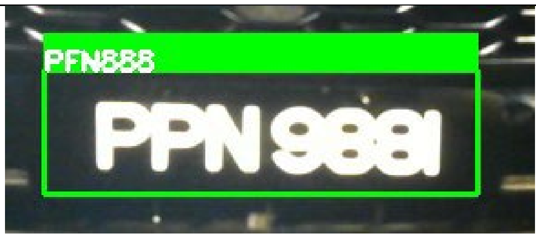
<table>
<tr><td rowspan="4">YOLOv5 model</td><td>
</td><td>Unable to recognize full number plate’s characters.</td></tr>
<tr><td>
</td><td>Unable to recognize full number plate’s characters.</td></tr>
<tr><td>
</td><td>Recognize the number plate wrongly with missing character.</td></tr>
<tr><td>
</td><td>When number plate is 30 degrees unable to detect.</td></tr>
</table>

Table 3.8.1 YOLOv5 model performance

YOLOv7 model		Able to recognize the upper part number plate's character only.
		Unable to recognize full number plate's characters due to character L is too close with A and F.
		Unable to recognize full number plate's characters.

Table 3.8.2 YOLOv7 model performance

YOLOv8 model		Unable to recognize full number plate's characters due to expose too much to light and reflected.
		Unable to recognize full number plate's characters when upside down.

Table 3.8.3 YOLOv8 model performance

Throughout the tables of YOLOv5, v7 and v8 can conclude that:

1. YOLOv5 perform poorly when due with angle of number plate not 90 degrees and cannot recognize the characters of number plate correctly.
2. YOLOv7 able to detect well but when recognize number plate, always missing characters and wrongly.

3. YOLOv8 performs well when dealing with number plates at different angles, but it performs poorly when the number plate is exposed to too much light and number plate characters is upside down.

3.11 Implementation Issues and Challenges

Throughout the implementation of all algorithms and frameworks, a multitude of issues and challenges have arisen. These challenges have the potential to hinder the progress as move forward with the project. Given that the development phase is inherently prone to encountering errors, it becomes imperative to address these errors promptly to ensure the smooth advancement of the system. However, the process of troubleshooting and resolving errors can be time-consuming, potentially impacting the project timeline.

The subsequent section provides a comprehensive list of the challenges encountered, shedding light on the intricacies involved in implementing the ALPR system.

1. The compatibility between different software components / environments.
2. Increased GPU usage during real-time detection.
3. Time-consuming model training.
4. Data collection and labelling.
5. Varied conditions of detected license plates.

In conclusion, while these challenges may pose obstacles, but also provide valuable learning opportunities that contribute to the refinement and enhancement of the overall project.

Chapter 4

System Evaluation and Discussion

4.1 YOLOv8 integrate with SRGAN and ESRGAN methods

In this project, the SRGAN and ESRGAN methods are proposed to integrate with the YOLOv8 model to develop an enhanced model for Automated License Plate System Recognition for Campus Gate System. When integrated with these two methods, the license plate detected by the pre-trained YOLOv8 model is passed through either SRGAN or ESRGAN. This process enhances the resolution and clarity of the license plate before proceeding to OCR.

SRGAN refer to Super-Resolution GAN methods while ESRGAN is the enhanced architecture of SRGAN method. The SRGAN model will be pre-trained using Google Colab to take advantage of GPU acceleration, which significantly shortens the training time. On the other hand, the ESRGAN model will utilize the pre-trained model from reference [51]. This approach allows for a comparison between three setups in this project: YOLOv8 without any GAN methods, YOLOv8 with the SRGAN method, and YOLOv8 with the ESRGAN method.

In the training of SRGAN, the key to optimization is minimizing the generator loss while simultaneously maximizing the discriminator loss. This approach leads to the generation of high-quality super-resolved images. The process occurs within the training loop, where the loss functions are computed and optimized. This process can be broken down into a few key points:

1. **Discriminator Loss (D_loss):**

   The discriminator aims to distinguish between real and generated (fake) images. The loss is formulated as:

$$D_{loss} = 1 - D(real\_img) + D(fake\_img)$$

Equation 4.1 Discriminator Loss

- D(real_img) is the output of the discriminator when given real images.

- D(fake_img) is the output of the discriminator when given fake images generated by generator.
- 1 is subtracted to encourage the discriminator to correctly classify real images as 1 and fake images as 0.

2. **Generator Loss (G_loss):**
   The generator aims to produce images that are realistic and close to the ground truth.
3. **Perceptual Loss:**
   The perceptual loss measures the difference between the features of the generated image and the ground truth image. It's calculated using a pre-trained deep neural network (typically a VGG network) to extract high-level features.
4. **Image Loss:**
   The image loss measures the pixel-wise difference between the generated image and the ground truth image, typically computed using mean squared error (MSE) or other similarity measures.
5. **Total Variation Loss:**
   The total variation loss encourages smoothness in the generated image by penalizing sharp changes between adjacent pixels.

By referencing author [60]'s proposed ideas, SRGAN is developed to pre-train an SRGAN model, which is then integrated with the YOLOv8 model. A total of 1000 epochs of training are run to train the SRGAN model, using a dataset containing 70 HR license plate images. These images are resized by dividing by 4 to create LR license plate images, as the SRGAN model will produce HR images with a 4x upscale after training.

However, when reaching the final training iteration, Loss_D (discriminator loss) is recorded as 1.0000, indicating that the discriminator is maximizing its loss, likely because it correctly classifies real and generated images. PSNR (Peak Signal-to-Noise Ratio) measures at 29.4483 dB, suggesting high image quality with low distortion. SSIM (Structural Similarity Index) is 0.9566, indicating high similarity between the generated and ground truth images.

Overall, these indicators suggest that the model has effectively achieved its objectives by generating high-quality super-resolution images that closely match the ground truth images.

The ESRGAN method is referenced from the authors of [51]'s proposed methods. By utilizing the author's pre-trained model and integrating it with YOLOv8, this integrated model will be compared with other methods to evaluate performance and obtain the best-suited results during real-time experiments in campus areas.

4.2 Testing setup

Three different models will be tested which are:

1. YOLOv8 without any additional methods.
2. YOLOv8 with SRGAN method.
3. YOLOv8 with ESRGAN method.

4.3 Static results

The same dataset used to evaluate all the YOLO models is reused to evaluate the performance of these two different models once again. Since the YOLOv8 model without any additional methods has been tested in a previous chapter, it will not be included in the static results. The results of three different models are shown in the table below:

Models	Number Plate recognition	Error Rate	Success Rate
YOLOv8 without any additional methods.	25/40	37.5%	62.5%
YOLOv8 with SRGAN method.	29/40	27.5%	72.5%
YOLOv8 with ESRGAN method.	34/40	15%	85%

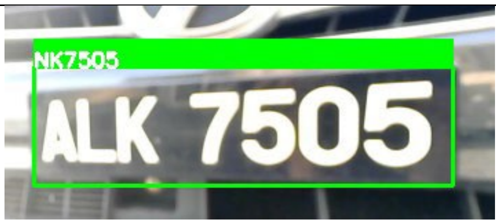
Table 4.2 All models' performance

From the table above, it is evident that the addition of different methods has resulted in a significant increase in performance. The YOLOv8 model integrated with SRGAN has shown a 10% increase in success rate compared to the normal YOLOv8 model. Similarly, the YOLOv8 model integrated with ESRGAN has exhibited a 22.5% increase in success rate compared to the normal YOLOv8 model. Both methods have demonstrated that the incorporation of additional techniques leads to an increase in the performance and accuracy of the ALPR system.

Although the results are quite satisfactory, the errors will be presented below for further discussion:

<table>
<tr><td rowspan="3">YOLOv8 with ESRGAN method</td><td></td><td>During recognition, unbale to recognize all the characters correctly.</td></tr>
<tr><td></td><td>During recognition, unbale to recognize all the characters correctly.</td></tr>
<tr><td></td><td>Unable to recognize full number plate’s characters when upside down.</td></tr>
</table>

Table 4.2.1 YOLOv8 with ESRGAN method encounter several issues

<table>
<tr><td rowspan="3">YOLOv8 with SRGAN method</td><td></td><td>The detection not able to detect the whole license plate.</td></tr>
<tr><td></td><td>When the license plate is 30 degrees, not able to recognize the characters correctly.</td></tr>
<tr><td></td><td>Unable to recognize full number plate’s characters when upside down.</td></tr>
</table>

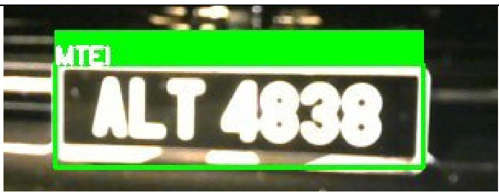
		The license plate images being overexposed to light leads to the OCR not functioning properly.

Table 4.2.2 YOLOv8 with SRGAN method encounter several issues

Both methods encounter similar issues, such as the inability to recognize license plates with upside-down characters. Additionally, some license plate images are overexposed to light, resulting in OCR malfunction.

4.4 Real-time monitoring

The experiment is set up at the guard house area of UTAR Kampar Campus to monitor and capture vehicles that pass through the guard house. The hardware setup is illustrated in the figure below, where a webcam is used to capture the license plate, and the ALPR program proceeds to detect and recognize the license plates of the vehicles passing through.

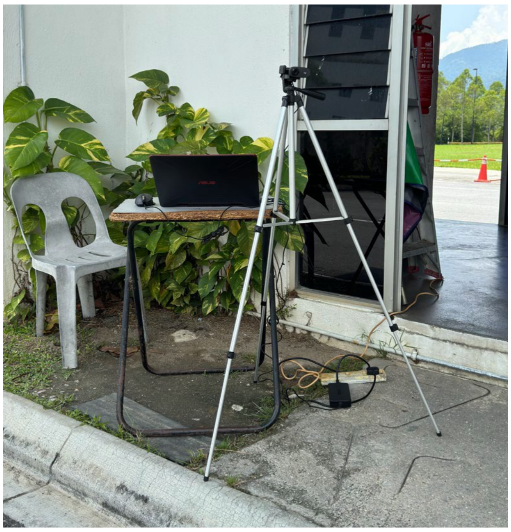


Figure 4.3.1 Hardware set up

Additionally, in the guard house area, there is a speed bump. When vehicles pass over it, they decelerate, allowing the ALPR system to detect and recognize the license plates within a limited time. The height of the webcam and the distance between the webcam and the vehicle are recorded. This data can be saved and analyzed in the future to improve the overall performance of the proposed ALPR system. In this experiment, the height of the webcam is 1.16 meters, and the distance between the webcam and the vehicle is approximately 4.71

meters. For each model, a total of 10 minutes will be allocated to evaluate performance, and recognized license plates will be recorded in an Excel file for further analysis. Since the Excel file will contain a large amount of data, only recognized license plate characters that conform to the correct Malaysia license plate format will be counted as true, while others will be considered false.

Although there's a speed bump, some vehicles may still accelerate, causing the ALPR system to be unable to recognize the license plate correctly. This issue will be further discussed in the following section.

Figure 4.3.2 the height and distance between the webcam and the vehicle

4.5 Real-time monitoring results

The recognized license plate is saved into an Excel file for further analysis. However, due to the large number of columns (more than 700) for each detected license plate, the detailed analysis will be conducted in the future upon deployment in the UTAR Kampar Campus area.

Nevertheless, all three models performed well in real-time detection and recognition, successfully saving the results into Excel for further analysis. The efficiency of the ALPR system in identifying passing car licence plates is shown in the image below.

Figure 4.3.3 a) ESRGAN method b) SRGAN method

4.6 Conclusion of the Experiments

Real-time monitoring has revealed several issues that must be addressed in the future:

1. All proposed ALPR systems are unable to detect number plates with upside-down characters, only recognizing those at the bottom.
2. False positives may occur if the vehicle's license plate is not centrally positioned.
3. The proposed ALPR system is unable to detect and recognize license plates within a limited time if the vehicle is moving too fast.
4. Adequate lighting conditions are necessary, as insufficient light or excessive shadow on the license plate will cause the OCR to malfunction.

These issues will be addressed and resolved in the future based on the recorded data. However, overall, the results are satisfactory, as the proposed ALPR system efficiently detects and recognizes the majority of vehicles passing through. Fine-tuning of both the SRGAN and ESRGAN methods will be conducted to achieve higher accuracy and effectiveness of the ALPR system.

Chapter 5

Conclusion

5.1 Conclusion

Before the development of the ALPR system, many people relied on access cards or parking credentials to enter the campus area. However, since the launch or development of the ALPR system, these traditional methods have become obsolete. The ALPR system has not only reduced the need for manpower but has also increased the safety of the campus area, protecting students and restricting unauthorized vehicles from entering.

The proposed Automated License Plate System Recognition for the Campus Gate System provides a convenient way for management to oversee vehicles entering and exiting the campus. All vehicle details and information can be collected and utilized for future analysis or enhancement of the ALPR system. By deploying the ALPR system on campus, operational costs can be reduced, and traffic congestion during peak hours, such as 8:00 am, can be alleviated due to the effectiveness and speed of the ALPR system. In addition, implementing a barrier gate in front of the guard house can allow the ALPR system to manage vehicle passage more effectively. Once the ALPR system is deployed on campus, the workload of the management team to register vehicles every semester can be significantly reduced, eliminating the need for car stickers to save costs and reduce waste.

Lastly, the completion of the proposed ALPR system is aimed at reducing traffic congestion in UTAR Kampar campus and benefiting all individuals, including students and staff, by resolving entry-related issues.

REFERENCES

[1]V. Sambhe, A. Bakshi, S. Gulhane, T. Sawant, and S. Udmale, “ALPR -An Intelligent Approach Towards Detection and Recognition of License Plates in Uncontrolled Environments Prognostics and Health Management (PHM) View Project Epileptic Seizure Classification and Prediction View Project Akshay Bakshi Veermata Jijabai Technological Institute 2 PUBLICATIONS 0 CITATIONS SEE PROFILE ALPR -An Intelligent Approach Towards Detection and Recognition of License Plates in Uncontrolled Environments,” ALPR - an Intelligent Approach Towards Detection and Recognition of License Plates in Uncontrolled Environments, Jan. 2023, doi: https://doi.org/10.1007/978-3-031-24848-1_18.

[2]D. R. Vedhaviyassh, R. Sudhan, G. Saranya, M. Safa, and D. Arun, “Comparative Analysis of EasyOCR and TesseractOCR for Automatic License Plate Recognition using Deep Learning Algorithm,” IEEE Xplore, Dec. 01, 2022. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10009215

[3]R. Laroca et al., “A Robust Real-Time Automatic License Plate Recognition based on the YOLO Detector.”

[4]J. Shashirangana, H. Padmasiri, D. Meedeniya, and C. Perera, “Automated License Plate Recognition: A Survey on Methods and Techniques,” IEEE Access, vol. 9, pp. 11203–11225, 2021, doi: https://doi.org/10.1109/access.2020.3047929.

[5]G. R. Gonçalves, S. P. G. da Silva, D. Menotti, and W. R. Schwartz, “Benchmark for license plate character segmentation,” Journal of Electronic Imaging, vol. 25, no. 5, p. 053034, Oct. 2016, doi: https://doi.org/10.1117/1.jei.25.5.053034.

[6]M. Khalid, “Vehicle License Plate Character Recognition by Neural Networks,” Nov. 2004.

[7]B.-F. . Wu, S.-P. . Lin, and C.-C. . Chiu, “Extracting characters from real vehicle licence plates out-of-doors,” IET Computer Vision, vol. 1, no. 1, pp. 2–10, Mar. 2007, doi: https://doi.org/10.1049/iet-cvi:20050132.

[8]S. Du, M. Ibrahim, M. Shehata, and W. Badawy, “Automatic License Plate Recognition (ALPR): A State-of-the-Art Review,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, no. 2, pp. 311–325, Feb. 2013, doi: https://doi.org/10.1109/tcsvt.2012.2203741.

[9]C. Buscht, R. Domert, C. Freytag', and H. Ziegler', “FEATURE BASED RECOGNITION OF TRAFFIC VIDEO STREAMS FOR ONLINE ROUTE TRACING,” May 1998.

[10]I. Paliyl, V. Turchenko', V. Koval', A. Sachenkol, and G. Markowsky2, “Approach to Recognition of License Plate Numbers Using Neural Networks,” Jul. 2004.

[11]S. Montazzolli and C. Jung, “Real-Time Brazilian License Plate Detection and Recognition Using Deep Convolutional Neural Networks,” 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Oct. 2017, doi: https://doi.org/10.1109/sibgrapi.2017.14.

References

[12]R. Laroca, L. Zanlorensi, G. Gonçalves, E. Todt, W. Robson Schwartz, and D. Menotti, "An Efficient and Layout-Independent Automatic License Plate Recognition System Based on the YOLO detector," An Efficient and Layout-Independent Automatic License Plate Recognition System Based on the YOLO Detector, doi: https://doi.org/10.1049/itr2.12030).
[13]N. Liyana Yaacob, A. Ahmed Alkahtani, F. M. Noman, A. W. Mahmood Zuhdi, and D. Habeeb, "License plate recognition for campus auto-gate system," Indonesian Journal of Electrical Engineering and Computer Science, vol. 21, no. 1, p. 128, Jan. 2021, doi: https://doi.org/10.11591/ijeecs.v21.i1.pp128-136.

[14]W. Wong and Keong, "Malaysian Automatic Number Plate Recognition System using Pearson Correlation," May 2016.

[15]P. Comelli, P. Ferragina, M. N. Granieri, and F. Stabile, "Optical recognition of motor vehicle license plates," IEEE Transactions on Vehicular Technology, vol. 44, no. 4, pp. 790–799, 1995, doi: https://doi.org/10.1109/25.467963.

[16]M. Sarfraz, M. Ahmed, and S. Ghazi, "Saudi Arabian License Plate Recognition System," Jul. 2003.

[17]P. Gokhlani, R. Jhamb, and S. Lakhani, "Automatic Vehicle Identification by Plate Recognition," 2021.

[18]R. Sharma, "Automatic License Plate Based Smart Vehicle Validation &Security by Gate Control &Email Send," Jul. 2021.

[19]M. Siam, "Practical Design of an Automatic License Plate Recognition Using Image Processing Technique," Certified International Journal of Engineering and Innovative Technology (IJEIT), vol. 9001, no. 4, pp. 2277–3754, 2008.

[20]M. Abdullah, M. Ali, H. Bakhtan, and S. Mokhtar, "NUMBER PLATE RECOGNITION OF MALAYSIA VEHICLES USING SMEARING ALGORITHM," Sci.Int.(Lahore), vol. 29, no. 4, pp. 823–827, 2017.

[21]I. Kusumadewi, C. A. Sari, D. R. I. Moses Setiadi, and E. H. Rachmawanto, "License Number Plate Recognition using Template Matching and Bounding Box Method," Journal of Physics: Conference Series, vol. 1201, no. 1, p. 012067, May 2019, doi: https://doi.org/10.1088/1742-6596/1201/1/012067.

[22]S. Nair, "Car Plates: Malaysia Rules and Regulations According To JPJ," drive safe & fast, Jul. 27, 2021. https://www.dsf.my/2021/07/jpj-malaysia-car-plates-specifications/ (accessed Sep. 07, 2023).

[23]I. Slimani, A. Zaarane, W. Al Okaishi, I. Atouf, and A. Hamdoun, "An automated license plate detection and recognition system based on wavelet decomposition and CNN," Array, vol. 8, p. 100040, Dec. 2020, doi: https://doi.org/10.1016/j.array.2020.100040.

[24]P. Agarwal, K. Chopra, M. Kashif, and V. Kumari, "Implementing ALPR for detection of traffic violations: a step towards sustainability," Procedia Computer Science, vol. 132, pp. 738–743, 2018, doi: https://doi.org/10.1016/j.procs.2018.05.085.

[25] “Department of Student Affairs (Kampar Campus),” dsa.kpr.utar.edu.my, Dec. 29, 2023. https://dsa.kpr.utar.edu.my/Vehicle-Stickers.php (accessed Feb. 24, 2024).

[26] “Access cards and parking credentials | Parking and Transportation - The University of Iowa,” *transportation.uiowa.edu*. https://transportation.uiowa.edu/parking/facilities/access-cards-and-parking-credentials (accessed Feb. 24, 2024).

[27]Y. Jamtsho, P. Riyamongkol, and R. Waranusast, “Real-time Bhutanese license plate localization using YOLO,” *ICT Express*, vol. 6, no. 2, pp. 121–124, Jun. 2020, doi: https://doi.org/10.1016/j.icte.2019.11.001.

[28]J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger.”

[29]J. Heras, “joheras/darknet-1,” *GitHub*, Feb. 20, 2022. https://github.com/joheras/darknet-1 (accessed Feb. 26, 2024).

[30]R. Laroca and D. Menotti, “Automatic License Plate Recognition: An Efficient and Layout-Independent System Based on the YOLO Detector.”

[31]Y. Kessentini, M. D. Besbes, S. Ammar, and A. Chabbouh, “A two-stage deep neural network for multi-norm license plate detection and recognition,” *Expert Systems with Applications*, vol. 136, pp. 159–170, Dec. 2019, doi: https://doi.org/10.1016/j.eswa.2019.06.036.

[32]H. Li, P. Wang, M. You, and C. Shen, "Reading car license plates using deep neural networks," Image and Vision Computing, vol. 72, pp. 14-23, 2018.

[33]J. Zhuang, S. Hou, Z. Wang, and Z. Zha, "Towards human-level license plate recognition," in European Conference on Computer Vision (ECCV), 2018, pp. 314-329.

[34]J. Wang, H. Huang, X. Qian, J. Cao, and Y. Dai, "Sequence recognition of Chinese license plates," Neurocomputing, vol. 317, pp. 149-158, 2018.

[35]S. M. Silva and C. R. Jung, "Real-time Brazilian license plate detection and recognition using deep convolutional neural networks," in Conference on Graphics, Patterns and Images (SIBGRAPI), 2017, pp. 55-62.

[36]M. Weber, "Caltech Cars dataset," [Online]. Available: http://www.vision.caltech.edu/Image_Datasets/cars_markus/cars_markus.tar, 1999. [Accessed: July 3, 2020].

[37]V. Srebric, "EnglishLP database," [Online]. Available: http://www.zemris.fer.hr/projects/LicensePlates/english/baza%20slika.zip, 2003. [Accessed: July 3, 2020].

[38]L. Dlagnekov and S. Belongie, "UCSD/Calit2 car license plate, make and model database," [Online]. Available: http://vision.ucsd.edu/belongie-grp/research/carRec/car_data.html, 2005. [Accessed: July 3, 2020].

[39]W. Zhou, H. Li, Y. Lu, and Q. Tian, "Principal visual word discovery for automatic license plate detection," IEEE Transactions on Image Processing, vol. 21, no. 9, pp. 4269-4279, 2012.

[40]G. S. Hsu, J. C. Chen, and Y. Z. Chung, "Application-oriented license plate recognition," IEEE Transactions on Vehicular Technology, vol. 62, no. 2, pp. 552-561, 2013.

[41]OpenALPR Inc., "OpenALPR-EU dataset," [Online]. Available: https://github.com/openalpr/benchmarks/tree/master/endtoend/eu, 2016. [Accessed: July 3, 2020].

[42]G. R. Gonçalves, S. P. G. da Silva, D. Menotti, and W. R. Schwartz, "Benchmark for license plate character segmentation," Journal of Electronic Imaging, vol. 25, no. 5, p. 053034, 2016.

[43]H. Sahota, "The History of YOLO Object Detection Models from YOLOv1 to YOLOv8," *Deci*, Jun. 05, 2023. https://deci.ai/blog/history-yolo-object-detection-models-from-yolov1-yolov8/

[44]R. Kundu, "YOLO: Real-Time Object Detection Explained," *www.v7labs.com*, Jan. 17, 2023. https://www.v7labs.com/blog/yolo-object-detection

[45]A. Mehra, "Evolution of YOLO Object Detection Model From V5 to V8," Labellerr, Mar. 31, 2023. https://www.labellerr.com/blog/evolution-of-yolo-object-detection-model-from-v5-to-v8/

[46]S. A. Sanchez, H. J. Romero, and A. D. Morales, "A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework," IOP Conference Series: Materials Science and Engineering, vol. 844, p. 012024, Jun. 2020, doi: https://doi.org/10.1088/1757-899x/844/1/012024.

[47]Z. Kelta, "YOLO Object Detection Explained: A Beginner's Guide," *www.datacamp.com*, Sep. 2022. https://www.datacamp.com/blog/yolo-object-detection-explained

[48]J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection."

[49]J. Nelson and J. S. Jun 12, "Responding to the Controversy about YOLOv5," Roboflow Blog, Jun. 12, 2020. https://blog.roboflow.com/yolov4-versus-yolov5/

[50]C.-Y. Wang, A. Bochkovskiy, and H.-Y. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors."

[51]X. Wang et al., "ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks."

[52]A. Jolicoeur-Martineau, "The relativistic discriminator: a key element missing from standard GAN."

References

[53]S. Combettes, “A basic intro to GANs (Generative Adversarial Networks),” *Medium*, Oct. 28, 2020. https://towardsdatascience.com/a-basic-intro-to-gans-generative-adversarial-networks-c62acbcefff3
[54] I. Goodfellow *et al.*, “Generative Adversarial Nets.”

[55]FILPAL, “A Simple Introduction to Generative Adversarial Network (GAN),” *Medium*, Oct. 22, 2023. https://filpal.medium.com/a-simple-introduction-to-generative-adversarial-network-gan-1ecef855625b (accessed Apr. 20, 2024).

[56]C. Dewi, R.-C. Chen, Y.-T. Liu, X. Jiang, and K. Hartomo, “Yolo V4 for advanced Traffic Sign Recognition with synthetic training data generated by various GAN,” *Yolo V4 for Advanced Traffic Sign Recognition with Synthetic Training Data Generated by Various GAN*, doi: https://doi.org/10.1109/ACCESS.2021.

[57]I. H. El-Shal, O. M. Fahmy, and M. A. Elattar, “License Plate Image Analysis Empowered by Generative Adversarial Neural Networks (GANs),” *IEEE Access*, vol. 10, pp. 30846–30857, 2022, doi: https://doi.org/10.1109/access.2022.3157714.

[58]G.-S. Hsu, J.-C. Chen, and Y.-Z. Chung, “Application-Oriented License Plate Recognition,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 2, pp. 552–561, Feb. 2013, doi: https://doi.org/10.1109/tvt.2012.2226218.

[59]C. Dong, C. C. Loy, K. He, and X. Tang, “Image Super-Resolution Using Deep Convolutional Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, Feb. 2016, doi: https://doi.org/10.1109/tpami.2015.2439281.

[60]C. Ledig et al., “Photo-Realistic single image Super-Resolution using a generative adversarial network,” Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, Jul. 2017, doi: 10.1109/cvpr.2017.19.

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 2, Year 3	Study week no.: 3
Student Name & ID: LIEW VOON CHOON, 21ACB04376	
Supervisor: Dr Vikneswary a/p Jayapal	
Project Title: Automated License Plate System Recognition For Campus Gate System	

1. WORK DONE

1. First FYP 2 meeting with Dr. Vikneswary, no work done yet.

2. WORK TO BE DONE

1. FYP 2 first meeting with Dr. Vikneswary. In this first meeting, Dr. Vikneswary has assigned me to complete some work for FYP 2:
   - Go through all the research papers and see what version of Yolo is used for the ALPR system.
   - Doing comparison with all the Yolo versions (from Yolo 5 until Yolo 8).
   - All in extra research paper knowledge for literature review

3. PROBLEMS ENCOUNTERED

1. First meeting no problem encountered.

4. SELF EVALUATION OF THE PROGRESS

1. Gain extra information on how to continue my FYP 2.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 2, Year 3	**Study week no.: 5**
Student Name & ID: LIEW VOON CHOON, 21ACB04376	
Supervisor: Dr Vikneswary a/p Jayapal	
Project Title: Automated License Plate System Recognition For Campus Gate System	

1. WORK DONE

1. Doing research of various YOLO version.
2. Add in more literature review for different YOLO versions.

2. WORK TO BE DONE

1. Develop and train different YOLO versions for evaluation.

2. Do comparison of different YOLO versions.

3. Capture static license plate for testing set

3. PROBLEMS ENCOUNTERED

1. No problem encountered.

4. SELF EVALUATION OF THE PROGRESS

1. Continue focusing on my FYP 2 with more efforts.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 2, Year 3	**Study week no.: 7**
Student Name & ID: LIEW VOON CHOON, 21ACB04376	
Supervisor: Dr Vikneswary a/p Jayapal	
Project Title: Automated License Plate System Recognition For Campus Gate System	

1. WORK DONE 1. Done comparison of various YOLO versions.
2. WORK TO BE DONE 1. Discussed with supervisor and Dr suggest me that add in another advance methods into the proposed ALPR system.
3. PROBLEMS ENCOUNTERED 1. Facing issue when developing YOLOv6, supervisor advise to ignore version 6 continue with version 5, 7 and 8.
4. SELF EVALUATION OF THE PROGRESS 1. Everything is still following the schedule.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 2, Year 3	**Study week no.: 9**
Student Name & ID: LIEW VOON CHOON, 21ACB04376	
Supervisor: Dr Vikneswary a/p Jayapal	
Project Title: Automated License Plate System Recognition For Campus Gate System	

1. WORK DONE

1. Doing research for any advance methods can add into the proposed system.
2. Various methods have been find and need to discuss with supervisor.
3. Supervisor suggest try to add in GAN method to make the system more advance.

2. WORK TO BE DONE

1. Literature review of GAN methods.
2. Go through every type of GAN and find the most suitable methods to be use.

3. PROBLEMS ENCOUNTERED

1. GAN methods are new to me and already not follow up with the schedule timeline for FYP 2.

4. SELF EVALUATION OF THE PROGRESS

1. Keep putting efforts and try my best to follow up the timeline of FYP 2.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 2, Year 3	Study week no.: 11
Student Name & ID: LIEW VOON CHOON, 21ACB04376	
Supervisor: Dr Vikneswary a/p Jayapal	
Project Title: Automated License Plate System Recognition For Campus Gate System	

1. WORK DONE

1. Approach with supervisor about already get ideas on how to implement GAN methods into the proposed system.

2. WORK TO BE DONE

1. Complete the program and doing testing on static result and real-time monitoring to evaluate the performance.
2. Complete the FYP 2 report.

3. PROBLEMS ENCOUNTERED

1. Only left two weeks and its very rush to complete all the things.

4. SELF EVALUATION OF THE PROGRESS

1. Put more efforts to complete all the work.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 2, Year 3	Study week no.: 13
Student Name & ID: LIEW VOON CHOON, 21ACB04376	
Supervisor: Dr Vikneswary a/p Jayapal	
Project Title: Automated License Plate System Recognition For Campus Gate System	

1. WORK DONE

1. Complete all the testing.
2. Pass up the FYP 2 report to supervisor.

2. WORK TO BE DONE

1. Prepare presentation slides

3. PROBLEMS ENCOUNTERED

1. No problem encountered for the last week.

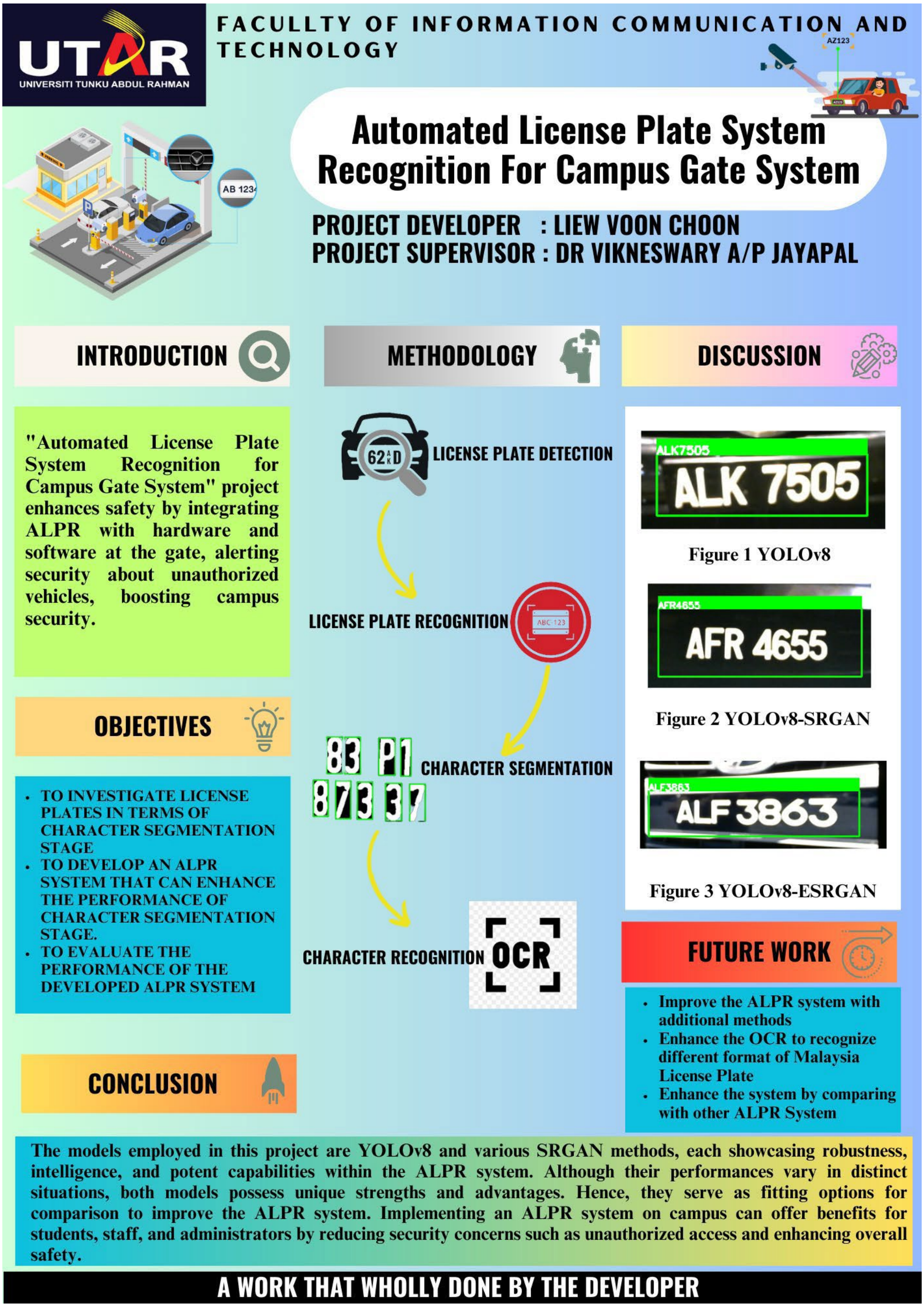
4. SELF EVALUATION OF THE PROGRESS

1. The efforts finally paid off.

Supervisor's signature

Student's signature

POSTER



UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
FACULLTY OF INFORMATION COMMUNICATION AND TECHNOLOGY
Automated License Plate System Recognition For Campus Gate System
PROJECT DEVELOPER : LIEW VOON CHOON
PROJECT SUPERVISOR : DR VIKNESWARY A/P JAYAPAL
INTRODUCTION
"Automated License Plate System Recognition for Campus Gate System" project enhances safety by integrating ALPR with hardware and software at the gate, alerting security about unauthorized vehicles, boosting campus security.
OBJECTIVES
• TO INVESTIGATE LICENSE PLATES IN TERMS OF CHARACTER SEGMENTATION STAGE
• TO DEVELOP AN ALPR SYSTEM THAT CAN ENHANCE THE PERFORMANCE OF CHARACTER SEGMENTATION STAGE.
• TO EVALUATE THE PERFORMANCE OF THE DEVELOPED ALPR SYSTEM
METHODOLOGY
LICENSE PLATE DETECTION
LICENSE PLATE RECOGNITION
CHARACTER SEGMENTATION
CHARACTER RECOGNITION
OCR
DISCUSSION
ALK 7505
Figure 1 YOLOv8
AFR 4655
Figure 2 YOLOv8-SRGAN
ALF 3863
Figure 3 YOLOv8-ESRGAN
FUTURE WORK
• Improve the ALPR system with additional methods
• Enhance the OCR to recognize different format of Malaysia License Plate
• Enhance the system by comparing with other ALPR System
CONCLUSION
The models employed in this project are YOLOv8 and various SRGAN methods, each showcasing robustness, intelligence, and potent capabilities within the ALPR system. Although their performances vary in distinct situations, both models possess unique strengths and advantages. Hence, they serve as fitting options for comparison to improve the ALPR system. Implementing an ALPR system on campus can offer benefits for students, staff, and administrators by reducing security concerns such as unauthorized access and enhancing overall safety.
A WORK THAT WHOLLY DONE BY THE DEVELOPER

PLAGIARISM CHECK RESULT

FYP2_Report_CT_LIEW VOON CHOON.pdf

ORIGINALITY REPORT

17%	13%	10%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Ibrahim H. El-Shal, Omar M. Fahmy, Mustafa A. Elattar. "License Plate Image Analysis Empowered by Generative Adversarial Neural Networks (GANs)", IEEE Access, 2022 Publication	1%
2	www.researchgate.net Internet Source	1%
3	ijeecs.iaescore.com Internet Source	1%
4	Submitted to Middle East Technical University Student Paper	1%
5	www.labellerr.com Internet Source	1%
6	doczz.net Internet Source	1%
7	Rayson Laroca, Luiz A. Zanlorensi, Gabriel R. Gonçalves, Eduardo Todt, William Robson Schwartz, David Menotti. "An efficient and layout-independent automatic license plate recognition system based on the YOLO	<1%

	detector", IET Intelligent Transport Systems, 2021 Publication	
8	ycesa.com Internet Source	<1 %
9	Submitted to Queen Mary and Westfield College Student Paper	<1 %
10	Jithmi Shashirangana, Heshan Padmasiri, Dulani Meedeniya, Charith Perera. "Automated License Plate Recognition: A Survey on Methods and Techniques", IEEE Access, 2020 Publication	<1 %
11	Submitted to Southern New Hampshire University - Continuing Education Student Paper	<1 %
12	Submitted to University of Birmingham Student Paper	<1 %
13	arxiv.org Internet Source	<1 %
14	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1 %
15	dokumen.pub Internet Source	<1 %
16	hdl.handle.net	

	Internet Source	<1 %
17	www.ijraset.com Internet Source	<1 %
18	deepai.org Internet Source	<1 %
19	Christine Dewi, Rung-Ching Chen, Yan-Ting Liu, Xiaoyi Jiang, Kristoko Dwi Hartomo. "Yolo V4 for advanced Traffic Sign Recognition with synthetic training data generated by various GAN", IEEE Access, 2021 Publication	<1 %
20	Submitted to National University of Ireland, Maynooth Student Paper	<1 %
21	www.mdpi.com Internet Source	<1 %
22	Submitted to CSU, San Jose State University Student Paper	<1 %
23	www.v7labs.com Internet Source	<1 %
24	Submitted to Liverpool John Moores University Student Paper	<1 %
25	mdpi-res.com Internet Source	<1 %

26	Submitted to Singapore Polytechnic Student Paper	<1 %
27	Submitted to University of Wales central institutions Student Paper	<1 %
28	www.breeam.com Internet Source	<1 %
29	"Computer Vision – ECCV 2018 Workshops", Springer Science and Business Media LLC, 2019 Publication	<1 %
30	"Computer Vision – ECCV 2020 Workshops", Springer Science and Business Media LLC, 2020 Publication	<1 %
31	www.ijritcc.org Internet Source	<1 %
32	Pramod Kanjalkar, Samyak Jain, Rutuj Saraf, Jyoti Kanjalkar. "Chapter 9 Intelligent Inventory Management in Retail Stores Using Four State of the Art Object Detection Models", Springer Science and Business Media LLC, 2024 Publication	<1 %
33	shravankumar147.substack.com Internet Source	<1 %

34	thesai.org Internet Source	<1 %
35	Submitted to Institute of Aeronautical Engineering (IARE) Student Paper	<1 %
36	Nur Liyana Yaacob, Ammar Ahmed Alkahtani, Fuad M. Noman, Ahmad Wafi Mahmood Zuhdi, Dhuha Habeeb. "License plate recognition for campus auto-gate system", Indonesian Journal of Electrical Engineering and Computer Science, 2021 Publication	<1 %
37	Submitted to University of North Carolina, Greensboro Student Paper	<1 %
38	Submitted to Kingston University Student Paper	<1 %
39	Submitted to UOW Malaysia KDU University College Sdn. Bhd Student Paper	<1 %
40	eprints.utem.edu.my Internet Source	<1 %
41	Submitted to Mepco Schlenk Engineering college Student Paper	<1 %
42	Submitted to University of Bradford Student Paper	

		<1%
43	dfrws.org Internet Source	<1%
44	"MultiMedia Modeling", Springer Science and Business Media LLC, 2020 Publication	<1%
45	Lecture Notes in Computer Science, 2014. Publication	<1%
46	Raj Sarode, Samiksha Varpe, Omkar Kolte, Leena Ragha. "Image Super Resolution using Enhanced Super Resolution Generative Adversarial Network", ITM Web of Conferences, 2022 Publication	<1%
47	Submitted to Sunway Education Group Student Paper	<1%
48	export.arxiv.org Internet Source	<1%
49	fabacademy.org Internet Source	<1%
50	discovery.researcher.life Internet Source	<1%
51	etda.libraries.psu.edu Internet Source	<1%

Plagiarism Check Result

52	www.medrxiv.org Internet Source	<1 %
53	eprints.utar.edu.my Internet Source	<1 %
54	www.coursehero.com Internet Source	<1 %
55	www.frontiersin.org Internet Source	<1 %
56	Aashish Nair, Jay Deshmukh, Akash Sonare, Tarun Mishra, Richard Joseph. "Image Outpainting using Wasserstein Generative Adversarial Network with Gradient Penalty", 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), 2022 Publication	<1 %
57	Submitted to University of Glasgow Student Paper	<1 %
58	Submitted to University of Hertfordshire Student Paper	<1 %
59	Submitted to University of Hong Kong Student Paper	<1 %
60	cdn.techscience.cn Internet Source	<1 %
61	www.grafiati.com	

	Internet Source	<1%
62	Submitted to University of Northumbria at Newcastle Student Paper	<1%
63	"Communications, Signal Processing, and Systems", Springer Science and Business Media LLC, 2020 Publication	<1%
64	Submitted to Institute of International Studies Student Paper	<1%
65	Submitted to Universiti Teknologi Malaysia Student Paper	<1%
66	"Computer Vision – ECCV 2018", Springer Science and Business Media LLC, 2018 Publication	<1%
67	Submitted to Brunel University Student Paper	<1%
68	theses.gla.ac.uk Internet Source	<1%
69	web.inf.ufpr.br Internet Source	<1%
70	www.igi-global.com Internet Source	<1%

71	Ajaz Ahmad Mir, Mahesh Patel. "Machine learning approaches for adequate prediction of flow resistance in alluvial channels with bedforms", Water Science & Technology, 2024 Publication	<1 %
72	Submitted to Heriot-Watt University Student Paper	<1 %
73	Submitted to Indian Institute of Technology Roorkee Student Paper	<1 %
74	Jian Sha, Xinyu Chen, Yaxin Chang, Man Zhang, Xue Li. "A spatial weather generator based on conditional deep convolution generative adversarial nets (cDCGAN)", Climate Dynamics, 2023 Publication	<1 %
75	Phu Hoang Ng, An Mai, Huong Nguyen. "Chapter 7 Building anAI-Powered IoT App forFall Detection Using Yolov8 Approach", Springer Science and Business Media LLC, 2023 Publication	<1 %
76	Submitted to University of Liverpool Student Paper	<1 %
77	medium.com Internet Source	<1 %

Plagiarism Check Result

78	upcommons.upc.edu Internet Source	<1 %
79	Submitted to Coastline Community College Student Paper	<1 %
80	Submitted to The University of Buckingham Student Paper	<1 %
81	Submitted to University of Bahrain Student Paper	<1 %
82	library.wrds.uwyo.edu Internet Source	<1 %
83	scholarship.tricolib.brynmawr.edu Internet Source	<1 %
84	fastercapital.com Internet Source	<1 %
85	www.sta.ca.gov Internet Source	<1 %
86	"Fourth International Congress on Information and Communication Technology", Springer Science and Business Media LLC, 2020 Publication	<1 %
87	Haifeng Wan, Lei Gao, Manman Su, Qinglong You, Hui Qu, Qirun Sun. "A Novel Neural Network Model for Traffic Sign Detection and	<1 %

	Recognition under Extreme Conditions", Journal of Sensors, 2021 Publication	
88	Submitted to Intercollege Student Paper	<1 %
89	Leonardo Assumpção Moreira, Livia Moreira Poelking, Alan Salomão Graça, Hideo Araki. "Use of Generative Adversarial Network Algorithm in Super-Resolution Images to Increase the Quality of Digital Elevation Models Based on ALOS PALSAR Data", Anuário do Instituto de Geociências, 2023 Publication	<1 %
90	Seda Kul, Sarper Kumcu, Ahmet Sayar. "Docker Container-Based Framework of Apache Kafka Node Ecosystem: Vehicle Tracking System by License Plate Recognition on Surveillance Camera Feeds", International Journal of Intelligent Transportation Systems Research, 2024 Publication	<1 %
91	Submitted to Staffordshire University Student Paper	<1 %
92	gwern.net Internet Source	<1 %
93	www.ijeast.com Internet Source	<1 %

94	www.ijert.org Internet Source	<1 %
95	www.nrel.gov Internet Source	<1 %
96	www.science.gov Internet Source	<1 %
97	"Image Analysis and Processing - ICIAP 2017", Springer Science and Business Media LLC, 2017 Publication	<1 %
98	Advances in Intelligent Systems and Computing, 2015. Publication	<1 %
99	Jianan Li, Xiaodan Liang, Yunchao Wei, Tingfa Xu, Jiashi Feng, Shuicheng Yan. "Perceptual Generative Adversarial Networks for Small Object Detection", 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017 Publication	<1 %
100	Saiful Azhari Mohammad, Handri Santoso. "Predict stock prices using the Generative Adversarial Networks", SinkrOn, 2022 Publication	<1 %
101	Submitted to University of Adelaide Student Paper	<1 %

102	Submitted to University of Melbourne Student Paper	<1 %
103	Submitted to University of North Texas Student Paper	<1 %
104	Wu Liu, Xinchen Liu, Huadomg Ma, Peng Cheng. "Beyond Human-level License Plate Super-resolution with Progressive Vehicle Search and Domain Priori GAN", Proceedings of the 2017 ACM on Multimedia Conference - MM '17, 2017 Publication	<1 %
105	aishwaryabhatmth522.sites.umassd.edu Internet Source	<1 %
106	assets.researchsquare.com Internet Source	<1 %
107	data.bloomberglp.com Internet Source	<1 %
108	github.com Internet Source	<1 %
109	iris.unitn.it Internet Source	<1 %
110	worldwidescience.org Internet Source	<1 %
111	www.radioeng.cz Internet Source	<1 %

Plagiarism Check Result

112	www.sciencegate.app Internet Source	<1 %
113	www.theseus.fi Internet Source	<1 %
114	A. Murat Tekalp. "Deep Learning for Image/Video Restoration and Super-resolution", Foundations and Trends® in Computer Graphics and Vision, 2022 Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

Plagiarism Check Result

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	LIEW VOON CHOON
ID Number(s)	21ACB04376
Programme / Course	COMPUTER ENGINEERING
Title of Final Year Project	AUTOMATED LICENSE PLATE SYSTEM RECOGNITION FOR CAMPUS GATE SYSTEM

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 17 **%** **Similarity by source** Internet Sources: 13 % Publications: 10 % Student Papers: 9 %	
Number of individual sources listed of more than 3% similarity: 0	

Parameters of originality required and limits approved by UTAR are as Follows:
(i) **Overall similarity index is 20% and below, and**
(ii) **Matching of individual sources listed must be less than 3% each, and**
(iii) **Matching texts in continuous block must not exceed 8 words**
Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Name: Dr. Vikneswary Jayapal

Date: 24/04/24

Signature of Co-Supervisor

Name: ______________

Date: ______________

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	**21ACB04376**
Student Name	**LIEW VOON CHOON**
Supervisor Name	**Ts Dr Vikneswary a/p Jayapal**

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
✓	Title Page
✓	Signed Report Status Declaration Form
✓	Signed FYP Thesis Submission Form
✓	Signed form of the Declaration of Originality
✓	Acknowledgement
✓	Abstract
✓	Table of Contents
✓	List of Figures (if applicable)
✓	List of Tables (if applicable)
✓	List of Symbols (if applicable)
✓	List of Abbreviations (if applicable)
✓	Chapters / Content
✓	Bibliography (or References)
✓	All references in bibliography are cited in the thesis, especially in the chapter of literature review
✓	Appendices (if applicable)
✓	Weekly Log
✓	Poster
✓	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
✓	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

____

(Signature of Student)

Date: **23/4/2024**