

**ARTIFICIAL INTELLIGENCE FOR
CLOUD-ASSISTED OBJECT DETECTION**

CHAN WEN JIE

UNIVERSITI TUNKU ABDUL RAHMAN

**ARTIFICIAL INTELLIGENCE FOR
CLOUD-ASSISTED OBJECT DETECTION**

CHAN WEN JIE


**A project report submitted in partial fulfilment of the
requirements for the award of Bachelor of Electrical and Electronic
Engineering with Honours**

**Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman**

May 2023

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : 

Name : Chan Wen Jie

ID No. : 18UEB04018

Date : 22nd May 2023

APPROVAL FOR SUBMISSION

I certify that this project report entitled “**ARTIFICIAL INTELLIGENCE FOR CLOUD-ASSISTED OBJECT DETECTION**” was prepared by **CHAN WEN JIE** as met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Electrical and Electronic Engineering with Honours at Universiti Tunku Abdul Rahman.

Approved by,

Signature

:



Supervisor

:

Ir Ts Dr Tham Mau Luen

Date

:

22nd May 2023

Signature

:

Co-Supervisor

:

Date

:

The copyright of this report belongs to the author under the terms of the copyright Act 1987 as qualified by Intellectual Property Policy of Universiti Tunku Abdul Rahman. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2023, Chan Wen Jie. All right reserved.

ACKNOWLEDGEMENTS

I would like to thank everyone who contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Ir Ts Dr Tham Mau Luen for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, I would also like to express my gratitude to my loving parents, friends, and coursemates who had helped and given me encouragement, motivation, and support while I have been conducting this project.

Furthermore, I would like to thank everyone that provide informative feedback when I posted questions in the Intel® DevCloud forum. Their suggestions helped me to solve most of the problems.

ABSTRACT

The research focuses on the integration of artificial intelligence (AI) and cloud computing to develop a License Plate Recognition (LPR) model. The existing LPR model is a combination of the detection (YOLOv4-tiny) model and the recognition (ResNet-FC) model. The LPR model is then deployed from the local environment to the Intel® Developer Cloud for the Edge for further improvement (e.g. addition to the feature of selection of the inferencing engine provided by Intel). After deployment, this LPR model can be accessed from anywhere, anytime as long as internet connectivity is available. The purpose of the LPR model is to detect license plates in videos uploaded by users, reducing the need for manual monitoring and recording of car plate numbers. Next, further improvement was made by replacing the existing detection model (YOLOv4-tiny) with the more advanced version by using the YOLOv7 series. Subsequently, the detection model (YOLOv7-tiny) with the highest Mean Average Precision (mAP)_{@.5} of 0.936, and mAP_{@.5:.95} of 0.720, will replace the YOLOv4-tiny. Among the commonly used Intel hardware, the inference engine with Intel® Core™ i7-1185G7E and Intel® Iris® Xe Graphics 530 GPU integrated with the CPU had the highest performance. This inference engine had a processing time of 29 s and a frame per second (FPS) of 5.7 when applying to the LPR system with the YOLOv4-tiny detection model.

TABLE OF CONTENTS

DECLARATION	1
APPROVAL FOR SUBMISSION	2
ACKNOWLEDGEMENTS	4
ABSTRACT	5
TABLE OF CONTENTS	6
LIST OF TABLES	9
LIST OF FIGURES	10
LIST OF SYMBOLS / ABBREVIATIONS	12
CHAPTER	
1	
INTRODUCTION	14
1.1	14
1.2	15
1.3	16
1.4	17
1.5	18
1.6	20
1.7	21
2	
LITERATURE REVIEW	22
2.1	22
2.2	23
2.3	24
2.3.1	26
2.4	30
2.5	31
2.5.1	31
2.5.2	32
2.5.3	32
2.5.4	33

		7
2.6	Edge Computing	35
	2.6.1 Edge AI	36
2.7	Case Study	37
	2.7.1 Molecular Dynamics Simulations on Cloud Computing and Machine Learning Platforms	37
	2.7.2 Smart Healthcare Analysis and Therapy for Voice Disorder using Cloud and Edge Computing	38
2.8	Summary	40
3	METHODOLOGY AND WORK PLAN	41
3.1	Introduction	41
3.2	Methodology	43
	3.2.1 Hardware Overview	43
	3.2.2 Software Overview	45
	3.2.3 Intel® Developer Cloud for the Edge Hardware Specifications (Cloud Resources Overview)	53
3.3	Work Plan	59
	3.3.1 Workflow	59
	3.3.2 Programming Flow Chart	67
	3.3.3 Gantt Chart	69
3.4	Work Budget	69
3.5	Summary	71
4	RESULTS AND DISCUSSION	72
4.1	Introduction	72
4.2	Evaluation of Intel Hardware	72
4.3	Comparison of Performances of the Yolov7 series	75
	4.3.1 FPS & Latency	75
	4.3.2 Benchmarking	76
4.4	Benefits of the Project	79
	4.4.1 Minimal Setup Effort	79
	4.4.2 Portable	80
	4.4.3 Free Trial on Intel Hardware	81

		8
4.5	Drawbacks of the project	81
4.5.1	Internet Connectivity	82
4.5.2	Chinese and Other Languages Characters	82
4.5.3	Fix Cloud Resources	82
4.6	Summary	83
5	CONCLUSIONS AND RECOMMENDATIONS	84
5.1	Conclusions	84
5.2	Recommendations for Future Work	85
	REFERENCES	86

LIST OF TABLES

Table 2.1: Comparison of Baseline Object Detectors	29
Table 3.1: Intel Hardware Price Comparison	58
Table 3.2: Uploading Speed for Each Size of File	59
Table 3.3: Training Time for YOLOv7, YOLOv7-tiny, and YOLOv7x	66
Table 3.4: Comparison of Budgets between the Local and Semi-local Setups	70
Table 4.1: Comparison between Intel® Core™ i7-1185G7E & Intel® Xeon® Gold 6338N	74
Table 4.2: Number of Layers & Parameter of Each Neural Network	76
Table 4.3: Benchmarking of Yolov7 Series	76
Table 4.4: Confusion Matrix	77

LIST OF FIGURES

Figure 2.1: Relationship between AI, ML and DL	23
Figure 2.2: Detection of Objects: Cat and Person	24
Figure 2.3: Percentage of respondents that used TensorFlow in the 2021 Kaggle Professional ML developer survey	25
Figure 2.4: Differences between the original segmentation model (left) over the PointRend enhancement (right).	25
Figure 2.5: Dot Products of the Input Pixels and the Filter in the Convolution layer	27
Figure 2.6: Downsampling in Pooling Layer	27
Figure 2.7: Framework of the CNNs	28
Figure 2.8: Comparison with Other Real-time Object Detectors	28
Figure 2.9: The overview of the user workflow for the Intel® DevCloud	34
Figure 2.10: The results of the Safety Gear Detection.	34
Figure 3.1: Project's Workflow	42
Figure 3.2: Intel NUC BXNUC10i7FNH3	44
Figure 3.3: Sonnet eGPU Breakaway Box 750	45
Figure 3.4: VMware Workstation Logo	45
Figure 3.5: Ubuntu Logo	46
Figure 3.6: Python Logo	46
Figure 3.7: OpenCV Logo	47
Figure 3.8: Jupyter Notebook Logo	48
Figure 3.9: Google Colaboratory Logo	48
Figure 3.10: Anaconda Logo	49
Figure 3.11: PyTorch Logo	50
Figure 3.12: Intel® Distribution of OpenVINO™ Toolkit Basic Working Flow	50

Figure 3.13: Intel® Distribution of OpenVINO™ Toolkit Logo	51
Figure 3.14: MyBinder Logo	52
Figure 3.15: Progress Bar for LPR Inferencing	60
Figure 3.16: Generated Nodes' Codes	61
Figure 3.17: Display the Inferred Video	62
Figure 3.18: Example of License Plate Labelling using labelling	64
Figure 3.19: Role of ONNX in Bridging the Development of ML Model	66
Figure 3.20: Overall Programming Flow Chart	68
Figure 3.21: Gantt Chart for Entire Project	69
Figure 3.22: UP Squared AI Vision X Developer Kit	71
Figure 4.1: Inferencing Engine Processing Time	73
Figure 4.2: Inferencing Engine FPS	73
Figure 4.3: Yolov7 Series Processing Time	75
Figure 4.4: Yolov7 Series FPS	75
Figure 4.5: Demonstration on the IoU	78
Figure 4.6: Smartphone for Inferencing	80

LIST OF SYMBOLS / ABBREVIATIONS

AI	Artificial Intelligent
LPR	License Plate Recognition
ARPANET	Advanced Research Projects Agency Network
CCTV	Closed-Circuit Television
O&M	Operations and Maintenance
ML	Machine Learning
DL	Deep Learning
API	Application Programming Interface
Mask R-CNN	Mask Region-based Convolutional Neural Network
CNN	Convolutional Neural Network
SaaS	Software as a service
PaaS	Platform as a Service
IaaS	Infrastructure as a Service
AWS	Amazon Web Services
CPU	Central Processing Unit
iGPU	Integrated Graphics Processing Unit
FPGA	Field-Programmable Gate Array
VPU	Vision Processing Unit
RTOS	Real-Time Operating System
OTA	Over the Air
OS	Operating System
YOLO	You Only Look Once
MS COCO	Microsoft Common Objects in Context
E-ELAN	Extended Efficient Layer Aggregation Network
DPM	Distributed Power Management
DRS	Distributed Resource Scheduler
IT	Information Technology
TDP	Thermal Design Power
AVS	Advanced Vector Extensions
SGX	Software Guard Extensions
TCC	Time Coordinated Computing
DevCloud	Developer Cloud

GUI	Graphical User Interface
OpenCV	Open Source Computer Vision
MMX	Multi-Media Extensions
SSE	Streaming SIMD Extensions
SIMD	Single Instruction, Multiple Data
IR	Intermediate Representation
ONNX	Open Neural Network Exchange
PC	Personal Computer
RAID array	Redundant Array of Independent Disks
PCIe	Peripheral Component Interconnect express
FPS	Frame Per Second
mAP	mean Average Precision
ROI	Region-of-Interest
IoT	Internet of Things
SLPs	Speech-Language Pathologists

CHAPTER 1

INTRODUCTION

1.1 General Introduction

In general, artificial intelligence (AI) is a technology that allows machines to replicate the capabilities of the human mind and perform tasks close to human behaviour. While in cloud computing, it is a technique to carry out the ordinary computer's capability in the virtual world instead of the physical hardware. For example, arithmetic & logic operations and manipulating information & data. In this research, AI in cloud computing will be carried out. With this, making smart, networked experiences feasible can be achieved.

In the 1940s, the concept of AI was yet to take shape, but the concept of developing a model for building a neural network was introduced. This concept gave birth to AI and raised the attention of all technologists to publish papers and bring out their points of view during the 1950s (Schroer, 2023). With the advancement of technology and industry revolution, AI is finally mature and deployed into the market for commercial use.

In real life, there are a lot of applications of AI. For example, human face detection, autonomous driving, and optical character recognition. Hence, to have a better illustration, license plate detection will be used throughout the whole research as an AI application. In Malaysia, one of the most popular shopping outlets had implemented the "Smart Parking System" which utilized License Plate Recognition (LPR).

On the other hand, the raw idea of cloud computing is initialized to connect people and data from anywhere at any time in order to achieve the purpose of multiple people doing the same task (Regalado, 2011). This idea was coined in the 1960s by Joseph Carl Robnett Licklider during his work at ARPANET (Advanced Research Projects Agency Network). Before the late 1990s, the cloud was still not capable of computing but just for storage usage. This can be supported by Foote (2021) who stated that the cloud was used to express the empty space between the provider and the end user during the early stage of the 1990s.

As internet services get more stable and faster, the implementation and development of cloud computing are getting started. Nowadays, one of the widely used cloud computing services is Google Docs.

After understanding the brief history of these technologies, both of them (AI in cloud computing) will be used in producing a better model. This technique became more popular, especially during the covid-19 pandemic, almost all of the business, teaching, and trading is moved to online mode. Thus, in this research, the LPR model will be built and then the model will be deployed into the cloud environment in order to achieve the objectives.

In short, the basic working principle of this model on the cloud is to receive an input which is a video that is uploaded by the user. The model will detect the license plate in the video and capture that particular frame image to be computed. The results will be stored on the cloud and sent back to the user. This kind of operation will greatly reduce the effort of labour to continuously monitor and record the car plate number manually through the Closed-Circuit Television (CCTV).

1.2 Importance of the Study

The rise of artificial intelligence (AI) and the development of cloud computing have finally taken place together. AI capabilities assist businesses to become more productive, advantageous and insight-driven in the field of enterprise cloud computing. The flexibility, mobility, and cost savings that come with managing data and applications on the cloud are advantageous to enterprises.

Due to the significant amount of AI and cloud computing, businesses can oversee data, find trends and insights in data, improve consumer experiences, and enhance operations. According to Datacenters.com Cloud (2022), it expressed the more specific ways how AI is affecting cloud computing such as using AI to power up a self-managing cloud, taking advantage of the dynamic cloud services, and using AI to improve the data management.

Furthermore, the impact of AI in cloud computing also contributes to preserve and conserve the environment. This is because most of the paperwork, and documentation is uploaded and stored in the cloud. The usage of paper is greatly reduced. With this, the cutting down of tree is lesser and reduce the environmental negative effect.

To illustrate the importance of LPR, the Sunway Pyramid is a good example. Sunway Pyramid has more than 10,000 parking lots and also has an average daily traffic volume of 50,000 cars. It is a great success as the LPR model deployed by SHENZHEN JIESHUN company, the current recognition rate has reached 99.5% (Parking Network, 2021). This indicated that the system is stable to be used and shows that the data management is in a proper and effective way.

However, since cloud service is involved, the security and privacy issues must be taken care of very well. It is very common to see everyone have a terminal on their hands in the modern era. Hence, anyone who has an ulterior motive could launch a cyber-attack towards any company/individual to get the information. Based on Triskele Labs (2023), cyber-attacks on these sites (Amazon, Google, and Microsoft) have quickly escalated in recent years. Incidentally, 20% of all cyberattacks in 2020 were cloud-based, this statistic showed that the cloud computing platform is the third most attacked cyberspace.

Therefore, it is important to study both positive and negative impacts on the society and environment when a new technology is invented and before launched into the market.

1.3 Problem Statement

Although AI is a common technique nowadays, it requires a high specification of computer or processor to carry out the computation. Especially for entrepreneurs who have business internationally. Tons of data needed to be stored after the computation is done. This induced another storage is required. In other words, the Operations and Maintenance (O&M) costs will increase as more hardware is needed to be purchased in order to accommodate tons of data. Indirectly, this also consumes physical space.

Hence, it is way better to carry out the computing in the cloud environment instead of in the edge terminal. This is because the cloud can act as the virtual world to store huge amounts of data which also can be said the storage is infinite. Thus, the cost is saved. With this, a win-win situation can be achieved.

In this research, the ALPR model will be built and then deploy to the cloud. The AI of ALPR will detect and store the detected car plate character in

the cloud. The cloud environment used will be the Intel® Developer Cloud for the Edge. With this AI in cloud computing, the delay in sending & receiving data from the terminals can be reduced, and the AI can analyse which car is frequently in & out of the building. In the future, the ALPR model may be intelligent enough to also compare the scanned license plate with the one in the database to determine whether the car is illegal or not.

1.4 Aim and Objectives

This project aims to develop a smart object detection model and deploy the model in the cloud with the minimum negative effect. The negative effects such as the increase in latency, the decrease in transmission of data rate, the increase in data losses, and most crucial security issues. Hence, another aim is to compare the model performance between cloud and edge platforms before the model can be used commercially.

The objectives of the projects are as follows:

1. To select the suitable cloud service provider.
2. To train a lightweight and accurate license plate detection model.
3. To deploy the optimized model with the highest mAP in the cloud environment.

1.5 Scope and Limitation of the Study

The scope of this project is to develop an AI model for car-plate detection & recognition model. After the model is successfully run on the local devices, the model will be deployed into the cloud environment. So, that the objectives of the project can be achieved.

One of the limitations of this study is I have limited free resources for cloud-based services. When I first looked for the Huawei Cloud, this service was not user-friendly and had a lot of charges. The prices are also confusing as they have a few pricing systems. Yet, some of the services are the prerequisite before the cloud service can be used. With these, I give up on Huawei Cloud and go for Intel® Developer Cloud for the Edge (Intel® DevCloud). Due to time constraints, only a quick look is taken, but it is more user-friendly and has a neat interface.

Furthermore, the limitation will be the data transmission rate since the cloud is used. If the speed is low, the performance will definitely drop and not be consistent. Hence, a better antenna or a place with a strong signal is needed to ensure the upload and download speeds are at the maximum. For extra information, some regions in Kuala Lumpur, Malaysia had implemented fifth-generation wireless (5G) technology. This is the most recent cellular technology, which was created to dramatically accelerate and boost the responsiveness of wireless networks. (Gillis, 2022). Hopefully, 5G can be widely implemented in Malaysia, so that this project can achieve a better result.

Next, the possible limitation will be faced in the future is the storage shortage on the cloud. The model deployment and also the continuous storing of data could use up the resources in a short period of time. Hence, this might induce another cost to top up the space for storage. For instance, Google Colab that used to test the workability of the training of the object detection model. Google Colab has a storage limitation of 108 GB, with only 77 GB available for users. Although this is typically sufficient for most tasks, it's important to be mindful of this limitation when working with larger datasets such as image or video data. In addition, the main platform used in this project – Intel® Developer Cloud for the Edge, is only providing us with storage of 50 GB (Intel, n.d.). Fortunately, the training was not done in this platform, instead, this

platform is only to deploy the AI model and test the performance of the Intel hardware.

Moreover, the services (Google Colab, and Intel® Developer Cloud for the Edge) have limited runtime. For Google Colab, it can address the limitation in processing power, but it has a time constraint of 12 hours for free GPU usage (Google, 2023b). On the other hand, Intel® Developer Cloud for the Edge has a free runtime of 10 hours. Nonetheless, they are still suitable for this project as it involves a small-scale dataset.

An additional limitation of the study is the availability of Malaysia's license plate dataset for model training. Although such a dataset can be easily obtained from Google Images, there is a lack of reliable sources for proper citation, and the number of images in the dataset is limited. This may affect the accuracy when performing the license plate detection.

Besides, the YOLOv7 model is considered as a new technology as it was only released on 7 July 2022 by Wong Kin Yiu (Wong, 2022). Hence, documentation and open source are very less, and most of the time, the self-develop in coding is required. Especially, when optimising and converting the YOLOv7 into Intermediate Representation (IR) format, lastly deploying the YOLOv7 model into Intel® Developer Cloud for the Edge, a lot of functions needed to be self-coded.

Last but not least, the real-time detection is unable to be performed. This is because the resources in the Intel® Developer Cloud for the Edge are shared among all the users. Therefore, Intel only allows us to submit the job to be queued to the nodes that consist of the targeted Intel hardware. When it is our turn, the unused targeted Intel hardware will perform the computation and send the results back. After the result is sent back, the occupied Intel hardware will be released and let the next user use it.

1.6 Contribution of the Study

This study has contributed to the practical implications for businesses involved in license plate recognition. The study analyses and measures the performance of the Intel hardware in performing car-plate detection & recognition. By evaluating the performance of the Intel hardware, the organization can have a better budget planning to purchase suitable Intel hardware for their own applications.

The system proposed in this study implements several techniques aimed at improving the accuracy of the detection model and reducing excessive computation. These techniques include centroid tracking and replacing the existing YOLOv4-tiny with YOLOv7-tiny.

Furthermore, in this project, one of the main subjects is the usage of the cloud. The main purpose for using cloud computing is due to its scalability. For example, the user bought an electronic device (smartphone) that is specified for gaming, but the user just performs daily tasks such as phone calling and messaging. This is a waste of resources. This same goes for cloud resources, the cloud service provider only provides the resources when the user requested them. When the user finished the intensive computing task, the unused resources will be released back to the cloud service provider, then the resources will be passed to the next user. This results in the optimization of the resources.

Lastly, this study eliminated the expensive local setup. The only requirement is a computer or laptop that can access the internet. Except for the training of the detection model is done by edge computing, all other computing is done in the cloud environment. This is very cost-effective and excessive budget can be allocated to another project or better usage.

1.7 Outline of the Report

The outline of the report consists of the following chapters:

1. Introduction

- Provides a clear and concise overview of the license plate detection and recognition AI model
- Emphasis on the importance of the study, and problem statement, followed by the aim and objectives of the project
- Highlight the scope, limitations and contribution

2. Literature Review

- Divide and discuss all the techniques and technologies that are related to the project
- Review the existing detection models
- Highlight the technique chosen to be used in this project

3. Methodology and Work Plan

- Plan the whole project timeline, milestones
- Discuss the hardware and software included in this project
- State the flow of the design

4. Results and Discussion

- Present the result of the project (license plate detection and recognition)
- Show the accuracy of the detection by using different models in the YOLOv7 series
- Demonstrate the deployment of the system into the cloud environment
- Illustrate the performance of the Intel hardware

5. Conclusion

- Conclude all the key findings
- Summarize the whole project by stating the project's achievement
- Identify the areas that still need development and enhancement
- Recommendations based on the understanding of the project's constraints

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, the literature on AI and cloud service will be reviewed, so that afterwards the combination result of AI in cloud-assisted object detection can be understood easier. Various applications were also reviewed to understand the advantages of the technique or platform chosen and how the balance can be achieved. In layman's terms, AI in cloud computing can refer to a virtual human life performing a task in a virtual environment and then sending the outcome to real people.

Before the object detection model is deployed into the cloud environment, the development and training were carried out at the local device. Object detection is used to distinguish and lock the targeted object on an input photo. Unlike traditional image processing which gives the result of the main object focusing on the photo, the object detection model is able to narrow down and put the square frame on a particular area to indicate the object. This objection detection model is one of the best examples to show the application of AI.

Next, cloud computing is utilized in providing remote access to storage, virtual machine or documentation. The convenience of the cloud is that it is able to allow anyone from anywhere, as long as the internet network is present, the person can carry out his/her tasks from different terminals.

Hence, it could be a huge benefit to us when these two-technology merged together. This could help a lot in the medical field to make the diagnosis more accurate; in day-care centres to monitor the needs and detect abnormal behaviour; and in business to reduce the burden of workers in analysing the data manually.

2.2 Artificial Intelligence (AI)

AI was defined as a “new technical science that studies and develops theories, methods, techniques, and application systems for simulating and extending human intelligence” (Huawei Enterprise Support Community, 2022). When it comes to AI, the relationship between AI, Machine Learning (ML) and Deep Learning (DL) cannot be ignored. AI is like an umbrella to both ML and DL, as it is a fundamental concept. Figure 2.1 shows the relationship between AI, ML and DL.

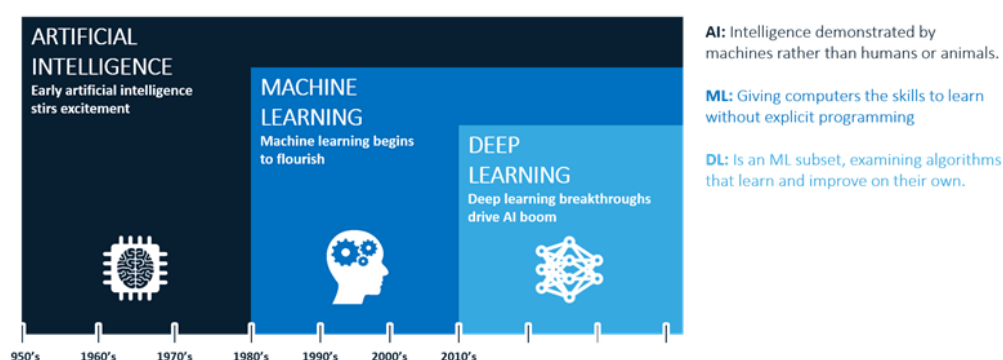


Figure 2.1: Relationship between AI, ML and DL (USoft, 2018)

ML refers to an AI system that can learn on its own using an algorithm. In other words, systems that continually learn without human intervention are also referred to as ML. The difference between DL and ML is just that DL is utilizing massive data sets. Due to the extensive learning required for intelligent behaviour, ML is used in the majority of AI projects (TechGig, 2020).

In general, the capability of three major schools of thought: Symbolism, Connectionism and Behaviourism determines whether the AI is strong or weak. Various types of AI can be applied according to specific scenarios. Several popular examples are computer vision, video processing, natural language processing, autonomous driving, and object detection. The object detection will be discussed in the next section.

2.3 Object Detection Model

Object detection is actually a challenging computer vision task because the position of the targeted object in the image needs to be predicted and defined the type correctly. To illustrate the object detection technique, Figure 2.2 shows the results of an image after going through the model.



Figure 2.2: Detection of Objects: Cat and Person (Fritz AI, 2022)

In the normal case, at first glance, the accuracy of 90% of the model indeed is very great, but it can be improved up to 99%. But the small gap is a big challenge to all the developers as a lot of problems will emerge. For example, imbalance of data set, overfitting issues, and low image resolution.

According to Brownlee (2019), one of the most recent techniques is the Mask Region-based Convolutional Neural Network (Mask R-CNN) model for object recognition tasks which utilized the Keras. Keras is a simple, flexible and powerful Application Programming Interface (API) for the TensorFlow library. According to Keras (2022), the popularity of both TensorFlow and PyTorch in Kaggle (a professional data science company) is shown in Figure 2.3.

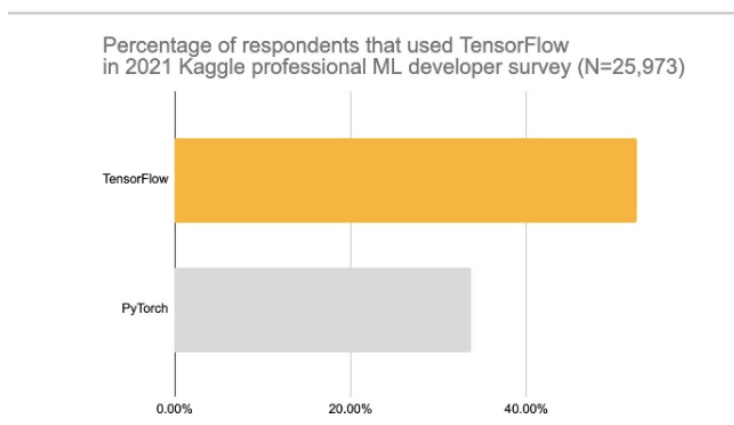


Figure 2.3: Percentage of respondents that used TensorFlow in the 2021 Kaggle Professional ML developer survey (Keras, 2022)

With this rate, PyTorch may lose its competitiveness and be replaced by TensorFlow in the future. Back to Mask R-CNN, it is quite different from the Convolutional Neural Network (CNN). CNN is a deep learning method that can recognise different objects and elements in an input image by assigning them different weights and biases that may be learned (Saha, 2018). On the other hand, Mask R-CNN is a state-of-the-art model for instance segmentation and is commonly used with a point-based rendering neural network module called “PointRend” in order to obtain a clean segmentation border (ArcGis Developers, 2022). Figure 2.4 below shows the differences between the original segmentation model over the PointRend enhancement.

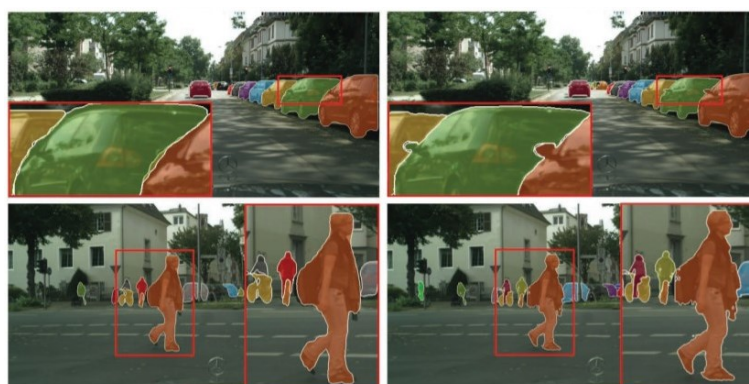


Figure 2.4: Differences between the original segmentation model (left) over the PointRend enhancement (right). (ArcGis Developers, 2022)

2.3.1 YOLOv7

In this project, the object detection model used to improve the existing LPR system will be the YOLOv7 series. The YOLOv7 stands for “You Only Look Once” version 7. The well-known family of YOLO is a collection of end-to-end deep-learning models developed for quick object detection. In YOLO, it carried out all of its predictions with the assistance of a single fully connected layer neural network. The YOLO technique utilises a simple but effective deep convolutional neural network (CNN) to identify objects in an input image (Kundu, 2023).

Convolutional Neural Networks (CNNs) share similarities with traditional Artificial Neural Networks (ANNs) as they consist of neurons that learn and optimize themselves. Each neuron in a CNN takes input and executes the algorithm, such as a non-linear function and scalar product., similar to ANNs. From the initial input of raw picture vectors through the final output of the class score (the weight), the neural network as a whole refers to a single scoring function. The last layer of the network includes loss functions related to classes, and common techniques used in ANNs apply to CNNs as well (O’Shea and Nash, 2015).

In general, three different layer types make up CNN: convolutional, pooling, and fully-connected layers (Gu et al., 2018). For better illustration, let the input be the colour image as it also suits this project. This means that an input image has three dimensions representing its RGB values, height, width, and depth. Firstly, in the convolutional layer, to detect specific features in the image, a filter/kernel moves across the image's receptive fields, scanning for the desired feature. This process is known as convolution. The filter calculates the dot product between the input pixels and itself, and the result is added to an output array until the filter covered the entire image. The output of a series of dot products between the input and filter is called a feature map or activation map (Yamashita et al., 2018). This can be shown in Figure 2.5 below.



Figure 2.5: Dot Products of the Input Pixels and the Filter in the Convolution layer (Yamashita et al., 2018)

The next layer is the pooling layer, a dimensionality reduction approach known as downsampling, which lowers the number of input components. The pooling approach is similar to the convolutional layer and applies a filter over the whole input, but this filter lacks of weights. Instead, the kernel populates the output array from the values in the receptive field using an aggregation function (Kumar, 2022). This can be shown in Figure 2.6 below. Despite the substantial information loss caused by the pooling layer, CNN benefits in several ways. They support less complexity, improved effectiveness, and the avoidance of overfitting.

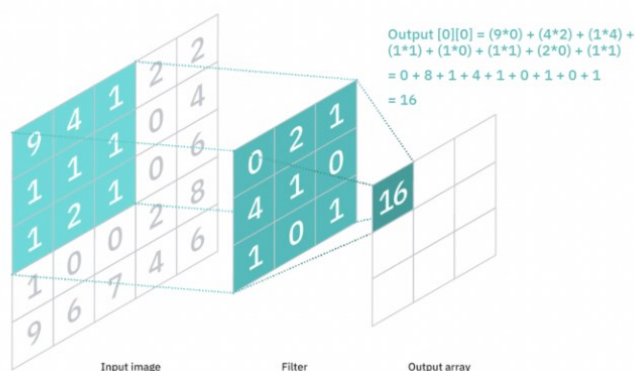


Figure 2.6: Downsampling in Pooling Layer (Kumar, 2022)

Based on the features that were gathered from previous layers and their respective filters, the classification process was carried out in the last layer. This last layer is called a fully-connected layer. In this layer, an activation function called softmax is often utilised to classify the inputs properly, yielding a probability ranging from 0 to 1 (IBM, 2023a).

Therefore, the whole framework of the CNNs can be illustrated in Figure 2.7 below.

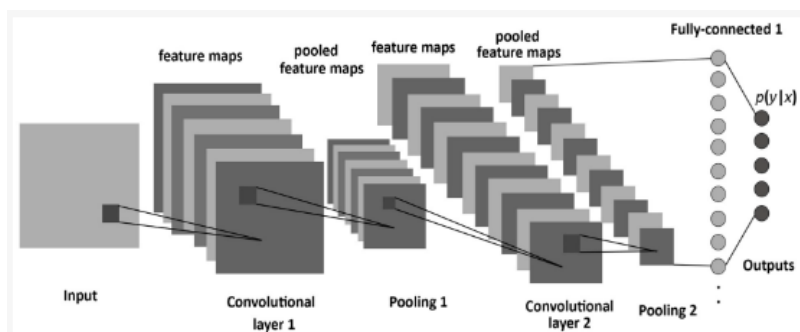


Figure 2.7: Framework of the CNNs (Albelwi and Mahmood, 2017)

Back to the subject of this section, YOLOv7 is chosen because of a few reasons. Firstly, in the range of FPS from 5 to 160, YOLOv7 works superior to any current object detectors in terms of both speed and accuracy on GPU V100, it has the highest accuracy of 56.8% average precision (AP) of all real-time object detectors with 30 FPS or more (Wang, Bochkovski and Liao, 2022). This can be proved by Figure 2.8 below which shows the comparison of the object detectors that were trained by using the popular dataset called MS COCO (Microsoft Common Objects in Context).

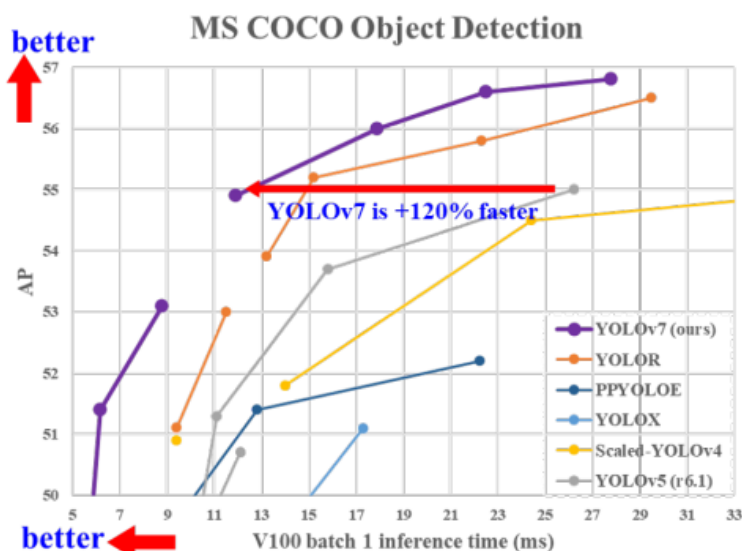


Figure 2.8: Comparison with Other Real-time Object Detectors(Wang, Bochkovski and Liao, 2022)

Figure 2.8 indicated that YOLOv7 is 120% faster than YOLOv5. Moreover, since the existing LPR system is utilizing YOLOv4-tiny and based on Table 2.1, the improvement for the AP^{val} from YOLOv4-tiny to YOLOv7-tiny is 10.7 %. This shows the great leap in development.

Table 2.1: Comparison of Baseline Object Detectors (Wang, Bochkovski and Liao, 2022)

Model	#Param.	FLOPs	Size	AP^{val}	AP_{50}^{val}	AP_{75}^{val}	AP_S^{val}	AP_M^{val}	AP_L^{val}
YOLOv4 [3]	64.4M	142.8G	640	49.7%	68.2%	54.3%	32.9%	54.8%	63.7%
YOLOv4-u5 (r6.1) [81]	46.5M	109.1G	640	50.2%	68.7%	54.6%	33.2%	55.5%	63.7%
YOLOv4-CSP [79]	52.9M	120.4G	640	50.3%	68.6%	54.9%	34.2%	55.6%	65.1%
YOLOv4-CSP [81]	52.9M	120.4G	640	50.8%	69.5%	55.3%	33.7%	56.0%	65.4%
YOLOv7	36.9M	104.7G	640	51.2%	69.7%	55.5%	35.2%	56.0%	66.7%
improvement	-43%	-15%	-	+0.4	+0.2	+0.2	+1.5	=	+1.3
YOLOv7-X [81]	96.9M	226.8G	640	52.7%	71.3%	57.4%	36.3%	57.5%	68.3%
YOLOv7-X	71.3M	189.9G	640	52.9%	71.1%	57.5%	36.9%	57.7%	68.6%
improvement	-36%	-19%	-	+0.2	-0.2	+0.1	+0.6	+0.2	+0.3
YOLOv4-tiny [79]	6.1	6.9	416	24.9%	42.1%	25.7%	8.7%	28.4%	39.2%
YOLOv7-tiny	6.2	5.8	416	35.2%	52.8%	37.3%	15.7%	38.0%	53.4%
improvement	+2%	-19%	-	+10.3	+10.7	+11.6	+7.0	+9.6	+14.2
YOLOv4-tiny-3l [79]	8.7	5.2	320	30.8%	47.3%	32.2%	10.9%	31.9%	51.5%
YOLOv7-tiny	6.2	3.5	320	30.8%	47.3%	32.2%	10.0%	31.9%	52.2%
improvement	-39%	-49%	-	=	=	=	-0.9	=	+0.7
YOLOv7-E6 [81]	115.8M	683.2G	1280	55.7%	73.2%	60.7%	40.1%	60.4%	69.2%
YOLOv7-E6	97.2M	515.2G	1280	55.9%	73.5%	61.1%	40.6%	60.3%	70.0%
improvement	-19%	-33%	-	+0.2	+0.3	+0.4	+0.5	-0.1	+0.8
YOLOv7-D6 [81]	151.7M	935.6G	1280	56.1%	73.9%	61.2%	42.4%	60.5%	69.9%
YOLOv7-D6	154.7M	806.8G	1280	56.3%	73.8%	61.4%	41.3%	60.6%	70.1%
YOLOv7-E6E	151.7M	843.2G	1280	56.8%	74.4%	62.1%	40.8%	62.1%	70.6%
improvement	=	-11%	-	+0.7	+0.5	+0.9	-1.6	+1.6	+0.7

In addition, one of the main improvements in YOLOv7 is the use of anchor boxes. To identify objects of various forms, anchor boxes are a collection of preconfigured boxes with various aspect ratios (Kundu, 2023). With nine anchor boxes, YOLO v7 can detect a wider variety of object shapes and sizes than earlier iterations, which helps to lessen the number of false positives.

Last but not least, it is worth noting that the YOLOv7 design is not built on the YOLOv6, instead it is based on Scaled YOLOv4, YOLOv4 and YOLO-R YOLO model architectures from beforehand. The computational block in the YOLOv7 backbone is named E- Extended Efficient Layer Aggregation Network (ELAN). This architecture lets the YOLOv7 learn better by employing "expand, shuffle, merge cardinality" to constantly increase the learning capability without obliterating the original gradient route (Boesch, 2023).

2.4 Cloud Computing

Cloud computing is a technique to carry out computing services over the internet to provide quicker and better innovation & creation, a wide variety of resources, and enable a competitive market. Services such as servers, networking, software, storage, databases, analytics and intelligence are provided (Azure, 2022b).

These mentioned services are able to benefit humankind from the perspectives of cost, time, efficiency, effectiveness, dependability, and reliability. Taking some of the top benefits concerned by the businessship, shifting to cloud computing definitely can reduce the cost as it eliminates the capital expense in purchasing the necessary hardware to set up and maintain on-site data centres. The hardware such as the cooling system and the racks to put the machines. According to TrackVia (2014), the increase in productivity when allowing employees to work remotely (one of the abilities in cloud computing) is 35-40% experienced by large companies such as British Telecom, Best Buy and Dow Chemicals. This also increases the reliability as the progress will be kept updated from time to time.

There are a few common types of cloud services offered: Software as a service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Aldahwan and Ramzan (2022) explained these 3 services in such a way that SaaS provide applications in the cloud environment to the consumers, but consumers do not have access to control the cloud networks, operation, and infrastructure. For example, Google's Gmail and Yahoo. In other words, SaaS can be considered a type of delivery model, which will help technologies that enable web services. Secondly, PaaS enables the applications to be used by consumers while deploying the applications to the cloud infrastructure.

However, only some control is given to the customer over the deployed applications, but not over the cloud infrastructure (networks, servers, and storage). Shim (2023) stated some of the examples of PaaS such as SAP Cloud, Microsoft Azure, and Heroku. Thirdly, IaaS allows the consumer to utilise the cloud infrastructure owned by the provider if requested. The highlighted point is that consumers are given access to deploy and run the software, but not in controlling the hardware cloud infrastructure, for example, Amazon Web Services (AWS).

Furthermore, these services can be deployed into 4 types of cloud environments: Public Cloud, Private Cloud, Hybrid Cloud and Community Cloud. Public Cloud as the name suggested, the cloud service is provided to the public through an open network. Private Cloud can be hosted by anyone, but more commonly it happens in a big organization to ensure privacy and control. Hence within a virtual environment, the actual computing resources are made used to provide computational power as a service. Hybrid Cloud is the combination of both private and public clouds, this enables the sharing of data and applications. The combination of private data and the public cloud is the typical type of Hybrid Cloud (Aldahwan and Ramzan, 2022). Lastly, Community Cloud is formed when an identical infrastructure is required by a number of organizations that agreed to share it. Since resources are shared, a high level of privacy and security is required, and policy compliance is strictly enforced, this causes the cost to be higher to do the maintenance.

2.5 Cloud Service Provider

After discussing the wonderfulness of cloud computing, it is time to look into the cloud service provider. In this section, the cloud services from Amazon Web Services, Microsoft, Intel and Huawei will be discussed.

2.5.1 Amazon Web Services

Amazon Web Services is an overview of all the products. The product discussed is Amazon SageMaker. With fully managed infrastructure, tools, and processes, it creates, trains, and deploys machine learning (ML) models for most of the use cases (Amazon Web Services, 2022). Furthermore, according to Kranz (2021), Amazon SageMaker is a managed service in the AWS public cloud and this platform automates the tedious laborious process of creating an AI pipeline that is ready for production.

This Amazon SageMaker is a good platform because the step-by-step guidance in building an ML model is provided which includes the code. Besides, if any problem is encountered, the AWS Community is ready to help. There are a lot of members sharing the solutions for hard-to-solve problems.

However, it is not free and the pricing system is quite confusing, but fortunately, the pricing calculator is provided. This allows interested people to

customize their services at the minimum cost. Yet, the functions in Amazon SageMaker, the virtual Central Processing Unit (vCPU), memory size, clock speed and et al. are needed to be chosen, indeed it is very detailed, but it is somehow confusing.

2.5.2 Microsoft

Microsoft provides a service called Microsoft Azure. The end-to-end machine learning lifecycle in this service uses an enterprise-grade service. At the same time, deep Visual Studio Code can be used to go from local to cloud training seamlessly, and auto-scale can be achieved using the powerful cloud-based CPU and GPU clusters (Azure, 2022a).

Microsoft Azure provides documentation and tutorials for the beginner. It is undeniable the interface is neat and tidy. Same as in Amazon SageMaker, the basic code is given to let the user for better understanding. After the code successfully runs at the edge terminal, the code can be submitted and run in the cloud. Furthermore, the monitoring also can be done in the studio.

Everything will have its price. The pricing calculator is not present in Microsoft Azure. This is because it is not very complicated to understand as it equips with the filter function and having tables to show the pricing rate. Hence, Microsoft Azure could be one of the considerations when comes to the selection of cloud services.

2.5.3 Huawei

Huawei also has a cloud service namely Huawei Cloud ModelArts. Huawei defined ModelArts as a one-stop development platform for AI developers. This is because data processing, development, training, management, and deployment of a model are done at just one stop. ModelArts assists AI developers in managing the AI development lifecycle by providing data preparation, semi-automated data labelling, large-scale distributed training, automatic modelling, and on-demand model deployment on devices, edges, and clouds (Huawei Cloud, 2023). Moreover, ModelArts has high performance as it is embedded with the self-developed MoXing deep learning framework that has the capability to accelerate the development of algorithms and training.

Huawei provides a well, systematic tutorial for a beginner to get started. However, the cost induced the issue. Before using ModelArts to deploy a model, a series of services is the prerequisite and not free of charge. This applied to the tutorial too. Hence, the tutorial can only be carried out theoretically.

2.5.4 Intel

Intel provides services called Intel® DevCloud. A cloud-based service called Intel® DevCloud for the Edge was designed to assist developers in creating and testing computer vision applications by utilizing the Intel® Distribution of OpenVINO™ toolkit (Intel, 2023k). A series of tutorials are provided for the developers to learn Python and C++ which are written in Jupyter Notebooks and have access to a number of example solutions that can be executed immediately from a web browser.

Hence, the benefit of using DevCloud is having faster access to Intel's development solutions, hardware, as well as software for the development of deep learning and computer vision applications. This is because only an internet connection and account are needed, it is very convenient. Another advantage is the accessibility to physically set up edge computers with the pre-installed Intel® Distribution of OpenVINO™ toolkit that is hosted in the cloud. This included the Central Processing Unit (CPU), integrated Graphics Processing Unit (iGPU), Field-Programmable Gate Array (FPGA), and Vision Processing Unit (VPU).

Figure 2.5 shows the overview of the user workflow for the Intel® DevCloud.

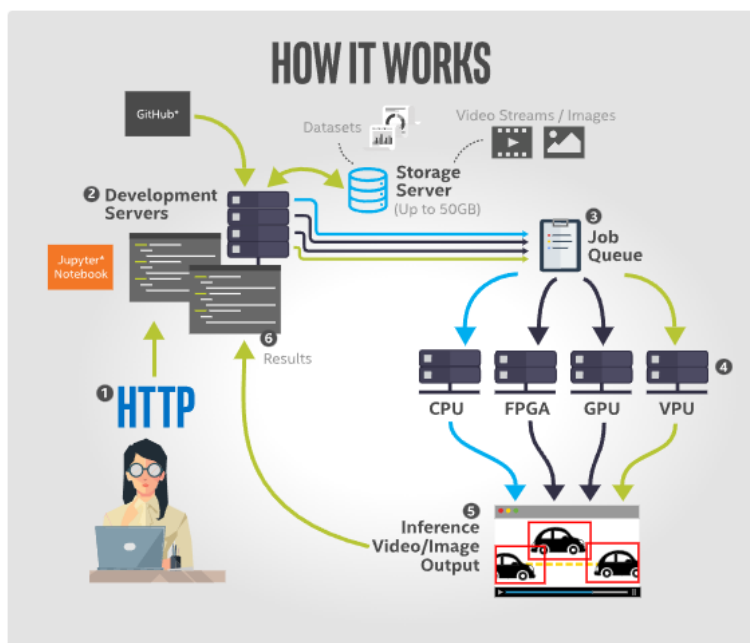


Figure 2.9: The overview of the user workflow for the Intel® DevCloud (Intel, n.d.)

It has a very clean flow. For illustration, the safety gear sample application is used to demonstrate. The user was able to edit the code in the Jupyter Notebook to learn and run the code step-by-step from the downloading of pre-trained models, followed by optimisation using OpenVINO™ and lastly implementing the model. Then, a path is set up for the video either pre-recorded or live. Next, the job is to queue and submit to carry out inference on a specific edge compute node using the available edge accelerators such as CPU, GPU, VPU, or FPGA. Lastly, the inference results are sent back and can be viewed in the Jupyter Notebook. Figure 2.6 shows the results of the Safety Gear Detection.

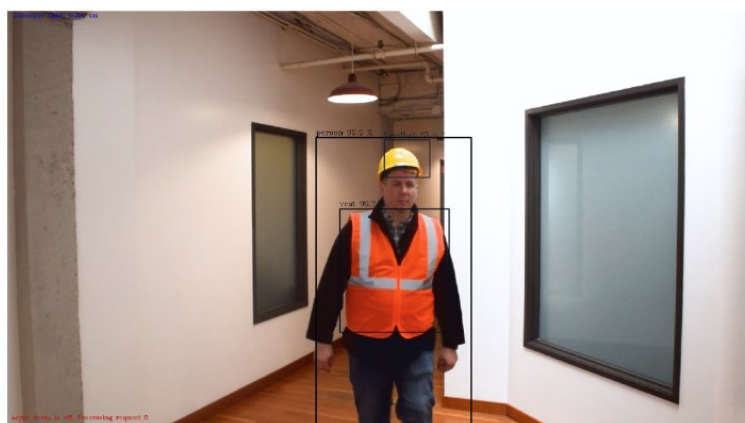


Figure 2.10: The results of the Safety Gear Detection. (Intel, n.d.)

Besides, Intel® DevCloud also provided a platform for us to create our own applications. And most importantly, it is free of charge. Though the Jupyter Notebook provided can only run 10 hours per day, it is more than enough. Therefore, Intel® DevCloud would be a very great choice for cloud computing.

2.6 Edge Computing

Edge computing is known as distributed Information Technology (IT) architecture in which the client data is handled as near as possible to the originating source at the periphery of the network (Bigelow, 2021). Though cloud computing is more and more utilized in the market, edge computing is still widely used and does not lose the competition against cloud computing. This is because cloud computing very depends on the internet speed, volume and bandwidth. If the number is connected, it will cause delays, or even congestion and disruption.

According to IBM (2023b), the above statement is also further supported, which stated that business applications are brought closer to data sources like Internet of Things (IoT) devices or local edge servers due to edge computing. As a result, strong commercial advantages can result from being close to the data's origin, including quicker insights, greater bandwidth availability and faster response times.

Since the most advantage is the less latency for edge computing, hence the most supporting technologies will be Real-Time Operating Systems (RTOS). The RTOS is normally applied to applications that required a high degree of reliability and precise timing. In the program that utilized RTOS, the task with higher priority will be switched to another if the resource is found occupied, and let the lower priority task with the resource run first so that the resource can be released to the higher priority task. This is called context switching. Thus, RTOSes are a huge potential advantage for edge computing when used with services that need defined response times (Caprolu et al., 2019).

For example, the pacemaker utilizes RTOS and is the edge of the network. Any computational activity that is started by the pacemaker itself is what we refer to as edge computing. The pacemaker can adjust the patient's heartbeat when it is out of rhythm and notifies the medical professional how

frequently this occurs (Fisher, 2020). Hence, the importance of edge computing can be seen.

2.6.1 Edge AI

As the name suggested, edge AI is the technology that is combining both edge computing and AI. This includes executing AI algorithms on local hardware equipped with edge computing capabilities. Edge AI enables users to process data on the device in real-time without requiring internet and system integration. This enables the AI model to be more responsive and effective in data processing, as well as increased security and privacy (Okeke, 2022).

Since edge AI perform the majority of data processing locally on an edge device. This meant that minimal information is consequently transferred to the cloud and other external sites. As a consequence, there is a decreased possibility of data being misused or treated improperly. This is one of the advantages of edge computing over cloud computing.

In edge AI systems, energy consumption could pose a major barrier to overhead communication. Therefore, for effective training and inference, lowering the number of communication rounds and the communication overhead per round is essential (Shi et al., 2020). Furthermore, compression, quantization, model partitioning, and sparsification are a few of the approaches that have been suggested as solutions to these problems since they may minimise the bulk of communicated data and lessen communication overhead.

In conjunction with a high number of overhead communications, privacy issues may rise as an increased chance for information leakage. It is also understood that the more data transmission, the higher the accuracy of the AI to carry out its task. Hence, a balance of data transmission rate, data loss, data accuracy, privacy and security issues shall be considered well.

Lastly, edge AI will face the challenge when dealing with computing resources (Singh and Gill, 2023). In comparison to central clouds, edge computing devices such as edge gateways and IoT sensors tend to have less processing power. Therefore, it might be difficult to run advanced AI models and techniques on such devices because they might require greater resources than the current ones.

2.7 Case Study

To have a better understanding of the title of the project, it is a crucial part to investigate the applications & carry out some case studies. This is to analyse the existing scenarios and how the cloud and AI systems are implemented to help the needs.

Every technology is invented to solve the problems. Hence, the case studies of molecular dynamics simulations and smart healthcare analysis and therapy for voice disorders will be discussed in this section.

2.7.1 Molecular Dynamics Simulations on Cloud Computing and Machine Learning Platforms

In Molecular Dynamics (MD) simulations, it is required a high-performance computing infrastructure such as a supercomputer to carry out the long-run simulations. The term "supercomputer" is frequently used to refer to the fastest high-performance systems available in the market today (Hosch, 2019). For instance, scientific and engineering work that required exceedingly high-speed computations such as in nuclear reactors, chemical compounding, and modelling world weather and climate.

This conventional simulation method to obtain the result is consuming a lot of time and energy, especially involving the supercomputer that required huge power consumption for just normal operation. Hence, in this paper, Sharma and Jadhao (2021) introduced Machine Learning (ML) to enhance MD simulation. By training and testing the ML models using a large collection of independent datasets, it is believed that the predictive power can be increased and reduced the computational cost. As stated in the paper, the relationship between the input parameter and outcome can be predicted by using artificial neural network-based regression models.

Besides using ML technology, Sharma and Jadhao (2021) proposed to use of a cloud computing platform to solve the large computing and storage requirements of simulations. The cloud is not only meant to be a storage, but also a computing platform. Unlike the local resources, the resources on the cloud are unlimited. Hence, there will always be sufficient ready to be used, the only and main concern is the monetary problem. Referring to the previous issue now gives rise to another important objective in cloud deployments which is to

optimize for cost with respective performance. According to Sharma and Jadhao (2021), compared to conventional cloud deployments, a 5 times reduction of cost is achieved by using SciSpot's cost-minimizing server selection and job scheduling policies.

Even though the uses of ML systems still have limitations, due to key advantages as stated in the paper: high-performance simulations, one-stop platform, large ecosystem, integration with data-driven approaches, good reproducibility and ease of sharing, ML technology has won a place to replace the conventional MD simulations. According to the results from Sharma and Jadhao (2021), the inference time is 10 000 times sped up which is from 30 minutes to 0.2 seconds using an ML surrogate.

2.7.2 Smart Healthcare Analysis and Therapy for Voice Disorder using Cloud and Edge Computing

Even though the cloud can also provide computing, in some cases edge computing is preferable as it is able to decrease feedback time, and diminished the cost of bandwidth and latency for operations. For example, in this article, therapy of voice disorder for initial transformations, the detection system was utilizing a deep learning approach while the specimen goes to edge computing. After that for a new transformation, edge computing transmits knowledge to the core cloud (Chandrasekhara Reddy, Sirisha and Reddy, 2018). It can be done through a cloud manager, a service provider which manages the administration and analysis.

Despite the integration of cloud computing and the Internet of Things (IoT), which enables multiple applications, this widespread transfer of all the aforementioned tasks to the cloud is ineffective in various schemes. Though now IoT has become popular, due to the large amount of data needed to be computed and analysed, it is not a wise way to upload all the data to the cloud via network. This takes a lot of time and most importantly it may cause network traffic jams. If this, unfortunately, happens in a medical centre during an emergency, especially requiring the patient information from another hospital/institute, a short few seconds could determine a patient lives or die.

In some cases, like in hospitals, it is better to use edge computing as the technology nowadays for cloud computing for such intensive data is not yet

mature. Chandrasekhara Reddy, Sirisha and Reddy (2018) recommended using edge computing (EC) for a smart healthcare framework. More specifically, a mechanism in the design for recognising and classifying voice disorders. In order to analyse and deal with complications, the expansion of speech-language pathologists (SLPs) in institute zones is necessary from time to time. According to the American Speech-Language-Hearing Association (2023), SLP is an expert that treats various types of communication and swallowing problems.

As mentioned beforehand, deep learning is used in edge computing. Hence, to achieve stable and real-time functionality of edge computing, SDN (Software-Defined Networking) is introduced. This SDN is an approach to networking that implements software-based controllers or application programming interfaces (APIs) to connect with the network's underlying hardware architecture and govern traffic (VMware, 2023). Unlike traditional networks which use dedicated hardware devices, SDN can create and control a virtual network (allows administrators to control the network). Therefore, SDN is much more flexible. With this SDN, Chandrasekhara Reddy et al. (2018) were able to demonstrate the edge computing administration system which consists of 3 categories: (1) Application Administration Framework, (2) EC applications platform administration design, and (3) EC hosting infrastructure administration system. With these, edge computing can reduce part of the transmission burden in the proposed system. To optimize the processing speed, the huge deep networks are stored in local caches.

In this case, cloud computing is not used for computing, but acts as a storage. The patients upload their tone specimens to the system. Then, the collected specimen is distributed to several cloud servers, and the cloud manager will send the specimens to the SLP to identify the causes/problems. The results will be sent to the laryngologists to open the prescriptions according to the problems of the patient. Then, the prescriptions will be sent back to the patient. To conclude the system in Chandrasekhara Reddy et al. (2018) paper, the system fully utilizes the functionality of cloud and edge computing. It facilitates the maintenance of real-time computing and networks. Furthermore, deep learning algorithms were also employed to examine patient information and deliver essential healthcare services.

2.8 Summary

In this chapter, the basic concept of AI is explained and one of the examples – the object detection model is demonstrated. The technique used is Mask R-CNN which was developed on top of the fundamentals of CNN. Next, after the model is successfully developed, it will be deployed to a cloud environment.

Cloud and edge computing are being compared. Both have their pros and cons. In the past, edge computing was preferable, as telecommunication technology was not yet mature, causing the over-the-air (OTA) transmission speed to be slow.

However, by the end of 2020, the 5G telecommunication technology will be more and more popular. The download rate and upload rate are 10 times faster than 4G. Hence, the migration from edge to cloud became a trend to move all the computing to the virtual environment.

The various cloud service providers are also being compared, even though Amazon Web Services, Microsoft Azure, and Huawei Cloud have very professional products and services, these are more suitable for business usage. Indeed, the deployment of AI models can still perform, but it is not convenient, as more cost is induced. Therefore, Intel® DevCloud will be used in this project.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

This project can be divided into eight major steps. Firstly, to understand and always refer to the aim of the project. This is to ensure that the direction is moving in the correct direction and is able to achieve the expected outcomes. Secondly, understand the variety of cloud services. This is to compare the pros and cons of cloud services so that the most suitable offer is chosen. Thirdly, installation of software. For example, VMware workstation to run Ubuntu in order to help me further understand the programming. Fourthly, the existing LPR system (including the license plate detection & recognition model) will be tested locally to ensure its workability. Fifthly, the LPR system is then migrated from the local to the cloud environment. Sixthly, after successfully deploying the LPR system to the cloud environment, the performance of the Intel hardware will be evaluated by inferencing the video using the LPR system. Seventhly, the YOLOv7 series will be trained and converted to IR format on the local device. This is to further improve the LPR system by replacing the current detection model that's utilizing YOLOv4-tiny. The YOLOv7 series included YOLOv7, YOLOv7-tiny and YOLOv7x. Lastly, the performance of the YOLOv7 series will be evaluated by utilizing cloud resources. This is to determine which one is the most suitable to use in real-time applications.

Steps 1 to 4 were done in the previous trimester, while the remaining steps were completed in this trimester of study. Figure 3.1 shows the workflow of the project.

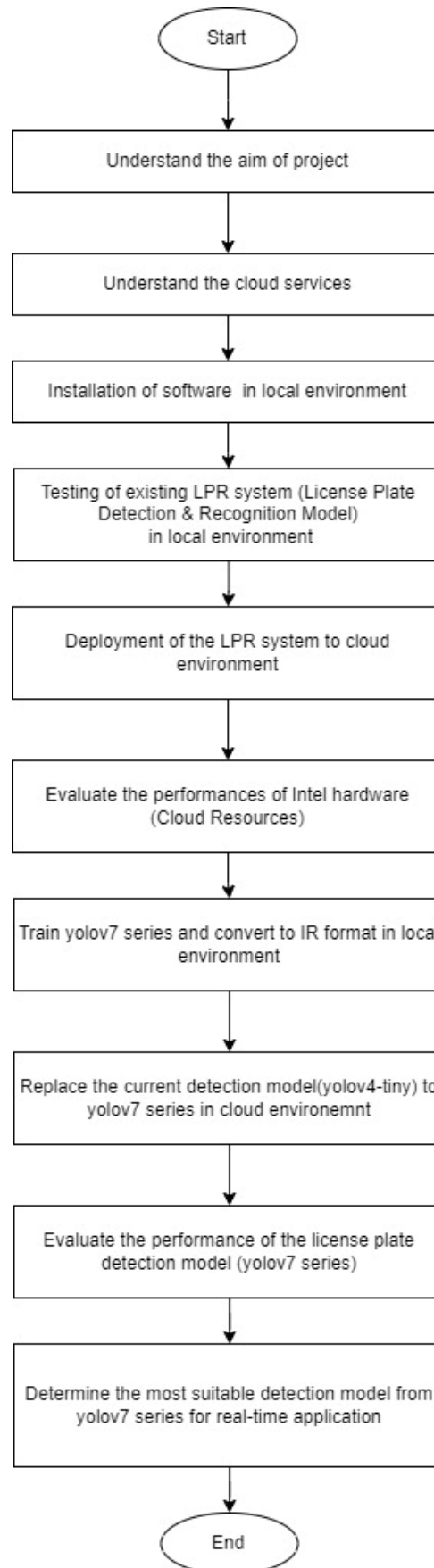


Figure 3.1: Project's Workflow

3.2 Methodology

The method to carry out this project will be discussed in this section. The hardware, software, and cloud resources will be described for a better overview.

3.2.1 Hardware Overview

The hardware devices used in this project were Intel NUC BXNUC10i7FNH3 with 64 GB RAM and 2 TB memory, and Sonnet eGPU Breakaway Box 750 - External GPU Chassis. Both of them are used together to form a powerful local computing device.

3.2.1.1 Intel NUC BXNUC10i7FNH3 64 GB RAM 2 TB

The Intel NUC BXNUC10i7FNH3 is a mini personal computer (PC) that features an Intel® Core i7-10710U processor, which has 6 cores and 12 threads (Intel, 2023e). The CPU also has integrated Intel® UHD Graphics, making it suitable for demanding applications such as video editing, 3D modelling, and machine learning development. Since it is a compact desktop computer, this gives the advantage of less space consumption and is highly portable.

The processor was built using 14 nm technology and has a base clock speed of 1.10 GHz, with a max turbo clock speed of 4.70 GHz. This NUC has a thermal design power (TDP) of 25 W. This TDP is a parameter that indicates the average power wasted by the CPU while it is working at the base clock with all cores engaged and under load. A TDP of 25 W is minimal in comparison to most PCs or laptops.

Regarding the RAM and storage capacity, the NUC supports up to 64 GB of DDR4 memory and has room for both M.2 form-factored SSD and HDD for internal storage (2TB), which can be configured in a Redundant Array of Independent Disks (RAID array) for improved performance and data redundancy. RAID is a well-known storage technology that combines multiple disk drive elements into a single unit.

Overall, the NUC is a powerful and versatile desktop computer that is ideal for users who need a lot of processing power and storage capacity in a compact form factor. Figure 3.2 below shows the Intel NUC BXNUC10i7FNH3.



Figure 3.2: Intel NUC BXNUC10i7FNH3 (MLACOM, 2023)

3.2.1.2 Sonnet eGPU Breakaway Box 750 - External GPU Chassis

The Sonnet eGPU Breakaway Box 750 is an external GPU chassis that uses Thunderbolt 3 to connect a desktop-class graphics card to your laptop or desktop computer. This is especially helpful for those who want greater graphics processing capability for demanding tasks.

The Breakaway Box 750 accepts full-length, full-height, and double-width Peripheral Component Interconnect express (PCIe) cards, including the most recent NVIDIA and AMD GPUs, which can be installed or updated without the need for any tools (Sonnet, 2023). The GPU NVIDIA GeForce GTX 1080 Ti, 11178.5MB was installed in this scenario.

The chassis also has a 750W power supply, which is more than enough to run even the most power-hungry graphics cards. Along with its performance capabilities, the Breakaway Box 750 has a quiet, temperature-controlled fan that keeps the GPU cool under high demand. It also boasts a sleek, brushed aluminium design that would look great with any workstation or desktop arrangement.

Overall, the Sonnet eGPU Breakaway Box 750 is an excellent choice for consumers who want more graphics processing power than that provided by their laptop or desktop computer without having to change the entire system. Figure 3.3 below shows the Sonnet eGPU Breakaway Box 750.



Figure 3.3: Sonnet eGPU Breakaway Box 750 (Sonnet, 2023)

3.2.2 Software Overview

In this project, the chosen software is Ubuntu OS, Python3, OpenCV, Jupyter Notebook, Google Colaboratory, Anaconda, PyTorch, Intel® Distribution of OpenVINO™ Toolkit, and MyBinder. These software are used for development. However, for research & testing the workability of each section, the VMware workstation was used to host a virtual OS – Ubuntu OS on a Windows device.

3.2.2.1 VMware Workstation

A virtual machine, a VMware workstation with the version of 16.2.4 is installed to run the operating system (OS) of Ubuntu with the version of 18.04.6. VMware is considered a powerful virtual machine for study purposes. This is because of VMware's Distributed Resource Scheduler (DRS) and VMware's Distributed Power Management (DPM). DRS is to control how physical resources are distributed among a group of virtual machines running on a group of hosts and maps VMs to hosts. Then performance improvement can be made by performing intelligent load balancing. DPM enhances DRS by enabling power consumption reduction through the consolidation of VMs onto fewer hosts (Gulati et al., 2012). Figure 3.4 below shows the VMware workstation logo.



Figure 3.4: VMware Workstation Logo (Serea, 2022)

3.2.2.2 Ubuntu OS

As mentioned in the previous section, the Ubuntu OS was used in this project instead of Windows. This is because Ubuntu OS has a simpler graphical user interface (GUI) as compared to Windows OS. This simpler GUI means having more resources allocated to perform the computing task. It is a user-friendly environment, especially for developers as various tools and libraries related to AI, ML and DL are available. For example, open-source libraries such as TensorFlow, PyTorch and OpenCV are also supported by Ubuntu OS. Figure 3.5 below shows the Ubuntu logo.



Figure 3.5: Ubuntu Logo (Freebie Supply, 2023)

3.2.2.3 Python

The programming language used here is Python. The reason is that it is intelligent enough to support AI, ML and DL applications. Unlike in C or C++, we need to manually declare variables to allocate addresses to be stored in the memory, but in Python, this process is not required. Hence, it shortens the development process and speeds up the computational time. Furthermore, Python is easy to use and understand the syntax of its simplicity and readily available modules. Another reason for utilizing Python is it is supported by Intel® Developer Cloud for the Edge that's using Jupyter Notebook. Therefore, Python is preferred over other languages. Figure 3.6 below shows the Python logo.



Figure 3.6: Python Logo (Python Software Foundation, 2023)

3.2.2.4 OpenCV

The full name for OpenCV is Open Source Computer Vision. It is a popular open-source computer vision and machine learning software library. For a variety of computer vision-related applications, such as processing images and videos, object identification and recognition, facial recognition, and motion tracking, OpenCV offers a large range of functions and algorithms. OpenCV focuses mostly on real-time vision applications as it utilises MMX and SSE instructions where they are available (OpenCV, 2023). In general, MMX (Multi-Media Extensions) and SSE (Streaming SIMD Extensions) are the tools that are used by modern CPUs to perform complex arithmetic and logical operations more efficiently. Furthermore, it is supported by Python as well as needed to conduct detection and visualisation on video frames, OpenCV is one of the irreplaceable components of this project. Figure 3.7 below shows the OpenCV logo.



Figure 3.7: OpenCV Logo (OpenCV, 2021)

3.2.2.5 Jupyter Notebook

With the aid of the open-source and free online application Jupyter Notebook, users could develop and share documents which include text, equations, graphics, and live code. It is frequently used in data science and scientific computing, but it may also be used for many other things, like research and teaching. Jupyter Notebook utilizes a web-based interface and is built upon the IPython project. It provides users with the ability to create documents, or notebooks, which can include a variety of content such as code, text, and images. Users can interactively modify and execute code within the notebook, and view the results in real-time. Figure 3.8 below shows the Jupyter Notebook logo.



Figure 3.8: Jupyter Notebook Logo(Lockwood. John, 2023)

3.2.2.6 Google Colaboratory

Google Colaboratory is also known as Google Colab. It is a free and powerful cloud-based platform that executes Python code through the web browser. Figure 3.9 below shows the Google Colaboratory logo.



Figure 3.9: Google Colaboratory Logo(Google, 2023a)

It has provided access to GPU such as Tesla T4 16 GB, computing tools that are desirable for the development of machine learning. Users just need to register an account to use the features in Google Colab. Importantly, Google Colab supports a number of well-known machine-learning libraries is another noteworthy element. These libraries consist of PyTorch, OpenCV, TensorFlow, and Keras. However, there are limitations on the usage of Google Colab's computing resources. The free Google Colab account limits the usage of its Google GPUs to 12 hours a day. In this project, 12 hours limitation is tolerable. This platform is only used to test the functionality of the training code for the YOLOv7 model. The input of the dataset is only 180 images and epochs of 100. The training only consumed around 4 hours. After confirmation of the functionality of the training code, the real training will be done for YOLOv7, YOLOv7-tiny and YOLOv7x models in the local environment using the hardware stated in section 3.2.1.1 with larger dataset and epochs.

3.2.2.7 Anaconda

Anaconda is an open-source Python programming language distribution that was developed for use in scientific computing and data science. Anaconda can be used on Windows, macOS, and Linux and offers users a platform to conveniently manage and install packages and dependencies for their data science projects, including Jupyter Notebook, NumPy, pandas, and Matplotlib.

Additionally, Anaconda has a package manager known as conda, which enables users to create virtual environments for their projects and install and manage packages within these environments. Noted that this feature will be used for the training of the AI detection models in the Ubuntu OS is which based on Linux. Figure 3.10 below shows the Anaconda logo.



Figure 3.10: Anaconda Logo(PNGEgg, 2023)

3.2.2.8 PyTorch

In this project, the chosen AI detection model is based on the PyTorch framework. Even though from the data in section 2.3, the usage of TensorFlow is more than PyTorch, it is not as accurate as the survey done among professional ML developers. Based on Wadawadagi (2023), he stated that TensorFlow excels in deploying models in production to produce commercial AI products, whereas PyTorch excels at research activities.

PyTorch is a machine learning framework that is open-source and highly utilized for deep neural network development and training. It is built on top of Python, enabling its effortless usage and integration with other Python-based tools. Additionally, PyTorch offers tools for effective model optimization, training and also provides pre-trained models and libraries to expedite development.

Furthermore, for research, repeatability is a crucial aspect. In Pytorch, deep learning repeatability is achieved by fixing all randomizations when training the AI models. Examples of constant variables are weight initialization and GPU-related randomization for classification and segmentation tasks (Alahmari et al., 2020). As a result, PyTorch is chosen as it is one of the most commonly used frameworks in deep learning and is prevalent in computer vision, and reinforcement learning applications. Figure 3.11 below shows the PyTorch logo.



Figure 3.11: PyTorch Logo (Lahon, 2020)

3.2.2.9 Intel® Distribution of OpenVINO™ Toolkit

The Intel® Distribution of OpenVINO™ Toolkit, which represents “Open Visual Inference and Neural Network Optimisation”. This is a powerful toolkit that is offered by Intel to the user for developing and deploying computer vision and deep learning applications. The toolkit also contains a model optimizer, which can transform models learned using various frameworks into an optimised Intermediate Representation (IR) format that can later be deployed on Intel hardware. The Intel hardware such as CPU, GPU, VPU, and FPGA are all compatible. The basic working flow of the Intel® Distribution of OpenVINO™ Toolkit is shown in Figure 3.12.

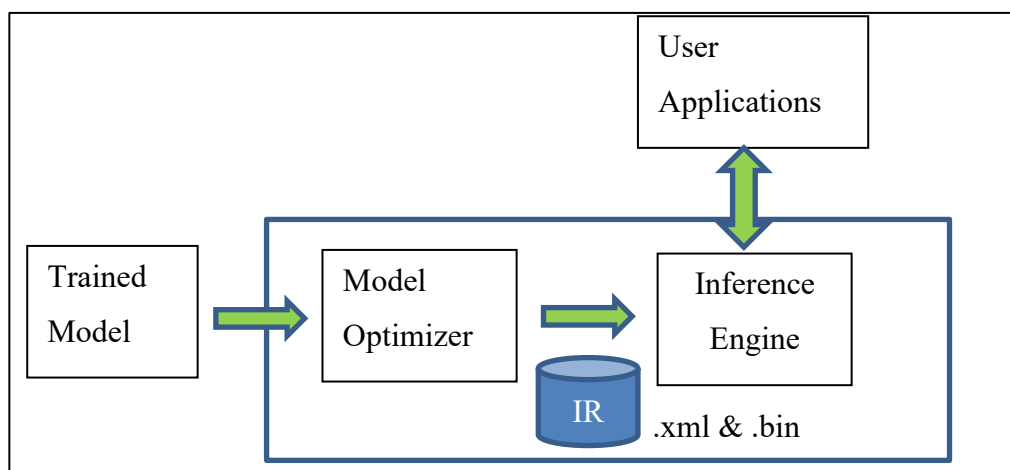


Figure 3.12: Intel® Distribution of OpenVINO™ Toolkit Basic Working Flow

In this project, the YOLOv7, YOLOv7-tiny and YOLOv7x trained model's weights file (in PyTorch extension) were exported as Open Neural Network Exchange (ONNX) extension format by using the python script inside the YOLOv7 repository. Later, the trained model in ONNX format was inputted into this toolkit to perform This toolkit is then converted and optimised for the trained model in ONNX format to IR format. The IR files then send to the inference engine in the cloud environment for model execution. Figure 3.13 below shows the Intel® Distribution of OpenVINO™ Toolkit logo.



Figure 3.13: Intel® Distribution of OpenVINO™ Toolkit Logo(logowik,2023)

3.2.2.10 MyBinder

MyBinder is an open-source and cost-free service that enables users to generate and share interactive computing environments based on Jupyter notebooks. It allows sharing of Jupyter notebooks, data, and code with others without the need for them to install any software on their local machine.

Upon initiating a MyBinder environment, it sets up a Docker container on a remote server that contains all the essential software dependencies including Python and any other necessary packages or libraries (MyBinder, 2023). Users can then upload their Jupyter notebooks and data files to this environment, which others can access through a web browser.

MyBinder is particularly helpful for creating reproducible research, collaborative coding, and teaching environments that enable students or colleagues to access a pre-configured computing environment without having to install any software on their computer.

In this project, the conversion of the trained model to IR format is done using this platform as it also provides the repository that supported Intel® Distribution of OpenVINO™ Toolkit. This reduced the redundancy to install the toolkit into the local environment. Figure 3.14 below shows the MyBinder logo.



Figure 3.14: MyBinder Logo (MyBinder, 2023)

3.2.3 Intel® Developer Cloud for the Edge Hardware Specifications (Cloud Resources Overview)

Among all the Intel hardware in the Intel® Developer Cloud for the Edge, the common hardware used to develop software are:

- Intel® Xeon® Gold 6258R Processor
- Intel® Core™ i7-1065G7 Processor
- Intel® Xeon® Gold 6338N Processor
- Intel® Core™ i7-1185G7E Processor (with integrated GPU Intel® Iris® Xe Graphics)
- Intel Atom® x6425RE Processor (with integrated GPU Intel HD Graphics 530).

In this part, the above 5 Intel hardware will be discussed.

3.2.3.1 Intel® Xeon® Gold 6258R Processor

The Intel® Xeon® Gold 6258R processor is a powerful server-grade CPU based on the 2nd generation Intel® Xeon® Scalable processor, formerly code-named “Cascade Lake”. It was introduced in the first quarter of 2020 and is manufactured using 14nm semiconductor technology (Intel, 2023i).

Featuring 28 physical cores and 56 threads, the Intel® Xeon® Gold 6258R processor delivers a base clock speed of 2.7 GHz and a maximum turbo clock speed of 4.0 GHz. With support for up to 1 TB of DDR4 memory and a TDP of 205 W, this processor offers an optimal balance of computing power and memory capacity for demanding workloads in enterprise-level servers, workstations, and data centre applications.

Moreover, the configuration of RAM of 96 GB, makes it well-suited for managing data-intensive tasks and handling large workloads.

The Intel® Xeon® Gold 6258R processor is designed to meet the high-performance computing demands of various applications, including virtualization, data analytics, and artificial intelligence. Its advanced features include Intel® Turbo Boost Technology 2.0, Intel® Hyper-Threading Technology, and Intel® AVX-512 instruction set extensions, providing high-performance computing capabilities to handle the most demanding workloads. For example, Intel® Turbo Boost Technology 2.0, utilises thermal and power

headroom, which dynamically boosts the processor's frequency as necessary to provide you with a speed boost when you need it and greater energy efficiency when you don't (Intel, 2023h).

In summary, the Intel® Xeon® Gold 6258R processor is a reliable and robust CPU, ideal for managing data-intensive tasks and handling large workloads in a variety of data centre applications, providing a stable and efficient computing environment.

3.2.3.2 Intel® Core™ i7-1065G7 Processor

The Intel® Core™ i7-1065G7 processor is a powerful CPU built for usage in laptops and other mobile devices. It is part of the 10th generation Intel® Core™ i7 processor family, formerly code-named “Ice Lake”, and was first introduced in the third quarter of 2019. It is built on the 10nm manufacturing process (Intel, 2023b), which offers improved energy efficiency and performance compared to previous generations.

This processor has 4 physical cores and 8 threads, a base clock speed of 1.3 GHz, and a maximum turbo frequency of 3.9 GHz. It supports up to 64 GB of LPDDR4-3733 memory and has a TDP of 15 W. This also includes 16 GB of RAM, which is well-suited for running multiple applications simultaneously and handling large workloads.

It also includes Intel® Turbo Boost Technology 2.0, Intel® Hyper-Threading Technology and integrated GPU Intel® Iris® Plus Graphics G7, providing advanced graphics capabilities for demanding applications. For instance, Intel® Hyper-Threading Technology provides each physical core with two processing threads (Intel, 2023d). Applications with a high number of threads may do more processes in parallel, completing assignments faster.

Overall, the Intel® Core™ i7-1065G7 processor is a powerful and reliable CPU that is well-suited for mobile computing applications. The inclusion of 16 GB of RAM and integrated Intel® Iris® Plus Graphics G7 make it even more capable of handling large workloads and managing demanding tasks. Ultimately, providing a high-performance computing experience for demanding tasks such as video editing, gaming, and other intensive applications.

3.2.3.3 Intel® Xeon® Gold 6338N Processor

The Intel® Xeon® Gold 6338N processor is a server-grade CPU designed for use in data centre and enterprise environments. It was released in the first quarter of 2021, it is part of the 3rd generation Intel® Xeon® scalable processor family, and it has formerly code-named “Ice Lake”. It is based on the 10nm technology and has a total of 32 cores and a total of 64 threads with a base clock speed of 2.2 GHz and a turbo boost frequency of up to 3.5 GHz (Intel, 2023j).

It supports up to 6TB of DDR4 memory, with a maximum memory speed of 2667MHz, and has a TDP of 185 W. Noticed that this processor came with the system configuration of RAM of 128 GB. This amount of RAM could be useful for running memory-intensive applications or virtual machines, among other use cases.

The Intel® Xeon® Gold 6338N processor also features the Intel® Advanced Vector Extensions 512 (Intel® AVX-512) instruction set, which is designed to accelerate complex calculations and workloads. According to (Intel, 2023a), applications can fit thirty-two double-precision and sixty-four single-precision floating point operations, as well as eight 64-bit and sixteen 32-bit integers, into the 512-bit vectors on each clock cycle. In other words, it is able to enhance performance when the same operations are performed on multiple data objects.

It also supports hardware-level security features. For example, Intel® Software Guard Extensions (SGX) can help protect sensitive data and workloads. This is done by generating hardware-enforced trusted execution protection (Intel, 2023f).

Overall, the Intel® Xeon® Gold 6338N processor offers high levels of processing power and is suitable for use in applications such as AI, machine learning, and high-performance computing.

3.2.3.4 Intel® Core™ i7-1185G7E Processor with integrated GPU Intel® Iris® Xe Graphics

The Intel® Core™ i7-1185G7E processor is an outstanding performance CPU designed for embedded systems such as laptops and mobile devices. It belongs to the 11th generation Intel® Core™ i7 processor family, formerly code-named "Tiger Lake". This processor has a base clock speed of 1.8 GHz and can boost up to 4.8 GHz (Intel, 2023c). The highlight of this processor is that it can configure the base frequency depending on the usage. By varying the base frequency to lower (1.2 GHz) or higher (2.8 GHz), the TDP also changed to 12 watts or 28 watts respectively from the 15 W.

Additionally, this processor can support up to 64 GB memory with a RAM of 16 GB. It has 4 cores and 8 threads, which allows it to handle multiple tasks at once with ease.

The Intel® Core™ i7-1185G7E processor also comes with Intel's latest integrated graphics processor, the Intel® Iris® Xe Graphics, which is capable of delivering high-quality graphics performance for gaming, video editing, and other GPU-intensive tasks. This processor is also designed with Intel's latest technologies, such as Thunderbolt 4, and Intel® Hyper-Threading Technology, which provide faster data transfer speeds, improved connectivity, and enhanced storage performance.

Overall, the Intel® Core™ i7-1185G7E processor is a powerful and efficient processor that delivers excellent performance and is suitable for demanding applications such as gaming, content creation, and scientific computing.

3.2.3.5 Intel Atom® x6425RE Processor with integrated GPU Intel® HD Graphics 530

The Intel Atom® x6425RE processor is a server-class processor designed for high-performance computing applications in data centres and edge computing environments. It is formerly code-named “Elkhart Lake” and one of the Intel Atom® Processor X Series, which is built on a 10nm technology.

The Intel Atom® x6425RE processor features 4 cores and 4 threads with a base clock speed of 1.9 GHz. It has a TDP of 12 W and supports up to 128 GB of DDR4 memory with RAM of 16 GB.

Additionally, unlike other processors, it does not feature Intel's Hyper-Threading technology. Instead, it has the advanced feature of Intel® Time Coordinated Computing (Intel® TCC). This is mostly used in real-time applications especially those for latency sensitive. Intel® TCC enabled processors provide the best compute and time performance to reduce jitter by helping to maximize efficiency. This can be achieved by combining time-sensitive and non-time-constrained applications onto an individual board(Intel, 2023g).

Furthermore, one of the attractive features is this integrated GPU Intel® HD Graphics 530 able to support 4K resolution at 60 Hz.

The Intel Atom® x6425RE processor is designed for use in servers and embedded systems that require high performance and low power consumption. It is particularly well-suited for use in edge computing applications, such as industrial automation, IoT gateways, and networking appliances, where low power consumption, high performance, and reliable operation are critical.

3.2.3.6 Price of each Intel Hardware

Table 3.1 below shows the comparison between Intel hardware prices.

Table 3.1: Intel Hardware Price Comparison

Intel® Xeon® Gold 6258R Processor	Intel® Core™ i7- 1065G7 Processor	Intel® Xeon® Gold 6338N Processor	Intel® Core™ i7- 1185G7E Processor (with integrated GPU Intel® Iris® Xe Graphics)	Intel Atom® x6425RE Processor (with integrated GPU Intel HD Graphics 530).
\$4523.00	\$469.00	\$3200.00	\$474.00	\$71.00
RM 20186.15	RM 2093.15	RM 14281.60	RM 2115.46	RM 316.87

Noted: The United States Dollar to Malaysian Ringgit is based on 27 April 2023, UTC-10.01 am.

3.3 Work Plan

The work plan consists of the deployment of the existing LPR system to the cloud environment, and the development of a new detection model to improve the LPR system. Moreover, the timeline and milestones will also be discussed in this section.

3.3.1 Workflow

3.3.1.1 Deployment of Existing LPR system to Cloud Environment

The existing LPR system is deployed to the cloud environment is done by the following steps: file uploading, python code script modification, job submission to queue, and lastly plotting the graph for later performance evaluation.

3.3.1.1.1 File Upload

The file upload to the cloud environment is a necessary step before beginning to develop the LPR system. The files included Python script, detection & recognition models in IR format, label text files and etc. This is because the cloud environment does not have the data or files created previously. File uploading is also a time-consuming process as it really depends on the internet speed. For better illustration, Table 3.2 was tabulated to show the upload speed for each size of the file.

Table 3.2: Uploading Speed for Each Size of File

Size (MB)	1 st Upload	2 nd Upload	3 rd Upload	Average
1	1 s	1 s	2 s	1.33 s
4.5	16 s	2 min 02 s	1 min 10 s	1 min 09 s
10	1 min 44 s	49 s	55 s	1 min 09 s
11.5	49 s	1 min 43 s	2 min 01 s	1 min 31 s
11.9	1 min 16 s	3 min 51 s	3 min 47 s	2 min 58 s
139	21 min 41 s	20 min 14 s	23 min 22 s	21 min 46 s
339	48 min 44 s	50 min 14 s	35 min 04 s	44 min 41 s

There are a few possible reasons for the fluctuation in the uploading time. Firstly, the high usage of the CPU on the local device causes slower computation in sending files up to the cloud. Secondly, it might be due to the time zone difference between Malaysia and Intel® Developer Cloud for the Edge service that is based on the Pacific Time Zone (PT). The time difference is 12 hours, this means that network traffic might be different. In other words, Intel® Developer Cloud for the Edge may reduce the network traffic during PT night time, resulting in a slower connection to the server during working hours (morning) in Malaysia.

3.3.1.1.2 Modify Python Script

Following by the modification of the Python script. In the local environment, there is no parameters or arguments parsed into the script for execution, all variable is set in the script. However, in dealing with a cloud environment, it is not wise to do so, as the cloud environment has its own way to declare the directory or path. Especially when need to submit the job to the queue by sending a shell script (.sh) to a different node (Intel hardware). The shell script is mainly to declare the required parameters (including the input video, detection & recognition models), followed by activating the virtual environment & OpenVINO variables, and lastly running the Python script to perform the inferencing.

Furthermore, some of the packages only exist in Intel® Developer Cloud for the Edge. For example, the applicationMetricWriter function displays the progress when the inferencing on the Intel hardware is running as shown in Figure 3.15 below.

```
[5]: # Submit job to the queue
job_id_xeon_cascade_lake = !qsub tinyyolov4.sh -l nodes=1:ldc018 -F "./results/xeon_cascade_lake/ CPU FP16 {InputVideo} 0.4 {NumRequests_CPU} 1" \
-N tinyyolov4_cascade -e tmp/ -o tmp/ -v VENV_PATH,OPENVINO_RUNTIME
print(job_id_xeon_cascade_lake[0])
id_xeon_cascade_lake = job_id_xeon_cascade_lake[0].split('.')[0]
# Progress Indicators
if job_id_xeon_cascade_lake:
    progressIndicator("./results/xeon_cascade_lake", 'i_progress'+id_xeon_cascade_lake+'.txt', "Inference", 0, 100)

852434.v-qsrvr-1.devcloud-edge
```




Figure 3.15: Progress Bar for LPR Inferencing

3.3.1.1.3 Job Submission

After settling the code in the Python script, generate the code for different nodes by this line of instruction as shown in Figure 3.16 below.

```
[1]: |pbsnodes | grep compnode | awk '{print $3}' | sort | uniq -c
```

```

6 idc001sk1,compnode,openvino-latest,intel-core,i5-6500te,intel-hd-530,ram8gb
6 idc002mx8,compnode,openvino-latest,intel-core,i5-6500te,intel-hd-530,ram8gb,myriadx-8-vpu
5 idc004nc2,compnode,openvino-latest,intel-core,i5-6500te,intel-hd-530,ram8gb,myriadx-1-vpu
1 idc006kbl,compnode,openvino-latest,intel-core,i5-7500t,intel-hd-630,ram8gb
2 idc007xv5,compnode,openvino-latest,intel-xeon,e3-12681-v5,intel-hd-p530,ram32gb
2 idc008u2g,compnode,openvino-latest,intel-atom,e3950,intel-hd-505,ram4gb,myriadx-1-vpu
1 idc009jkl,compnode,openvino-latest,intel-core,i5-7500,intel-hd-630,ram8gb
1 idc010jal,compnode,openvino-latest,intel-celeron,j3355,intel-hd-500,ram4gb
1 idc011arK2250s,compnode,openvino-latest,intel-core,i5-6442ee,intel-hd-530,ram8gb,myriadx-3-vpu
1 idc012ark12201,compnode,openvino-latest,intel-atom,e3940,intel-hd-500,ram8gb,myriadx-2-vpu
1 idc013ds580,compnode,openvino-latest,intel-atom,e3950,intel-hd-505,ram8gb
4 idc014,compnode,openvino-latest,intel-core,i7-8665ue,intel-uhd-620,ram16gb,myriadx-2-vpu
3 idc015ai5,compnode,openvino-latest,intel-core,i7-8665ue,intel-uhd-620,ram8gb
2 idc016ai7,compnode,openvino-latest,intel-core,i7-8665ue,intel-uhd-620,ram16gb
1 idc017,compnode,openvino-latest,intel-xeon,gold6258r,no-gpu,ram96gb
1 idc018,compnode,openvino-latest,intel-xeon,gold6258r,no-gpu,ram96gb,dstreamer
2 idc021,compnode,openvino-latest,intel-xeon,silver4214r,no-gpu,ram48gb
10 idc022,compnode,openvino-latest,intel-core,i7-10710u,intel-uhd-620,ram16gb
6 idc023,compnode,openvino-latest,intel-core,i5-8365ue,intel-uhd-620,ram8gb,myriadx-2-vpu

```

In the above output from executing the previous cell, the properties describe the node, and the number on the left is the number of available nodes of that architecture.

Figure 3.16: Generated Nodes' Codes

Then, choose the interested node to submit the job to. This submission can be done by running the segment of code in Figure 3.15. In Figure 3.15, the node submitted is “idc018”, indicating that the inference engine is Intel® Xeon® Gold 6258R Processor.

3.3.1.1.4 Display the Inferred Video

After the inferencing is done, the inferred video was displayed by executing this line of code as shown in Figure 3.17. This inferred video is viewed in the browser, the user also can download the inferred video to view it.

View results from an Intel® Xeon® Gold 6258R CPU

```
[6]: count = 0
while not (
os.path.exists(f"./results/xeon_cascade_lake/output_{id_xeon_cascade_lake}.webm")
and os.path.exists(f"./results/xeon_cascade_lake/stats_{id_xeon_cascade_lake}.txt")
): # Wait until the video and stats file is created.
time.sleep(1)
print(".", end="")
count = count + 1
# Wait 1 minute to check if results populates, or break from the infinite loop
if count >= 60:
raise Exception("Results did not populate")
videoHTML(
"Intel® Xeon® Gold 6258R CPU",
[f"./results/xeon_cascade_lake/output_{id_xeon_cascade_lake}.webm"],
f"./results/xeon_cascade_lake/stats_{id_xeon_cascade_lake}.txt",
)
```

[6]:

Intel® Xeon® Gold 6258R CPU

165 frames processed in 37.4 seconds

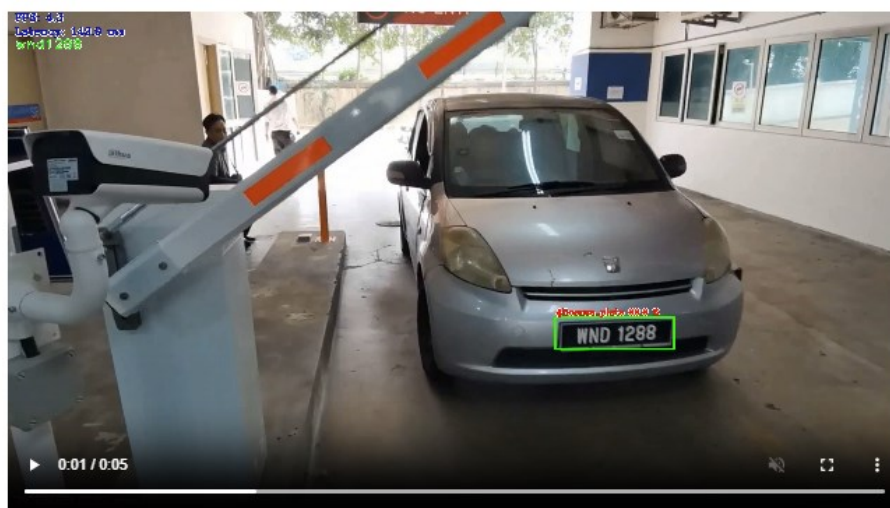


Figure 3.17: Display the Inferred Video

Until this stage, the deployment of the existing LPR system to the cloud environment is considered a success. It is time to improve the existing LPR system. This is done by replacing the current detection model from YOLOv4-tiny to the YOLOv7 series.

However, to suit the real-time application, the YOLOv7 series (YOLOv7, YOLOv7-tiny, YOLOv7x) have to be evaluated. Hence, another cycle in producing the AI objection model started with data preparation.

3.3.1.2 Data Preparation

The data is prepared by the following steps: data sourcing, data argumentation, data labelling & annotation, and data distribution to training: validation: test. Noted that all the data are images.

3.3.1.2.1 Data Sourcing

The data is collected from various sources such as websites (Kaggle, Roboflow, Google Images) and the accumulation from the database of the lecturer. A total of 1600 images were collected, including those already undergoing data augmentation.

3.3.1.2.2 Data Augmentation

Data augmentation is a technique commonly used in machine learning and computer vision, especially object detection training, to artificially enhance the size of a dataset by producing new training examples from the original pictures or data via various transformations. The transformation included:

- Image rotation
- Image scaling
- Image flipping
- Image translation
- Image cropping
- Changing the light contrast of the image
- Changing the resolution of the image

The goal of data augmentation is to diversify the variety of the training data, and hence improve the trained model's performance and generalisation capacity.

3.3.1.2.3 Data Labelling & Annotation

After collecting all the data, the data labelling process takes place. This process is also known as data annotation. In the context of object detection, data labelling included annotating images with bounding boxes or other forms of region-of-interest (ROI) annotations to identify the location and extent of items of interest within the images.

Data labelling is an important stage in supervised machine learning, including object detection, since it offers ground truth information used to train and evaluate machine learning models. The labelled data is used to train object

identification models, which enable the model to generate predictions on unseen data during inference by learning the link between input photos and the relevant object locations or classes.

Depending on the complexity and type of the data, data labelling can be done manually or with automated methods. Automated data labelling uses pre-defined rules or algorithms to automatically produce annotations based on particular criteria, such as pixel intensity or colour thresholds. In this project, to increase reliability and accuracy, the labelling is done by human annotators. So that each image is diligently examined and manually drawn bounding boxes tightly around the objects of interest. The labelling is done by using the “labelImg” data annotation tool. Figure 3.18 below shows the example of license plate labelling using labelImg.



Figure 3.18: Example of License Plate Labelling using labelImg

3.3.1.2.4 Data Distribution to Train:Val:Test

By the rule of thumb, the common practice is that the ratio for the train:val:test is either 8:1:1 or 7:1.5:1.5. The train set is used to train the object detection model. Hence, it should be large enough for the model to gain insight into the underlying data patterns and undergoes generalization. Next, the function of the validation set is to finely tune the model's hyperparameters, such as learning rate, batch size, and regularisation strength, as well as to perform model selection. Another important factor for having a validation set is to prevent overfitting, verifying and validating the model's performance. Moreover, the test set is used to evaluate the trained model's final performance and offer a neutral estimate of

its generalization ability. It should not be utilized for model selection or hyperparameter tuning and should be preserved independently of the training and validation sets.

In terms of testing performance, increasing the size of the training set from 30% to 80% may also improve testing performance. However, when the training size rose from 80% to 90%, the testing performance trended in the opposite direction (Nguyen et al., 2021).

In general, the size of the training set had a significant impact on the ML models' ability to predict. Therefore, the ratio for the train:val:test selected in this project is 8:1:1.

3.3.1.3 Training of the Detection Model

The training of the YOLOv7 series was first done on the Google Colab using the smaller scale of the dataset which is 160 images for a train set, 20 images for a validation set, and 20 images for a train set. This is due to the limitation of the Google Colab such as limited run time, and GPU resources as a free user.

After ensuring the workability of the trained models, the real training was done in the local environment using the Intel NUC BXNUC10i7FNH3 that's connected to Sonnet eGPU Breakaway Box 750, integrated with GPU NVIDIA GeForce GTX 1080 Ti. Before the training started, the virtual environment is created using the "conda" command. The virtual environment is just like an isolation compartment in the mini PC. All the installation of the dependencies and packages is done here, and would not affect the normal OS.

After all the setup is completed, the training starts. With the help of GPU, the training speed will be faster as compared to the training that's just using the CPU. In order to test the performances of the YOLOv7 series detection model, the hyperparameters are fixed for all models. For instance, the epoch is set to 300, and the batch size is set to 8.

The epoch is the number of iterations to pass the data into the network to perform weightage calculation and adjustment. Noticed that for all 3 YOLOv7 series detection models, the mean average precision (mAP) had started to converge when comes to the 250th epoch. 300 epochs are sufficient for training. The batch size is fixed to 8 is due to the limited memory allocation when training the YOLOv7x as it has the largest network size among all 3

models. Therefore, to be consistent, this value of batch size is selected. Subsequently, the training times are then recorded and tabulated in Table 3.3 as shown below.

Table 3.3: Training Time for YOLOv7, YOLOv7-tiny, and YOLOv7x

Model	YOLOv7	YOLOv7-tiny	YOLOv7x
Time to complete	6.713 hours	2.352 hours	9.826 hours

3.3.1.4 Conversion of the Detection Model to IR format

The first step of the conversion is exporting the trained YOLOv7 series detection models from the PyTorch extension (.pt file that consists of the weightage of the models) to the ONNX extension (.onnx). Instead of direct conversion to IR format, this action can ensure the interoperability of the detection models. ONNX is an open standard for describing machine learning models that allow for interoperability. It is guaranteed that a PyTorch model is compatible with other deep learning frameworks (TensorFlow, Keras, Caffe etc.) and can be deployed in a range of contexts by converting it to ONNX format (Jog, 2020). Figure 3.19 below shows the role of ONNX in bridging the development of ML models.

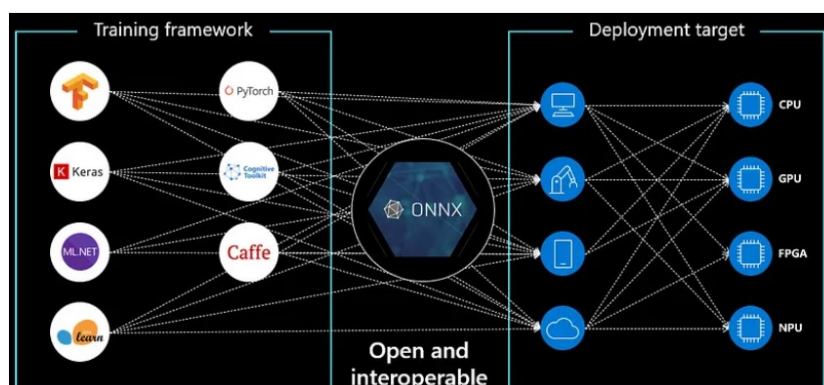


Figure 3.19: Role of ONNX in Bridging the Development of ML Models (Jog, 2020)

In other words, the detection models can be trained in any deep learning framework. The ONNX will function as a "middle-man" to make sure that the conversion from PyTorch format to IR format is successful.

Then, the detection that is in the ONNX extension will be optimized and converted to IR format, which consists of .xml and .bin files. This conversion is done using Intel® Distribution of OpenVINO™ Toolkit.

For the entire conversion from PyTorch to ONNX and lastly, to IR format, this is all done in the online platform – MyBinder. This reduced the burden to install all the packages such as onnxruntime, openvino and torch. Besides saving the memory space on the local device, the version of each package also can be ignored when utilizing MyBinder as it is intelligent enough to choose the compatible version.

In addition, this platform also can help to test the functionality of the detection model first before the model is used somewhere else.

3.3.1.5 Replace the Old Detection Model of the LPR system in Cloud Environment

This step is simple. The thing that needs to do is upload the 3 detection models (YOLOv7, YOLOv7-tiny, and YOLOv7x) that are in IR format to the cloud environment. Then change the old detection model's path to point to the new detection model's directory.

The remaining steps will be the same as described in section 3.3.1. However, an additional step is required to plot the graph to evaluate the performance of these 3 new detection models.

3.3.2 Programming Flow Chart

Figure 3.20 below shows the overall programming flowchart for the existing LPR system. This also applies to the improved version of the LPR system that utilised YOLOv7, YOLOv7-tiny and YOLOv7x detection models.

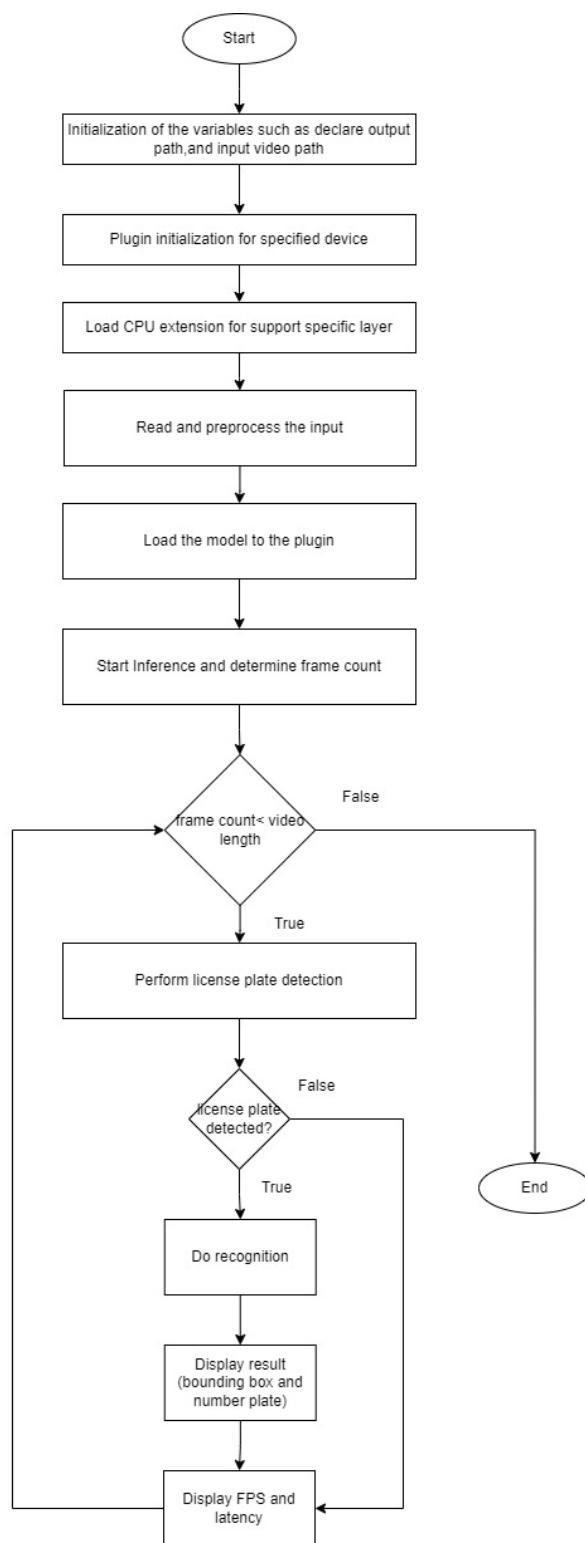


Figure 3.20: Overall Programming Flow Chart

3.3.3 Gantt Chart

Figure 3.21 shows the Gantt chart planned for the entire project. In the early stage of the project, it was started according to the plan. However, in the later stage, due to the limitation of the cloud environment, another alternative way is needed to solve the issues such as being incompatible with the package that is provided in the Intel® Developer Cloud for the Edge. Hence, the progress was dragged 2 weeks behind schedule. Nevertheless, the project managed to be completed in time.

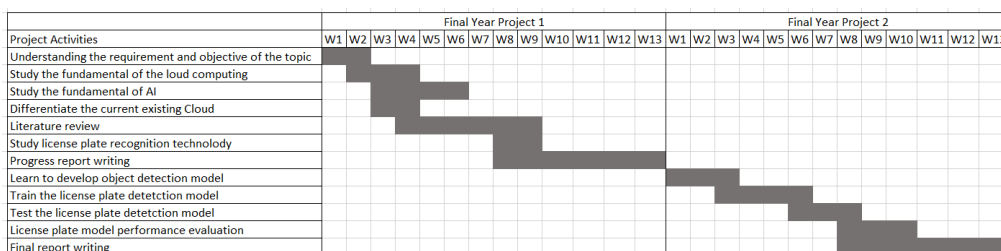


Figure 3.21: Gantt Chart for Entire Project

3.4 Work Budget

This project emphasizes the development of the LPR system in the cloud environment to cut the cost of the existing local setup. Hence, Table 3.4 shows the comparison of budgets between the local and semi-local (with cloud environment) setups. This is to highlight the cost induced for each phase of the project. This also shows the benefit of having Intel DevCloud.

Table 3.4: Comparison of Budgets between the Local and Semi-local Setups

	Full local setup (without cloud environment)	Semi-local setup (with cloud environment)
Coding	<ul style="list-style-type: none"> • Own laptop • Free of charges 	
Training of the models	<ul style="list-style-type: none"> • Borrow device from supervisor • Free of charges 	
Video capturing	<ul style="list-style-type: none"> • Real-time • Fast processing camera required • Cost induced 	<ul style="list-style-type: none"> • Upload the video file to the cloud • Any device with a camera will do • Free of charges
Video processing	<ul style="list-style-type: none"> • High computing power device required • Fix resource • Cost induced 	<ul style="list-style-type: none"> • Intel cloud resources • A variety of processors can be chosen including CPU, GPU, FPGA and VPU. • Free of charges
Video storing	<ul style="list-style-type: none"> • High storage device required • Continuous • Cost induced 	<ul style="list-style-type: none"> • Intel cloud resources • Only store when there is input • Free of charges

Based on Table 3.4, the stages of the project that required hardware were shown. The hardware cost will be induced during the running of the LPR system such as video capturing, video processing and video storing if the full local setup is chosen. In the market, there is a device that is composed of these 3 functions which is UP Squared AI Vision X Developer Kit. This kit cost \$419.00 which is RM 1901.63. Figure 3.22 shows the diagram of the developer kit.



Figure 3.22: UP Squared AI Vision X Developer Kit

On the other hand, these processing are free of charge. Therefore, it can be deduced that the cost can be greatly saved if the cloud resources are being utilised.

3.5 Summary

In this chapter, the required resources such as hardware, software, and cloud were discussed. By understanding their specification and limitation, the development can be done in a proper way. This highly reduces the time to troubleshoot the problems and have a better direction in seeking alternative solutions. Furthermore, the incompatibility issue also can be avoided.

Moreover, the work plan is also discussed from the beginning of the hands-on to the end. In summary, the hands-on started by deploying the existing LPR system to the cloud environment, followed by developing the new license plate detection model, and lastly improving the LPR system by replacing the old detection model (YOLOv4-tiny) with the new detection model (YOLOv7 series).

In addition, the work budget is also being compared with and without a cloud environment. It is clearly shown that the setup with the cloud environment is better as it can cut down the cost-free of charge.

The purpose of this discussion is to ensure the project is carried out in the correct direction. Most importantly, it is to ensure the reproducibility of the projects.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

In this chapter, the results of the project are presented according to 2 major sections: Evaluation of the Intel hardware in the cloud environment and comparison of performance between the YOLOv7 series.

For the first section (4.2), the license plate detection model used is YOLOv4-tiny, and the recognition model used is ResNet-FC. To be consistent, the input video fed into the model is the same throughout the project. By varying the job submission node, the performance of each Intel hardware is evaluated from the perspectives of FPS, latency and inferencing time

For the second section (4.3), the YOLOv4-tiny was replaced with YOLOv7, YOLOv7-tiny, and YOLOv7x. The performance of the YOLOv7 series is compared, and the most suitable detection model from the YOLOv7 series will be determined for real-time application.

In addition, the benefits and drawbacks of this project which are based on the cloud environment were discussed.

4.2 Evaluation of Intel Hardware

The 5 popular Intel hardware as mentioned in section 3.2.3 will be used for evaluation. The hardware is as follows:

- Intel® Xeon® Gold 6258R Processor
- Intel® Core™ i7-1065G7 Processor
- Intel® Xeon® Gold 6338N Processor
- Intel® Core™ i7-1185G7E Processor (with integrated GPU Intel® Iris® Xe Graphics)
- Intel Atom® x6425RE Processor (with integrated GPU Intel HD Graphics 530).

By submitting the job to different nodes, the existing LPR system (YOLOv4-tiny) will infer the input video using different Intel hardware. The results are shown in Figure 4.1 and Figure 4.2 for Inferencing Engine Processing Time and Inferencing Engine FPS respectively.

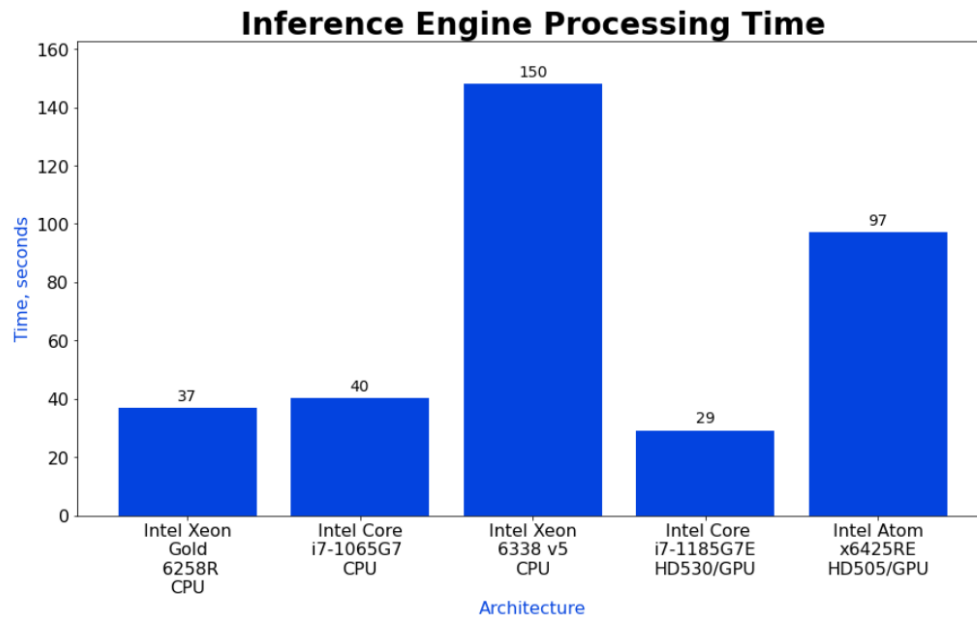


Figure 4.1: Inferencing Engine Processing Time

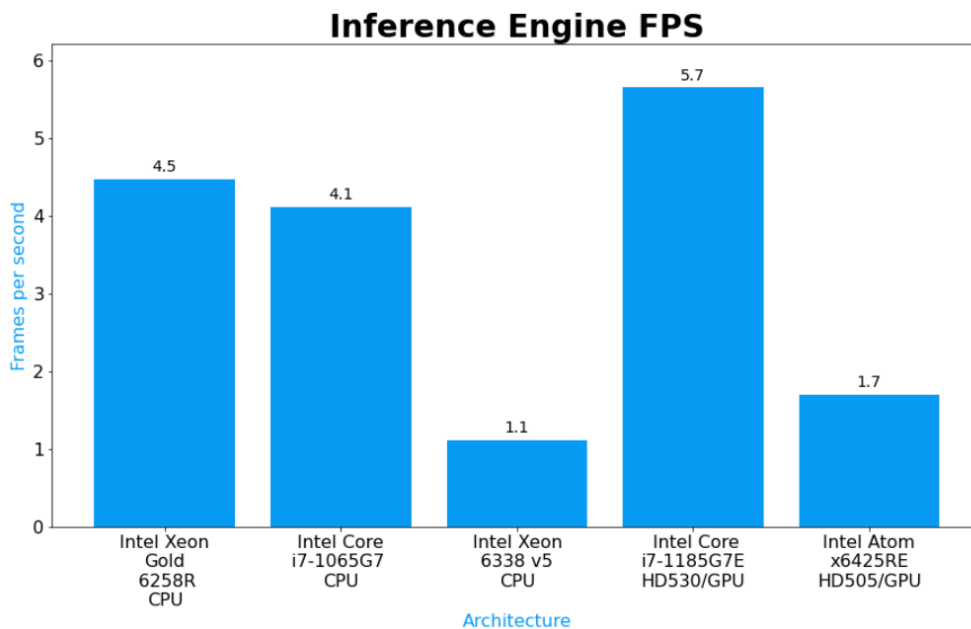


Figure 4.2: Inferencing Engine FPS

The processing time is the total time to execute the instruction from the beginning of the code to the end in the Python script. However, the time for the initialization section in the code is very prompt, and hence it is negligible. Therefore, the processing time in Figure 4.1 is also known as inferencing time. The FPS is calculated using equation 4.1.

$$FPS = \frac{\text{number of frames}}{\text{processing time}} \quad (4.1)$$

Thus, it can be concluded that the video input has a number of frames of 165. Based on Figure 4.1 and Figure 4.2, among the 5 Intel hardware, the inference engine with Intel® Core™ i7-1185G7E with integrated GPU Intel® Iris® Xe Graphics 530 had the highest performance. This inference engine had a processing time of 29 s and an FPS of 5.7. While the lowest performance is the Intel® Xeon® Gold 6338N, which has a processing time of 140 s and an FPS of 1.2.

There are a few reasons that result in these outcomes. The factors are compared in the table form as shown in Table 4.1 below.

Table 4.1: Comparison between Intel® Core™ i7-1185G7E & Intel® Xeon® Gold 6338N

	Intel® Core™ i7-1185G7E	Intel® Xeon® Gold 6338N
GPU	Intel® Iris® Xe Graphics 530	None
Vertical Segment	Embedded	Server
Max Memory Size	64 GB	6 TB
Total Core	4	32
Total Threads	8	64
Packaging Size	45.5mm x 25mm	77.5mm x 56.5mm

Firstly, Intel® Core™ i7-1185G7E has an integrated GPU that's meant for graphical processing. This is a huge advantage over Intel® Xeon® Gold 6338N. Moreover, the Intel® Core™ i7-1185G7E is designed for embedded systems with lesser cores and threads, which focuses on smaller processing. Meanwhile, Intel® Xeon® Gold 6338N is built mainly for server usage, which required more cores and threads to process the computation that had a maximum memory size of up to 6TB. Last but not least, the packaging size would also affect the computational speed. The smaller the size, the shorter path for signal transmission, resulting in shorter processing time.

4.3 Comparison of Performances of the Yolov7 series

After performing the LPR system using the YOLOv4-tiny in the cloud environment, the YOLOv4-tiny is now replaced with YOLOv7, YOLOv7-tiny, and YOLOv7x. To ensure consistency, the default CPU will be used for inferencing. The default CPU will be Intel® Xeon® Gold 6128 processor.

The performances of these 3-detection models will be compared in terms of inferencing FPS & latency, and also mean average precision (mAP).

4.3.1 FPS & Latency

The processing time and FPS results were shown in Figure 4.3 and Figure 4.4 after using the 3 types of YOLOv7 series detection models in the LPR system.

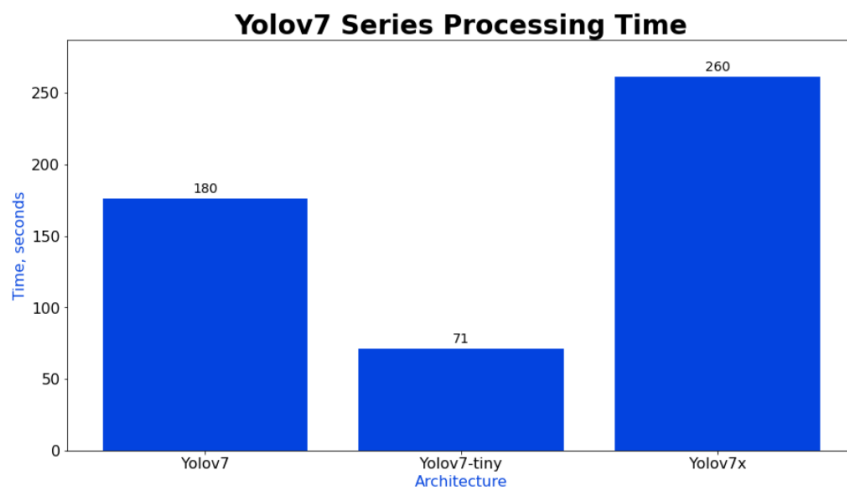


Figure 4.3: Yolov7 Series Processing Time

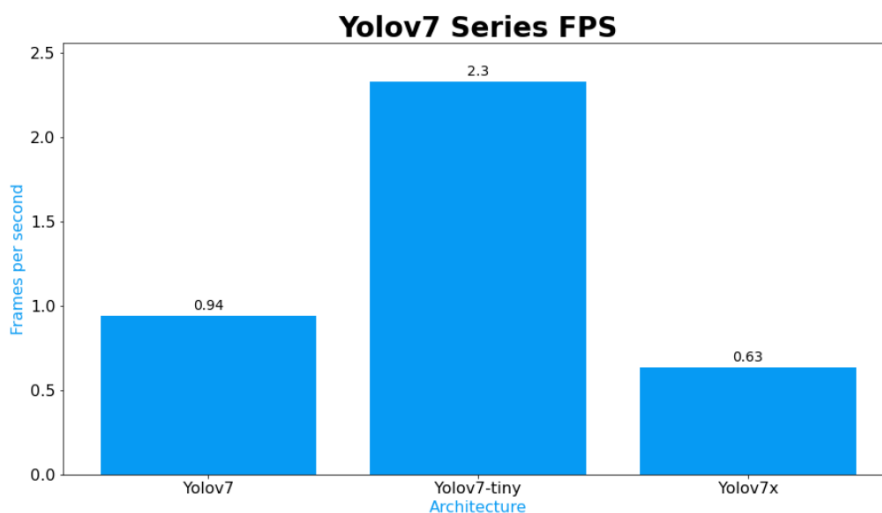


Figure 4.4: Yolov7 Series FPS

Based on the formula, the number of frames obtained will be 165, which indicated that the same input is fed into the LPR system. From Figure 4.3 and Figure 4.4, the highest performance will be YOLOv7-tiny followed by YOLOv7 and lastly YOLOv7x.

This is solely due to the size of the neural network. Table 4.2 below shows the number of layers & parameters of each neural network. Most of the time, the parameters are referred to as the neurons.

Table 4.2: Number of Layers & Parameter of Each Neural Network

Model	Layers	Parameters
YOLOv7	314	36481772
YOLOv7-tiny	208	6007596
YOLOv7x	362	70782444

In general, the larger the network size, the longer the time taken for inferencing. This is because of the increased complexity of the network, and more computation required to perform inference and make a prediction.

4.3.2 Benchmarking

The benchmarking of the new detection models and the results are tabulated in Table 4.3 below. This process is done by running the Python “test” script. For better performance comparison, the values – P, R, mAP@.5, and mAP@.5:.95 will be discussed beforehand.

Table 4.3: Benchmarking of Yolov7 Series

	P	R	mAP@.5	mAP@.5:.95
YOLOv7	0.943	0.877	0.930	0.714
YOLOv7-tiny	0.947	0.861	0.936	0.720
YOLOv7x	0.953	0.872	0.942	0.717

The P stands for “Precision”. It is the ratio of true positive (TP) detections to the total number of positive detections (TP + false positive (FP)) as shown in equation 4.2. It examines the model's ability to accurately detect

objects. For better illustration, Table 4.4 is tabulated to show the entire confusion matrix.

$$P = \frac{TP}{TP+FP} \quad (4.2)$$

where P = Precision

TP = True-Positive

FP = False-positive

Table 4.4: Confusion Matrix

		Actual	
		True	False
Predicted	Positive	True-Positive (TP), object present and model detected	False-Positive (FP), object absent and model detected
	Negative	True-Negative (TN), object present and model did not detect	False-Negative (FN), object absent and model did not detect

Next, R stands for “Recall”. It is the ratio of TP detections to total ground truth positives (TP + FN) as shown in equation 4.3. It assesses the model's ability to detect all relevant items.

$$R = \frac{TP}{TP+FN} \quad (4.3)$$

where R = Recall

TP = True-Positive

FN = False-Positive

Furthermore, noticed that there are mAP@.5 and mAP@.5:.95. In general, both are mean average precision that is commonly used as evaluation

metrics in object detection tasks. In general, mAP can be calculated by equation 4.4 below.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (4.4)$$

where AP_i = the average Precision of class i
 N = number of classes

In this project, the only class is “license_plate”, hence $mAP = AP$. By rule of thumb, the threshold of 0.5 is used in determining the mAP, and it is denoted as “mAP@.5”. mAP@.5 calculates the model's AP score at an Intersection of Union (IoU) threshold of 0.5. This means that the IoU between the anticipated bounding box and the ground truth bounding box must be larger than or equal to 0.5 for an item detection to be declared true positive. The demonstration is done through Figure 4.5 with the assistance of equation 4.5 below.

$$IoU(A, B) = \frac{A \cap B}{A \cup B} \quad (4.5)$$

where A = predicted box area
 B = ground truth box area

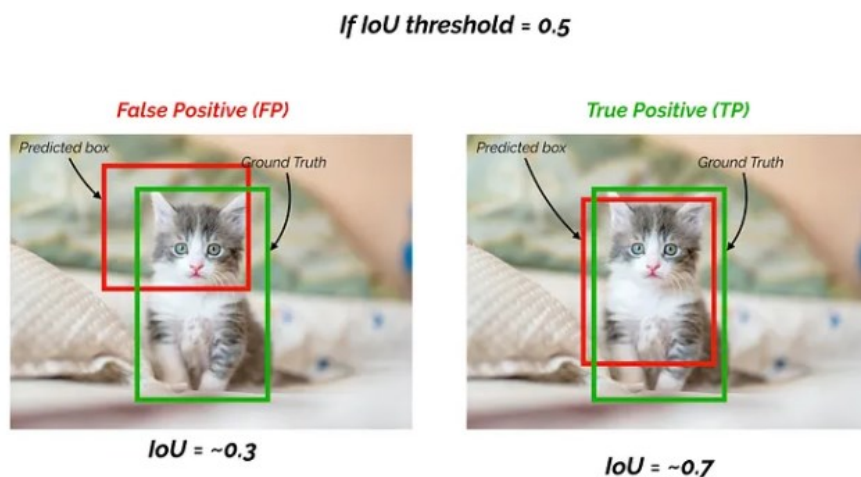


Figure 4.5: Demonstration on the IoU(Yohanandan, 2020)

On the other hand, $\text{mAP}@.5:.95$ is to resolve the restriction of $\text{mAP}@.5$, which does not take into account the object detector's ability to detect objects with a higher level of overlap. $\text{mAP}@.5:.95$ calculates the model's AP score across a range of IoU thresholds, from 0.5 to 0.95, using a 0.05 step size. This implies that the performance of the model is assessed for different amounts of overlap between the predicted and ground truth bounding boxes. The AP score for each class is determined for each IoU threshold, and the average AP score across all classes is calculated. Finally, the $\text{mAP}@.5:.95$ is determined as the mean of all IoU threshold AP values.

Even though the value of $\text{mAP}@.5:.95$ is lower, it gives a more comprehensive assessment of the model's performance since it takes into account the object detector's ability to recognise things properly over a variety of IoU thresholds.

Referring back to Table 4.3, the overall performances of the 3 detection models are hard to rank, since all the values are almost the same. This indicating the models are all trained well.

However, when the 3 models' evaluation metrics included Figure 4.3 and Figure 4.4, the increasing order of the overall performances will be YOLOv7x, followed by YOLOv7, and lastly YOLOv7-tiny. The reason is that YOLOv7-tiny was able to perform inferencing with the highest FPS, and lowest latency among all 3 models. With an FPS of 2.3, this YOLOv7-tiny is suitable for real-time applications.

4.4 Benefits of the Project

The factors that drove or aroused this project due to several benefits such as minimal setup effort, highly portable, and lastly free trial on Intel Hardware in the cloud environment (Intel® Developer Cloud for the Edge). These few benefits will be discussed in this section.

4.4.1 Minimal Setup Effort

For the local environment, the packages and libraries that are required for Python coding needed to install before the function inside the Python script can be used. This potentially causes the issues such as incompatible packages and an outage of memory storage for installing the packages.

On the other hand, in the cloud environment, the issues are resolved. The only thing that needs to do is to activate the pre-installed environment that contains all the necessary dependencies. Also, upload all the necessary files. Hence, the time to process environment setup can be greatly saved.

By providing a minimum setup and configurations, Intel can assure users can quickly and simply get up and running on the cloud environment without having to spend a significant amount of time customizing their environment. This is especially crucial for developers who may be working on numerous projects at the same time and need to transition between environments fast.

Last but not least, a minimal setup can help to reduce Intel® Developer Cloud for the Edge resource usage, allowing more users to access the platform at the same time. This is especially significant for users who are conducting computationally intensive applications and require access to strong hardware resources.

In brief, the minimum configuration provided users with a lightweight, efficient, and simple environment for writing and testing code on Intel architecture, while also maximising platform availability and utilisation of resources.

4.4.2 Portable

As long as the LPR system had been deployed to the cloud environment, it can be accessed anytime, anywhere, on any device with the presence of an internet connection. For example, Figure 4.6 that is using a smartphone for inferencing instead of using a laptop.

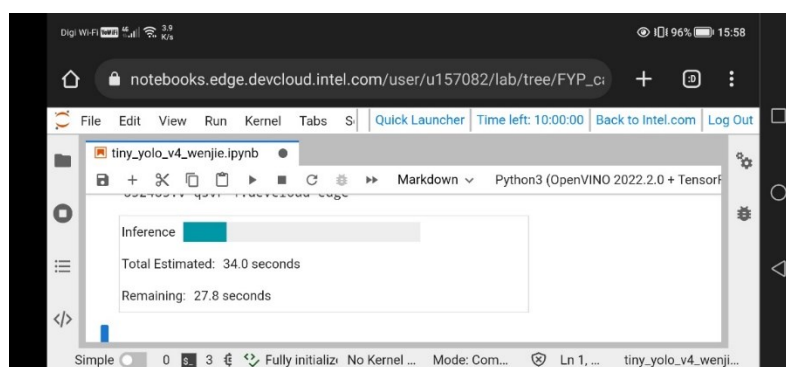


Figure 4.6: Smartphone for Inferencing

The remarkable fact is that when the user opens the Intel® Developer Cloud for the Edge simultaneously on the smartphone and laptop, it will not synchronize on time. It will delay for approximately 2 minutes to update. It will pop out the message to confirm with the user to either revert or overwrite the files.

Even though this feature provides the user with the huge convenience to view the progress of development. However, it is advised to not open the Intel® Developer Cloud for the Edge simultaneously on multiple devices, to avoid any mistakes such as replication and removal of the work done.

4.4.3 Free Trial on Intel Hardware

Intel® Developer Cloud for the Edge provides free access to Intel hardware such as Intel® Xeon® Scalable processors, Intel® Core™ processors, and Intel® Movidius™ Vision Processing Units (VPUs). This enables developers to test and optimise their edge applications on real-world hardware without having to invest in expensive hardware. This further lowers the expenses related to hardware acquisition, maintenance, and updates.

In addition, by using a variety of Intel hardware, the identification of performance bottlenecks can be done, and ensuring that applications function well in real-world circumstances. This indirectly assisted the developers to optimise their applications solution for improved performance and efficiency, resulting in faster and more accurate edge processing.

After testing all the available Intel hardware, the developer can acquire a better understanding of their applications' hardware needs. This knowledge can be utilised to make smarter decisions when acquiring Intel gear for personal use. Based on the insights gathered from testing their apps on the Intel® Developer Cloud for the Edge, developers should have the ability to select hardware that suits their unique performance needs and budget.

4.5 Drawbacks of the project

For any project, when there are pros, there will be always accompanied by cons. The drawbacks of the project would be the mandatory internet connection, Chinese characters and other languages cannot be recognized, and the cloud resource is fixed. These disadvantages will be discussed in this section.

4.5.1 Internet Connectivity

To get access to the cloud environment, an internet connection is necessary. In this modern era, without an internet connection, a lot of things cannot be done especially for programmers, and software developers that consistently required to look for the solution, resources, packages or libraries.

However, the internet connection is not a major issue in Malaysia. According to Digital Business survey, Malaysia has one of the highest internet penetration rates in South Asia. The country's internet penetration rate was 84% in 2021 (Allo Technology Sdn. Bhd., 2023). Additionally, the forecast of internet penetration will be increased to 89.57 % by the year 2025 (Statista, 2021).

In summary, the internet connectivity issue is negligible in Malaysia as Malaysia is a fast-developing country that very takes care of the people. But the performance would be greatly affected depending on the speed of the internet.

4.5.2 Chinese and Other Languages Characters

One of the drawbacks is the limited language characters can be recognized. For example, Chinese characters have simplified versions and traditional versions that further increased the complexity.

Another reason for this limited recognition power is due to the dataset that fed into the model during the training. By looking at the database of the lecturer, there are only images of Malaysia license plates that are mostly a combination of 26 English characters and 10 integer numbers.

Although only 26 English characters and 10 integer numbers can be recognized, it is more than enough as this LPR system is targeting to deploy in Malaysia only.

4.5.3 Fix Cloud Resources

The free cloud resources could bring a lot of benefits to us, but at the same time would cause inconvenience too. For example, the problem that meets in the project is unable to customize the environment of the node when submitting the job to it. The dependencies for each of the nodes are fixed, the user cannot install anything on the nodes, only can use the dependencies. As a consequence, the

LPR system with the YOLOv4-tiny can submit the job to the node, however, the LPR system with the YOLOv7 series cannot.

However, when this issue is analysed from another perspective, it is a good action taken by Intel. The fixing of the packages ensures all the users had the same resources to use. If all the users have access to install anything they wanted, the particular node will be messed up. Hence, standardizing the nodes' software is a good approach to guarantee the smoothness of the whole operation in Intel® Developer Cloud for the Edge.

On the other hand, for each user, Intel is very generous to give access to the user for the customization features only on the default CPU, Intel® Xeon® Gold 6128 processor. Therefore, the LPR system with the YOLOv7 series is deployed and evaluated based on this CPU.

4.6 Summary

In short, it can be deduced that the best Intel inference engine to run the LPR system is Intel® Core™ i7-1185G7E Processor with integrated GPU Intel® Iris® Xe Graphics. With its price and specifications stated in section 0, and the discussion in section 4.2, it has a performance of 29 seconds of processing time and FPS of 5.7.

Next, by deploying the YOLOv7 series detection models to the cloud, it can be deduced that under the same hyperparameters, input video and inference engine, the YOLOv7-tiny detection model has the highest performance with a processing time of 71 seconds and FPS of 2.3.

Last but not least, by utilizing Intel® Developer Cloud for the Edge, the benefits are minimal setup effort, highly portable and better planning in purchasing Intel hardware. In contrast, the drawbacks are the mandatory internet connection, Chinese and other language characters cannot be recognized, and the cloud resources are fixed.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

In this project, all the aims and objectives were achieved. The existing cloud service providers were compared, and selected Intel® Developer Cloud for the Edge as it is user-friendly, and most importantly it is free of charge. The second objective is to implement the model with increased performance (lightweight and accurate). This is completed by replacing the old (YOLOv4-tiny) detection model with the new (YOLOv7-tiny) detection model. Lastly, the LPR system with the optimized detection and recognition model was successfully deployed to the cloud environment.

The performance of the LPR system is also affected by hardware. Therefore, by evaluating the Intel hardware, it can be concluded that Intel® Core™ i7-1185G7E Processor with integrated GPU Intel® Iris® Xe Graphics is designed for embedded systems. It has a performance of 29 seconds of processing time and an FPS of 5.7. It is also worth noting that this Intel hardware's price is \$474.00 (RM 2115.46), which is considered affordable based on the specification among all the 5 Intel hardware compared.

For the improved LPR system, with the detection model of YOLOv7-tiny, it has the detection accuracy up to $mAP@0.5 = 0.936$, and $mAP@0.5:0.95 = 0.720$. For the mAP of 3 detection models, the differences are negligible. However, when come to inferencing time, the YOLOv7-tiny has the shortest as compared to YOLOv7 and YOLOv7x. Hence, YOLOv7-tiny will be replacing the YOLOv4-tiny in the LPR system.

Therefore, the combination of the Intel hardware (Intel® Core™ i7-1185G7E Processor with integrated GPU Intel® Iris® Xe Graphics) and LPR system with YOLOv7-tiny detection model will be the most suitable for real-time application.

Last but not least, cloud computing and local computing have their own pros and cons. The selection is always depending on the developers' specifications and requirements. In this project, even though cloud computing

has its own drawbacks as discussed, these almost cause negligible effect. Hence, the cloud environment was chosen to perform the computation.

In a nutshell, the project is considered a success.

5.2 Recommendations for Future Work

Malaysian licence plates are classified into two types: single-row and double-row. The imbalance ratio of both categories in the present license plate dataset contributes to the root reason for poor recognition accuracy. Therefore, the gathered dataset should contain a roughly 1:1 ratio of both categories.

The next recommendation would be to include the technique of geofencing when performing detection. Geofencing is a technique for zoning a particular area. When an object is discovered within the boundary limit of a simple virtual boundary surrounding a certain region, another action is triggered. In this recommendation, geofencing can be used to limit the licence plate detection area to a specific location. The size of the geofencing can be adjusted depending on the conditions at the testing location. The purpose of using geofencing in the LPR system is to reduce processing resources while improving LP identification accuracy.

Since this project is done in the cloud environment, it is recommended to relocate the device to a place with a stronger internet connection and wide internet coverage. Moreover, if the financial condition is allowed, the antenna could be bought to ensure the internet line is actively pulled to the device. This is to ensure the project is conducted in a better condition.

REFERENCES

Alahmari, S.S., Goldgof, D.B., Mouton, P.R. and Hall, L.O., 2020. Challenges for the Repeatability of Deep Learning Models. *IEEE Access*, 8, pp.211860–211868. <https://doi.org/10.1109/ACCESS.2020.3039833>.

Albelwi, S. and Mahmood, A., 2017. A Framework for Designing the Architectures of Deep Convolutional Neural Networks. *Entropy*, 19(6), p.242. <https://doi.org/10.3390/e19060242>.

Aldahwan, N.S. and Ramzan, M.S., 2022. Descriptive Literature Review and Classification of Community Cloud Computing Research. *Scientific Programming*, 2022, pp.1–12. <https://doi.org/10.1155/2022/8194140>.

Allo Technology Sdn. Bhd., 2023. *Here is How Malaysian Internet Use Changed in 2021*. [online] Available at: <<https://www.allo.my/blog-here-is-how-malaysia-internet-use/>> [Accessed 29 April 2023].

Amazon Web Services, 2022. *Machine Learning*. [online] Available at: <<https://aws.amazon.com/sagemaker/>> [Accessed 25 April 2023].

American Speech-Language-Hearing Association, 2023. *Who Are Speech-Language Pathologists, and What Do They Do?* [online] Available at: <<https://www.asha.org/public/who-are-speech-language-pathologists/>> [Accessed 25 April 2023].

ArcGis Developers, 2022. *How Mask R-CNN Works?* [online] Available at: <<https://developers.arcgis.com/python/guide/how-maskrcnn-works/>> [Accessed 25 April 2023].

Azure, 2022a. *Azure Machine Learning*. [online] Available at: <<https://azure.microsoft.com/en-us/products/machine-learning/#product-overview>> [Accessed 23 April 2023].

Azure, 2022b. *What Is Cloud Computing?*. [online] Available at: <<https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-cloud-computing/#benefits>> [Accessed 25 April 2023].

Bigelow, S.J., 2021. *What Is Edge Computing? Everything You Need to Know*. [online] Available at: <<https://www.techtarget.com/searchdatacenter/definition/edge-computing>> [Accessed 25 April 2023].

Boesch, G., 2023. *YOLOv7: The Fastest Object Detection Algorithm (2023)*. [online] viso.ai. Available at: <<https://viso.ai/deep-learning/yolov7-guide/>> [Accessed 30 April 2023].

Brownlee, J., 2019. *How to Use Mask R-CNN in Keras for Object Detection in Photographs* . [online] Available at: <<https://machinelearningmastery.com/how-to-perform-object-detection-in-photographs-with-mask-r-cnn-in-keras/>> [Accessed 25 April 2023].

Caprolu, M., Di Pietro, R., Lombardi, F. and Raponi, S., 2019. Edge Computing Perspectives: Architectures, Technologies, and Open Security Issues. *Proceedings - 2019 IEEE International Conference on Edge Computing, EDGE 2019 - Part of the 2019 IEEE World Congress on Services*, pp.116–123. <https://doi.org/10.1109/EDGE.2019.00035>.

Chandrasekhara Reddy, T., Sirisha, G. and Reddy, A.M., 2018. Smart Healthcare Analysis and Therapy for Voice Disorder using Cloud and Edge Computing. *Proceedings of the 4th International Conference on Applied and Theoretical Computing and Communication Technology, iCATccT 2018*, pp.103–106. <https://doi.org/10.1109/ICATCCT44854.2018.9001280>.

Datacenters.com Cloud, 2022. *Taking AI to the Cloud*. [online] Available at: <<https://www.datacenters.com/news/artificial-intelligence-in-cloud-computing>> [Accessed 23 April 2023].

Fisher, T., 2020. *Five Things You Need At The Edge For Real-Time Intelligence*. [online] Available at: <<https://www.forbes.com/sites/forbestechcouncil/2020/06/26/five-things-you-need-at-the-edge-for-real-time-intelligence/?sh=2c2f441fcc9a>> [Accessed 23 April 2023].

Foote, K.D., 2021. *A Brief History of Cloud Computing - DATAVERSITY*. [online] Available at: <<https://www.dataversity.net/brief-history-cloud-computing/#>> [Accessed 23 April 2023].

Freebie Supply, 2023. *Ubuntu Logo PNG Transparent & SVG Vector* . [online] Available at: <<https://freebiesupply.com/logos/ubuntu-logo/>> [Accessed 28 April 2023].

Fritz AI, 2022. *Object Detection Guide*. [online] Available at: <<https://www.fritz.ai/object-detection/#top>> [Accessed 23 April 2023].

Gillis, A.S., 2022. *What is 5G? Definition, Benefits and Use Cases - TechTarget.com*. [online] Available at: <<https://www.techtarget.com/searchnetworking/definition/5G>> [Accessed 23 April 2023].

Google, 2023a. *Brand Resource Center | Brand terms*. [online] Available at: <<https://about.google/brand-resource-center/logos-list/>> [Accessed 28 April 2023].

Google, 2023b. *Google Colab*. [online] Available at: <<https://research.google.com/colaboratory/faq.html>> [Accessed 23 April 2023].

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J. and Chen, T., 2018. Recent advances in convolutional neural networks. *Pattern Recognition*, 77, pp.354–377. <https://doi.org/10.1016/J.PATCOG.2017.10.013>.

Gulati, A., Holler, A., Ji, M., Shanmuganathan, G., Waldspurger, C. and Zhu, X., 2012. VMware Distributed Resource Management: Design, Implementation, and Lessons Learned. *VMware Technical Journal*, 1(1), pp.45–64.

Hosch, W.L., 2019. *supercomputer*. [online] Britannica. Available at: <<https://www.britannica.com/technology/supercomputer>> [Accessed 23 April 2023].

Huawei Cloud, 2023. *What Is ModelArts?* [online] Available at: <https://support.huaweicloud.com/intl/en-us/productdesc-modelarts/modelarts_01_0001.html> [Accessed 23 April 2023].

Huawei Enterprise Support Community, 2022. *What is AI?* . [online] Available at: <<https://forum.huawei.com/enterprise/en/what-is-ai/thread/691643-895?page=1&authorid=2976461>> [Accessed 23 April 2023].

IBM, 2023a. *What are Convolutional Neural Networks?* . [online] Available at: <<https://www.ibm.com/topics/convolutional-neural-networks>> [Accessed 30 April 2023].

IBM, 2023b. *What Is Edge Computing*. [online] Available at: <<https://www.ibm.com/cloud/what-is-edge-computing>> [Accessed 23 April 2023].

Intel, 2023a. *Intel® Advanced Vector Extensions 512 (Intel® AVX-512) Overview*. [online] Available at: <<https://www.intel.com/content/www/us/en/architecture-and-technology/avx-512-overview.html>> [Accessed 28 April 2023].

Intel, 2023b. *Intel Core i71065G7 Processor 8M Cache up to 3.90 GHz Product Specifications*. [online] Available at: <<https://ark.intel.com/content/www/us/en/ark/products/196597/intel-core-i71065g7-processor-8m-cache-up-to-3-90-ghz.html>> [Accessed 27 April 2023].

Intel, 2023c. *Intel Core i71185G7E Processor 12M Cache up to 4.40 GHz Product Specifications*. [online] Available at: <<https://ark.intel.com/content/www/us/en/ark/products/208076/intel-core-i71185g7e-processor-12m-cache-up-to-4-40-ghz.html>> [Accessed 27 April 2023].

Intel, 2023d. *Intel® Hyper-Threading Technology*. [online] Available at: <<https://www.intel.com/content/www/us/en/architecture-and-technology/hyper-threading/hyper-threading-technology.html>> [Accessed 28 April 2023].

Intel, 2023e. *Intel NUC 10 Performance kit NUC10i7FNH Product Specifications*. [online] Available at: <<https://ark.intel.com/content/www/us/en/ark/products/188811/intel-nuc-10-performance-kit-nuc10i7fnh.html>> [Accessed 28 April 2023].

Intel, 2023f. *Intel® Software Guard Extensions*. [online] Available at: <<https://www.intel.com/content/www/us/en/developer/tools/software-guard-extensions/overview.html>> [Accessed 28 April 2023].

Intel, 2023g. *Intel® Time Coordinated Computing*. [online] Available at: <<https://www.intel.com/content/www/us/en/developer/tools/time-coordinated-computing-tools/overview.html>> [Accessed 28 April 2023].

Intel, 2023h. *Intel® Turbo Boost 2.0: High Performance Intel Turbo Boost Technology...* [online] Available at: <<https://www.intel.com/content/www/us/en/architecture-and-technology/turbo-boost/turbo-boost-technology.html>> [Accessed 28 April 2023].

Intel, 2023i. *Intel Xeon Gold 6258R Processor 38.5M Cache 2.70 GHz Product Specifications*. [online] Available at: <<https://ark.intel.com/content/www/us/en/ark/products/199350/intel-xeon-gold-6258r-processor-38-5m-cache-2-70-ghz.html>> [Accessed 27 April 2023].

Intel, 2023j. *Intel Xeon Gold 6338N Processor 48M Cache 2.20 GHz Product Specifications*. [online] Available at: <<https://ark.intel.com/content/www/us/en/ark/products/212633/intel-xeon-gold-6338n-processor-48m-cache-2-20-ghz.html>> [Accessed 27 April 2023].

Intel, 2023k. *Overview of Intel® Developer Cloud for the Edge*. [online] Available at: <<https://www.intel.com/content/www/us/en/developer/tools/devcloud/edge/overview.html>> [Accessed 23 April 2023].

Intel, n.d. Intel DevCloud for the Edge.

Jog, C., 2020. *The Two Benefits of the ONNX Library for ML Models*. [online] Medium. Available at: <<https://medium.com/trueface-ai/two-benefits-of-the-onnx-library-for-ml-models-4b3e417df52e>> [Accessed 29 April 2023].

Keras, 2022. *Keras: Deep Learning for humans*. [online] Available at: <<https://keras.io/>> [Accessed 23 April 2023].

Kranz, G., 2021. *What is Amazon SageMaker?* [online] Available at: <<https://www.techtarget.com/searchaws/definition/Amazon-SageMaker>> [Accessed 25 April 2023].

Kumar, N., 2022. *Understanding The Concept Of Convolutional Neural Networks (CNNs) - MarkTechPost.* [online] Available at: <<https://www.marktechpost.com/2022/01/24/a-detailed-understanding-of-convolutional-neural-networks/>> [Accessed 30 April 2023].

Kundu, R., 2023. *YOLO Algorithm for Object Detection Explained [+Examples].* [online] Available at: <<https://www.v7labs.com/blog/yolo-object-detection>> [Accessed 30 April 2023].

Lahon, A., 2020. *5 Statistical Functions in PyTorch .* [online] Towards Data Science. Available at: <<https://towardsdatascience.com/5-statistical-functions-in-pytorch-2d75e3dcc1fd>> [Accessed 28 April 2023].

Lockwood. John, 2023. *Jupyter Notebook: A Complete Introduction - CodeSolid.com.* [online] Available at: <<https://codesolid.com/jupyter-notebook-a-complete-introduction/>> [Accessed 28 April 2023].

logowik, 2023. *OpenVINO Logo PNG vector in SVG, PDF, AI, CDR format.* [online] Available at: <<https://logowik.com/openvino-logo-vector-37936.html>> [Accessed 28 April 2023].

MLACOM, 2023. *INTEL NUC 10 Performance kit NUC10i7FNH, i7-10710U (BXNUC10I7FNH2) .* [online] Available at: <https://www.mlacom.si/intel/i_2108036_intel-nuc-10-performance-kit-nuc10i7fnh-frost-canyon-bxnuc10i7fnh> [Accessed 28 April 2023].

MyBinder, 2023. *Binder.* [online] Available at: <<https://mybinder.org/>> [Accessed 28 April 2023].

Nguyen, Q.H., Ly, H.-B., Ho, L.S., Al-Ansari, N., Le, H. Van, Tran, V.Q., Prakash, I. and Pham, B.T., 2021. Influence of Data Splitting on Performance of Machine Learning Models in Prediction of Shear Strength of Soil. *Mathematical Problems in Engineering*, 2021, pp.1–15. <https://doi.org/10.1155/2021/4832864>.

Okeke, F., 2022. *The Benefits of Edge AI .* [online] TechRepublic. Available at: <<https://www.techrepublic.com/article/benefits-edge-ai/>> [Accessed 30 April 2023].

OpenCV, 2021. *OpenCV-Python Is Now An Official OpenCV Project.* [online] Available at: <<https://opencv.org/opencv-python-is-now-an-official-opencv-project/>> [Accessed 28 April 2023].

OpenCV, 2023. *About .* [online] Available at: <<https://opencv.org/about/>> [Accessed 28 April 2023].

O'Shea, K. and Nash, R., 2015. An Introduction to Convolutional Neural Networks. *arXiv preprint arXiv:1511.08458*.

Parking Network, 2021. *Jieshun Deploy Malaysia's Largest LPR Parking System for Sunway Pyramid*. [online] Available at: <<https://www.parking.net/parking-news/jieshun/malaysias-largest-lpr-parking-system-for-sunway-pyramid>> [Accessed 23 April 2023].

PNGEgg, 2023. *Anaconda png images* . [online] Available at: <<https://www.pngegg.com/en/search?q=anaconda>> [Accessed 28 April 2023].

Python Software Foundation, 2023. *The Python Logo*. [online] Available at: <<https://www.python.org/community/logos/>> [Accessed 28 April 2023].

Regalado, A., 2011. *Who Coined 'Cloud Computing'?* . [online] Available at: <<https://www.technologyreview.com/2011/10/31/257406/who-coined-cloud-computing/>> [Accessed 23 April 2023].

Saha, S., 2018. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way* . [online] Towards Data Science. Available at: <<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>> [Accessed 23 April 2023].

Schroer, A., 2023. *What Is Artificial Intelligence?* [online] Available at: <<https://builtin.com/artificial-intelligence>> [Accessed 23 April 2023].

Serea, R., 2022. *VMware Workstation Player 16.2.4 - Neowin*. [online] Neowin. Available at: <<https://www.neowin.net/software/vmware-workstation-player-1624/>> [Accessed 28 April 2023].

Sharma, P. and Jadhao, V., 2021. Molecular Dynamics Simulations on Cloud Computing and Machine Learning Platforms. *IEEE International Conference on Cloud Computing, CLOUD*, 2021-September, pp.751–753. <https://doi.org/10.1109/CLOUD53861.2021.00101>.

Shim, T., 2023. *15 Popular Platform as a Service (PaaS) Examples*. [online] Available at: <<https://www.webhostingsecretrevealed.net/blog/web-business-ideas/paas-examples/>> [Accessed 23 April 2023].

Shi, Y., Yang, K., Jiang, T., Zhang, J. and Letaief, K.B., 2020. Communication-Efficient Edge AI: Algorithms and Systems. *IEEE Communications Surveys and Tutorials*, 22(4), pp.2167–2191. <https://doi.org/10.1109/COMST.2020.3007787>.

Singh, R. and Gill, S.S., 2023. Edge AI: A survey. *Internet of Things and Cyber-Physical Systems*, 3, pp.71–92. <https://doi.org/10.1016/j.iotcps.2023.02.004>.

Sonnet, 2023. *eGPU Breakaway Box 750/750ex* . [online] Available at: <<https://www.sonnettech.com/product/egpu-breakaway-box/overview.html>> [Accessed 28 April 2023].

Statista, 2021. *Malaysia: internet penetration rate* . [online] Available at: <<https://www.statista.com/statistics/975058/internet-penetration-rate-in-malaysia/>> [Accessed 29 April 2023].

TechGig, 2020. *Understanding the difference between AI, ML, and DL*. [online] Available at: <<https://content.techgig.com/technology/understanding-the-difference-between-ai-ml-and-dl/articleshow/75493798.cms>> [Accessed 23 April 2023].

TrackVia, 2014. *Increase productivity by switching to the cloud (infographic)*. [online] Available at: <<https://trackvia.com/blog/cloud-computing/increase-productivity-switching-cloud/>> [Accessed 23 April 2023].

Triskele Labs, 2023. *Cloud cyber attacks: The latest cloud computing security issues*. [online] Available at: <<https://www.triskelelabs.com/blog/cloud-cyber-attacks-the-latest-cloud-computing-security-issues>> [Accessed 23 April 2023].

USoft, 2018. *The significant difference between AI, ML and Deep Learning*. [online] Available at: <<https://www.usoft.com/blog/difference-between-ai-ml-and-deep-learning>> [Accessed 23 April 2023].

VMware, 2023. *What is Software-Defined Networking (SDN)?* . [online] Available at: <<https://www.vmware.com/topics/glossary/content/software-defined-networking.html>> [Accessed 23 April 2023].

Wadawadagi, V., 2023. *PyTorch vs TensorFlow: Deep Learning Frameworks [2023]*. [online] Available at: <<https://www.knowledgehut.com/blog/data-science/pytorch-vs-tensorflow>> [Accessed 28 April 2023].

Wang, C.-Y., Bochkovskiy, A. and Liao, H.-Y.M., 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*.

Wong, K.Y., 2022. *Releases · WongKinYiu/yolov7*. [online] Available at: <<https://github.com/WongKinYiu/yolov7/releases>> [Accessed 23 April 2023].

Yamashita, R., Nishio, M., Do, R.K.G. and Togashi, K., 2018. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, [online] 9(4), pp.611–629. <https://doi.org/10.1007/S13244-018-0639-9>.

Yohanandan, S., 2020. *mAP (mean Average Precision) might confuse you!* . [online] Towards Data Science. Available at: <<https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2>> [Accessed 29 April 2023].