COMP4801 Final year project

FYP23010
A mobile application for navigating HKU visitors with computer vision

Supervisor: Dr. Luo Ping

Group Members:
Liu Kan Man, 3035790733
Wong Riley Hoi-kiu, 3035829039
Ng Enoch, 3035781536

Individual Report by Liu Kan Man, 3035790733

Date of Submission: 26/04/2024

Abstract

Engineers face significant challenges when it comes to indoor positioning and navigation, particularly in the complex building layout of the University of Hong Kong (HKU). Although GPS-based navigation apps like Google Maps are widely used, they cannot guarantee effective indoor navigation, leaving the problem of navigating to specific rooms or lecture venues unresolved. Additionally, research has shown that GPS is unreliable for indoor use. This project aims to overcome these limitations by developing a vision-based navigation system specifically tailored for guiding visitors within the HKU campus, focusing on the Main Building due to the scope limitation. The project also explores the potential applications of computer vision in positioning. This report presents the system's designs and implementations, as well as the computer vision (CV) model used for vision positioning. The system follows a client-server architecture, incorporating a mobile app, a backend service, a mapping service, a database, and a location detection model. The analysis conducted provides valuable insights on cost-effectiveness and limitations for the development of vision positioning systems. The paper also includes in-depth research on the potentials of GPT4 technology for vision-based localization. Future enhancements for the project involve integrating the system with emerging technologies and expanding the project scope in a cost-effective manner.

Acknowledgment

I am grateful for the guidance and support from our group supervisor, Dr. Luo Ping, and the Department of Computer Science.

My heartfelt thanks go out to my groupmates, Enoch Ng and Wong Riley Hoi-kiu, who have been playing important roles and contributing to the project with their passions and efforts.

Table of Contents

Table of Figures

Table of Tables

Abbreviations

app	Application
CV	Computer Vision
FPS	Frame per second
HKU	The University of Hong Kong
UI	User Interface
YOLO	You Only Look Once
SGD	Stochastic Gradient Descent
Adam	Adaptive Momentum Estimation
Adamax	Adaptive Momentum Estimation with Infinity Norm
AdamW	Adaptive Momentum Estimation with Weight Decay
NAdam	Nesterov-accelerated Adaptive Momentum Estimation
RAdam	Rectified Adaptive Momentum Estimation
RMSProp	Root-mean-square Propagation
API	Application Interface
GPT	Generative Pre-Trained Transformers
GPS	Global Position System

1 Introduction

1.1 Background

As one of the well-recognized universities, the campus of the University of Hong Kong (HKU) experiences a significant daily influx of individuals. Statistics [1] state that HKU has over 13,000 new students a year. Moreover, the university hosts a multitude of events, including talks, ceremonies, and workshops, occurring with great frequency. These events serve as a significant draw for visitors to the campus. Newcomers to the university often encounter confusion when they are presented with venue names and codes, leaving them uncertain about how to navigate their way to the designated venues before they get familiar with the campus. Nevertheless, the university currently lacks a dedicated navigation tool. Instead, visitors are compelled to rely on generic GPS-based navigation apps such as Google Maps, along with text-based guides sourced from the university's websites. Regrettably, these existing apps fail to provide visual representations of the indoor environments within the campus. Furthermore, research findings [2] indicate the inherent limitations of GPS technology in accurately determining precise indoor locations. As a result, there is a noticeable demand for a system to provide a substantial number of visitors with enhanced navigation experiences.

1.2 Existing Approach: GPS-based Navigation Applications

As delineated in Section 1.1, individuals depend on GPS-based navigation apps as their primary means of navigating within the HKU campus. However, these existing apps exhibit several limitations, including restricted functionality, deficient visualization capabilities, and unreliable navigation performance. Their reliance on satellite and aerial imagery to generate maps renders them incapable of comprehensively representing indoor environments. These apps are only able to identify the buildings but lack the capacity to ascertain the specific locations of individual rooms within said buildings. Consequently, users are left devoid of any visual depiction and information of the indoor spaces. Furthermore, papers [3], [4], [5] have substantiated the insufficiency and inaccuracy of GPS location information for indoor navigation. GPS signals are susceptible to interference from obstacles and adverse atmospheric conditions, thereby rendering their accuracy unreliable. The intricate building complex of the HKU campus exacerbates this issue, amplifying the challenges faced by users.

1.3 Motivation on developing vision-based navigation system

The motivation behind developing a vision-based navigation app stems from the limitations of the existing solutions mentioned in Section 1.2. A new navigation system can provide HKU visitors with user-friendly experiences, along with additional functionality.

The computer vision approach stands out among indoor navigation technologies due to its high accuracy, robustness, infrastructure flexibility, and additional functionalities.

Vision-based indoor positioning can leverage features like object recognition, depth sensing, and motion tracking to provide precise location estimates. The estimations are less susceptible to interference from environmental factors like signal attenuation, multi-path propagation, or radio frequency interference [3]. Furthermore, vision-based systems do not require the installation of additional infrastructure or hardware, like beacons or RFID tags [4]. Also, vision-based systems can provide additional functionalities beyond just positioning, like object tracking, gesture recognition, activity monitoring, and augmented reality overlays, enhancing the overall user experience [6]. Considering the rapid development and growing popularity of computer vision applications, a vision-based system has a large extendibility and a great potential to combine with different devices and technologies.

1.4 Objectives

The primary objective of this project is to deliver a dedicated mobile app that caters to the navigation needs of both students and visitors at HKU. The app should be able to facilitate precise detection of users' locations, efficient path navigation, and intuitive visualization of detection results and navigational guidance.

Another objective is to evaluate the cost-effectiveness of the computer vision method in comparison to other navigation approaches. This evaluation will encompass the optimization of each step involved in model development, including data collection, data processing, model training, and model fitting. By conducting a comprehensive cost-effectiveness analysis, this project aims to provide valuable insights and contribute to the existing body of knowledge on computer vision applications, particularly for Visual Positioning System (VPS).

1.5 Outline

The report is organized into 8 chapters. Chapter 1 provides an overview of the project's motivation and objectives. Chapter 2 presents the design details of the system, app, and model. Chapter 3 focuses on the final outcomes achieved, including the testing results. Chapter 4 highlights the encountered difficulty and the corresponding responses formulated. Chapter 5 outlines the project milestones. Chapter 6 discusses the future work to further enhance the project. Chapter 7 mphasizes the contributions made by each group member. Finally, Chapter 8 concludes the report, summarizing the project as a whole.

2 Methodology

This chapter provides a detailed discussion of the project's design and implementation. It begins with the system architecture (Section 2.1) and app design (Section 2.2), followed by an exploration of the mapping and navigation component (Section 2.3) and the backend service (Section 2.4). Moreover, the chapter introduces the model design (Section 2.5), the data processing flow (Section 2.6) and the database management system (Section 2.7).

2.1 System Architecture

The section presents an overview of the system architecture, followed by the flowcharts of the client side and server side.

2.1.1 System Overview

As shown in Figure 1, the system uses the client-server structure. The client side has the mobile app user interface (UI). Further details regarding the app UI are discussed in Section 2.2. The app is linked to Google Maps Service and FengMap Service via corresponding APIs. The mapping and navigation services are further explained in Section 2.3. The server handles the backend service and the CV model. More information about the backend service and the CV model is discussed in Section 2.4 and Section 2.5, respectively. Data for training the model is processed with the assistance of Roboflow, which is presented in Section 2.6. The database management system is employed to store data for both the client side and server side. Further information will be discussed in Section 2.7.

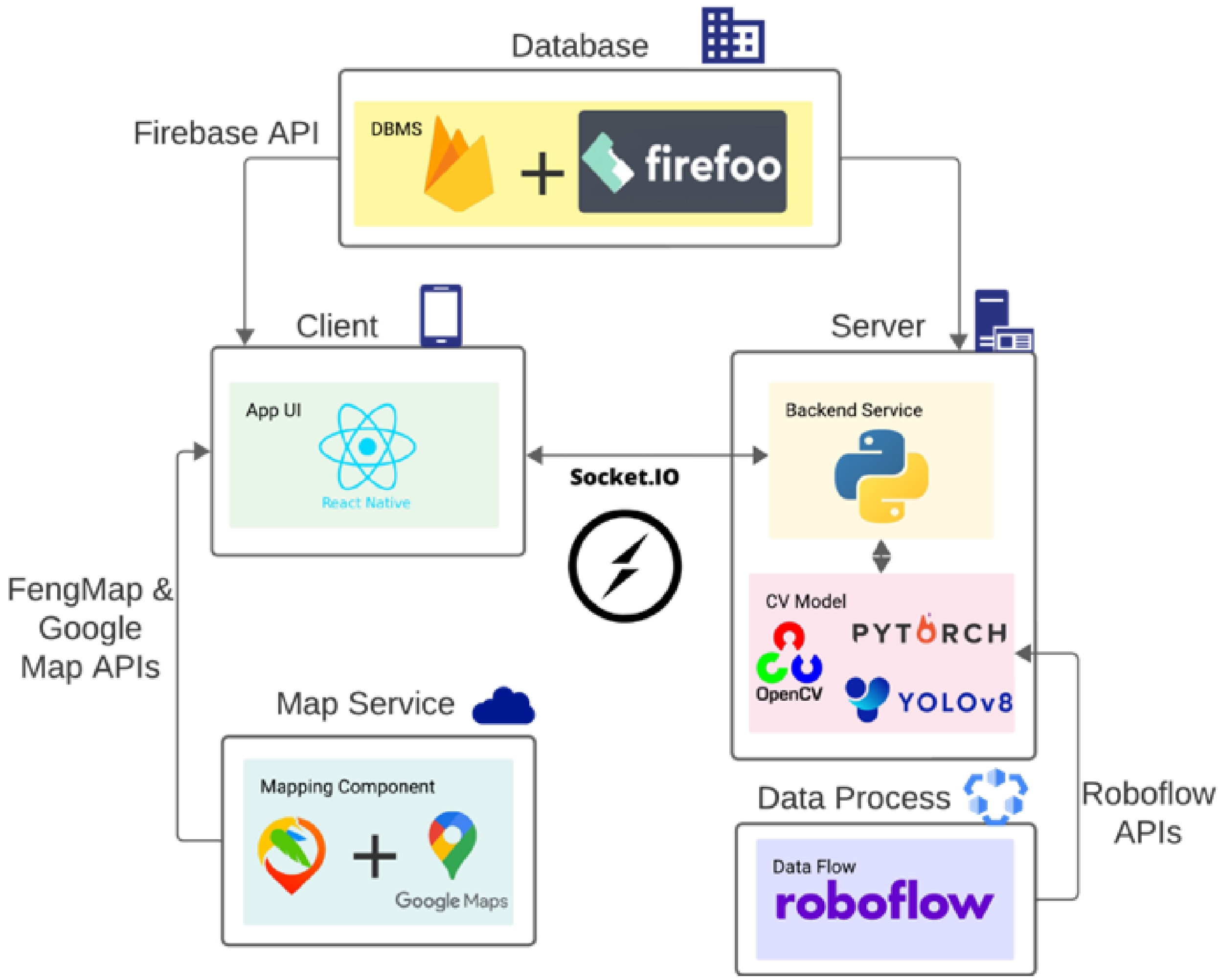


Figure 1 System structure diagram

2.1.2 System Flow

2.1.2.1 Client Side

As illustrated in Figure 2, the client-side functionality of the system follows the subsequent steps. Upon launching the app, the connectivity status is verified. If an active connection to the server is detected, the app is able to initiate the streaming of videos to the server and awaits a response. Conversely, in offline mode, users can set their current location using manual input. Once the current location is established and the destination is provided by the user, the mapping algorithm is available to optimize the generated path. The resulting optimized path is then rendered and displayed on the app UI.

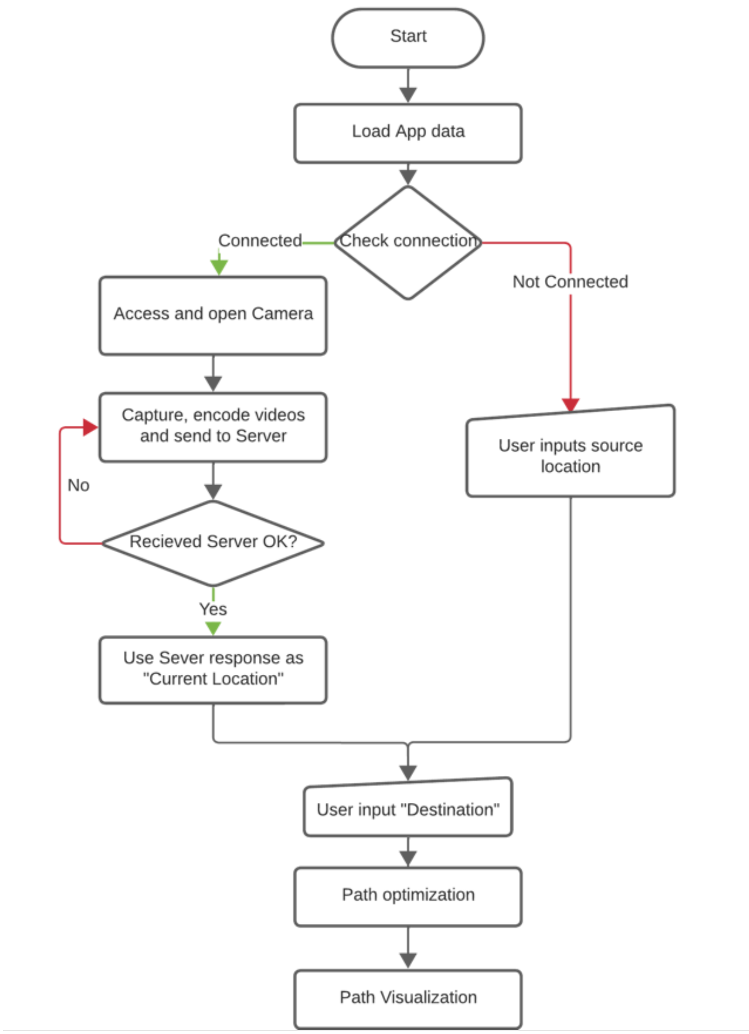


Figure 2 Flowchart of the client side

2.1.2.2 Server Side

The server-side workflow, as depicted in Figure 3, commences upon receiving a request from the client. The server establishes a connection and begins to receive the streamed photos. Subsequently, the received videos are sequentially decoded and processed by the object detection model. The model outputs the location estimation and associated scores. The server transmits the location result back to the client when the model returns the result.

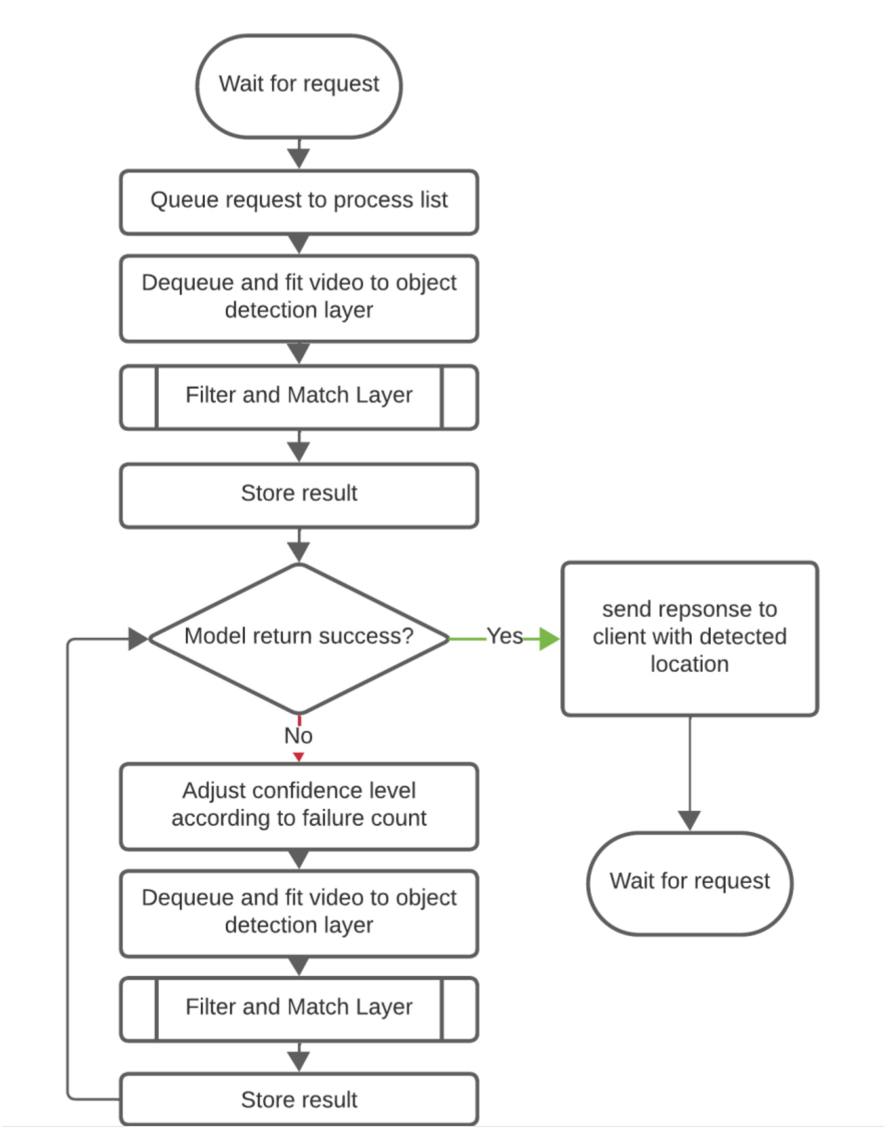


Figure 3 Flowchart of the server side

2.2 Mobile Application

This section starts with a brief introduction to the mobile app, followed by the features of the app. Additionally, the chapter discusses the selection of the development framework, considering the factors that influenced this decision.

2.2.1 App Introduction

The proposed app encompasses a UI that facilitates the visualization of a virtual map, the camera feed, and the inclusion of user controls. Additionally, the app is responsible for implementing pathfinding functionality, and providing information about the campus.

2.2.2 App Features

The app offers 2 major features: vision-based positioning and path optimization. Some additional features are implemented to facilitate users' navigation experience in HKU.

2.2.2.1 Vision-based Positioning

The app employs the camera of the user's smartphone to repeatedly capture videos of the surroundings with a predefined video length (i.e., 1 second). The videos are encoded into base64 format by the app. The encoded strings are securely transmitted to the server and processed using the detection model, which will be further discussed in Section

2.5 Model Design). Once a sufficient number of images have been analyzed, and the model's confidence level has been met, the resulting location is sent back to the app. As the app receives a successful result, it displays the detected location. This location information can be utilized for path finding purposes as well.

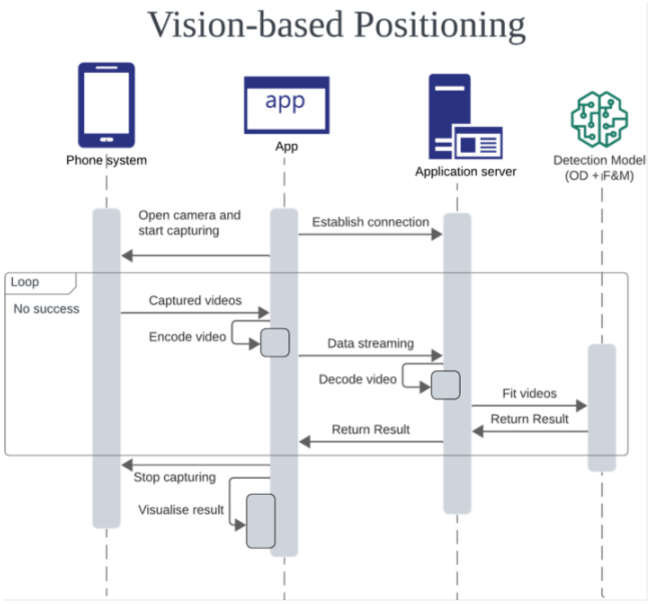


Figure 4 Sequence diagram for Vision-based Positioning

2.2.2.2 Path optimization

The app enables users to use their location and desired destination to navigate within the HKU campus. It then optimizes the best route and displays it on the map, providing navigational cues for guidance. This versatile approach ensures seamless navigation within the HKU campus, regardless of indoor or outdoor environment. Users can switch between outdoor map and indoor map easily.

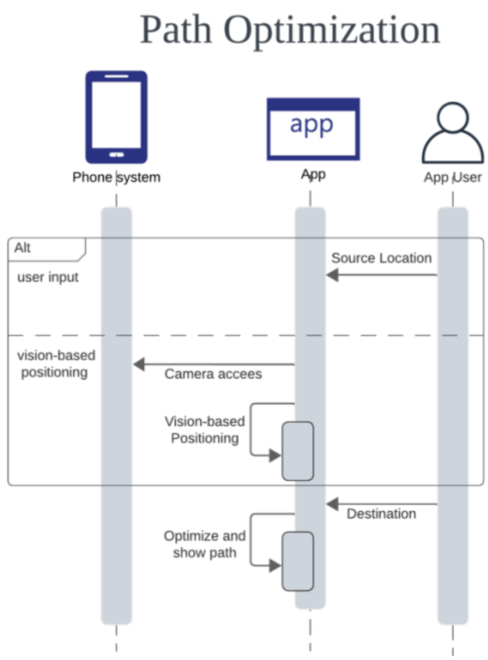


Figure 5 Sequence diagram for Path Optimization

2.2.2.3 Additional Feature: Accessing HKU Campus information

The app offers users a page to access comprehensive information about the HKU campus, by integrating HKU Map. This page provides details about various buildings, including pictures, locations, departments, websites, and available facilities for visitors. Users can effortlessly explore the campus while navigating with the app, enhancing their overall experience.

2.2.3 Selection for Frontend Development Framework: React Native

React Native was chosen for the mobile app UI development. React Native is elected for its cross-platform nature, the vast ecosystem, and the large community. It allows developers to write code once and deploy the code on both Android and iOS, saving time and effort compared to developing separate apps for each platform using other frameworks. Moreover, the extensive library of pre-built components available in the React Native ecosystem can be easily integrated into applications, reducing the need for building UI elements from scratch and accelerating development speed. Additionally, React Native has a large and active community. The strong community support adds value to React Native by providing a rich pool of resources and making it easier for developers to find solutions to challenges during the app development process.

2.3 Mapping and Navigation Component

This section describes the use of the mapping and navigation component and explains the selection of the development tool for the component.

2.3.1 Mapping and Navigation Component Introduction

The mapping and navigation component is responsible for mapping users' locations onto the campus map and optimizing the paths to the destination. The map should cover the scope of the project, that is the Main Building. This component will be using online mapping services to ensure the campus map is standardized and up to date.

2.3.2 Selection of Development Tool

There are a number of choices for the mapping development tool, including Google Maps, FengMap, Mapbox, OpenStreetMap, IndoorAtlas, MappedIn, and ArcGIS. Considering factors for the tool selection are accessibility, money cost, development complexity, and performance. Table 1 shows the comparisons of the shortlisted tools. More details about the development tools are included in Appendix A – E.

Tool	Accessibility	Money Cost	Development Complexity	Performance
Google Maps	✓	Map: Free API: Pay As You Go	- Easy - Dependent to Google Maps Team	- Integrated with the map outside campus - Place to Place navigation - No Control to Map - No visualisation for indoor environment
FengMap	✓	Map: Fiexed Cost API: Free to Use	- Draw map from scratch - Well support for mapping and navigation	- Poorly integrated with the map outside campus - Point to Point navigation - Good Indoor Map Visualisation
Mapbox	✓	Expensive	- Draw map from scratch - Well support for mapping and navigation	- Integrated with the map outside campus - No visualisation for indoor environment

OpenStreetMap	✓	Free (Open Source)	- Draw map from scratch - Poor support for mapping - Required an additional navigation solution	- No Navigation - No visualisation for indoor environment (Floor plan)
IndoorAtlas	✓	Expensive	- Draw map from scratch - Well support for mapping and navigation	- Poorly integrated with the map outside campus - Point to Point Navigation - Good Indoor Map Visualisation
ArcGIS	✓	Expensive	- Draw map from scratch - Well support for mapping and navigation	- Poorly integrated with the map outside campus - Point to Point Navigation - Good Indoor Map Visualisation
MappedIn	**No Access in Hong Kong**	NA	NA	NA

Table 1 Comparisons of development tools for mapping and navigation

2.3.2.1 Google Maps

Google Maps services offer user-friendly interfaces and reasonable costs, making them easily accessible. Google Maps provides an indoor map for the Main Building, which saves effort for the project. Moreover, users are generally already familiar with the Google Maps interface. Implementing Google Maps services in the app would eliminate the need for users to acquaint themselves with a new map, resulting in a seamless user experience.

However, it should be noted that the completion of building navigation within the Main Building is a task that extends beyond the scope and timeline of this final-year project. Achieving this would require providing additional information to the Google Maps team and collaborating closely with them. Consequently, considering the limitations and constraints of the project, Google Maps may not be the optimal solution for fulfilling the project requirements.

2.3.2.2 FengMap, Mapbox, IndoorAtlas, and ArcGIS

FengMap, Mapbox, IndoorAtlas, and ArcGIS are notable providers of specialized services for indoor map development and navigation. These platforms offer robust support for creating visually appealing indoor maps, optimizing paths, and incorporating features like floor selectors.

However, it should be noted that building indoor maps from scratch using these tools can be time-consuming, requiring significant development efforts. Another aspect to consider is the cost associated with these development tools, which unfortunately exceeds the project's allocated budget. While all four platforms provide similar services, FengMap stands out as the most cost-effective option among them.

2.3.2.3 OpenStreetMap

OpenStreetMap is an open-source resource that can be used to develop and view indoor maps. However, it provides the least support for development, and it must work with various additional tools to provide mapping and navigation functions, rendering an extremely high development complexity.

2.3.2.4 Implementation

The shortlisted development tools exhibit varying performance in outdoor and indoor scenarios. To address this, a hybrid approach is proposed, allowing users to seamlessly switch between an outdoor map and an indoor map. For the outdoor map, Google Maps is chosen due to its extensive maturity and widespread user familiarity. On the other hand, for the indoor map, FengMap is selected as the most cost-effective option among specialized indoor mapping tools.

However, it should be noted that the allocated budget remains insufficient to develop the entire indoor map for the Main Building. Consequently, only 2 floors will be included in the map, and there may be limitations in achieving an accurate scale due to budget constraints. Nevertheless, it is important to highlight that the scale of the map does not impact the visualization or other essential features provided by the map.

By combining Google Maps for the outdoor map and FengMap for the indoor map, the proposed hybrid approach aims to strike a balance between user familiarity, functionality, and cost-effectiveness. While limitations exist, this approach optimizes the available resources to provide users with a comprehensive navigation experience encompassing both outdoor and limited indoor areas of the Main Building.

2.4 Backend Service

This section discusses the development framework and the transmission technology selected for the backend service of the project. This section also describes the implementation of the backend service.

2.4.1 Server Development

Python is chosen for server development when utilizing machine. Python has a vast array of libraries and frameworks specifically designed for machine learning, making it easier to implement and deploy machine learning models on the server. Also, Python's versatility allows for seamless integration with other technologies commonly used in server development, such as web frameworks like Django or Flask. These frameworks enable the creation of robust and scalable server applications with minimal effort.

2.4.2 Selection of Transmission Technology: WebSocket

WebSocket is the chosen communication protocol for the project. WebSocket provides a persistent, full-duplex communication channel over a single TCP connection. This enables real-time, bidirectional communication between clients and the server. WebSocket has low latency and eliminates the need for frequent HTTP request-response cycles, resulting in faster and more efficient communication. WebSocket also offers enhanced scalability and resource utilization. Its event-driven model reduces unnecessary requests and server load, allowing servers to handle a larger number of concurrent connections with lower resource consumption. Furthermore, WebSocket supports cross-origin communication, enabling clients from different domains to establish secure connections and exchange data.

Socket.IO is used to establish the WebSocket connection. Socket.IO simplifies the implementation of real-time, bidirectional communication using WebSocket. It offers compatibility with various browsers and environments, automatic reconnection, and event-based messaging, which align with the project requirement.

2.4.3 Backend Service Implementation

The WebSocket connection is initiated upon launching the app. During active camera usage by users to scan their surroundings, short video clips captured by the phone camera are transmitted to the server in base64 format. Upon receipt of these video messages, the server proceeds to process the encoded data strings and fit them into the model. Upon deriving the output of the model, denoting the detected location, the server transmits the result back to the client through the existing WebSocket connection. The resulting message encapsulates a unique message ID, the success state, the coordinates, the corresponding location name, building name, and floor number.

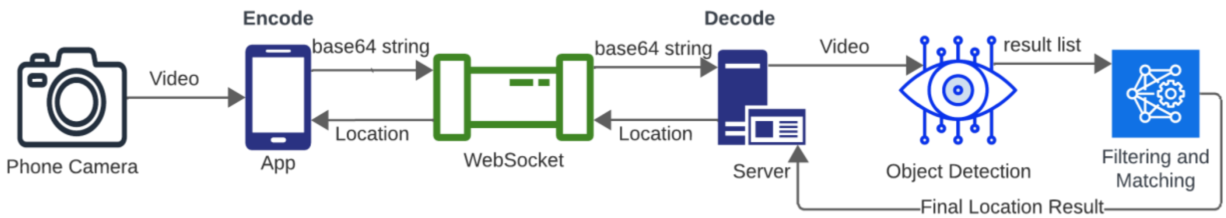


Figure 6 Flow chart for the flow of captured videos to the model

2.5 Model Design

This section explains the decision made for the model structure, followed by the selection of model, the implementation of the two layers, and the filtering score adjustment mechanism.

2.5.1 Model Selection and Structure Design

Three types of algorithms have been shortlisted for the computer vision model: image similarity algorithms, image classification algorithms, and object detection algorithms.

Image similarity algorithms are primarily designed to measure the similarity or dissimilarity between two images. They are useful for tasks such as image retrieval or content-based image searching. However, in the context of a vision positioning system, image similarity algorithms alone may not provide sufficient information to accurately determine the location of a user within a complex environment. They lack the ability to identify specific objects or landmarks within an image, which is crucial for precise positioning.

Image classification algorithms, on the other hand, are focused on assigning a predefined label or category to an input image. While they are effective in recognizing and classifying objects within images, they do not provide detailed information about the spatial location or precise boundaries of those objects. This limitation makes image classification algorithms less suitable for the task of accurately positioning users within a physical environment.

Object detection algorithms, on the contrary, offer a comprehensive solution for vision positioning systems. These algorithms are designed to not only recognize and classify objects within an image but also precisely locate and delineate their boundaries. By identifying and localizing specific objects or landmarks in real-time, object detection algorithms can provide the necessary information for accurate positioning and navigation within a complex environment.

The task of detecting buildings, markers, and other features in the images is assigned to the object detection layer. However, the initial implementation of a single-layer model did not yield optimal accuracy results. Therefore, in order to improve the accuracy of detection, a filtering and matching layer has been incorporated. This additional layer filters out objects that have relatively low confidence scores and matches them to coordinates based on detection patterns. By introducing this filtering and matching layer, the overall accuracy of the object detection process is enhanced.

The model selection and implementations of the two layers are discussed in Section 2.5.2 Object Detection Model) and Section 2.5.3 Filtering and Matching Layer), respectively.

2.5.2 Object Detection Model

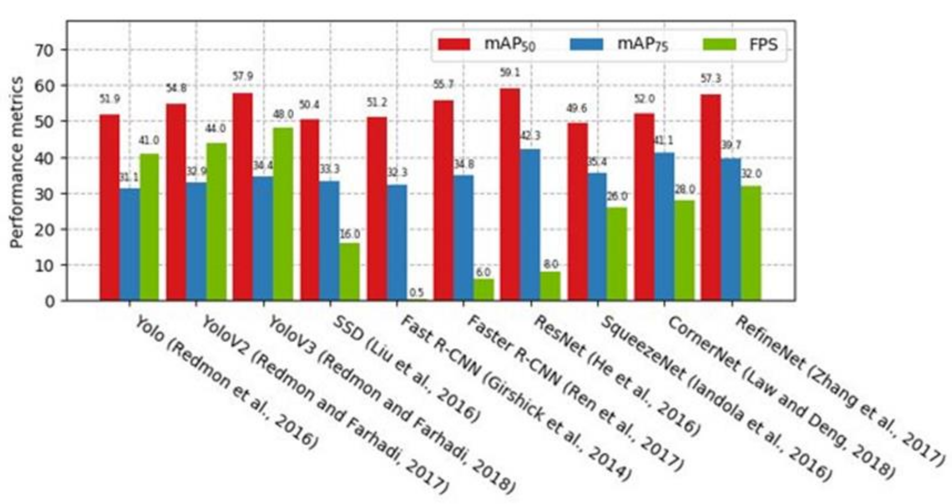


Figure 7 Performance metrics chart for different object detection models

A variety of algorithms for object detection have been made available to the public. In Figure 7, Sorin [7] conducted a performance comparison of object detection and recognition, revealing that You-only-look-once (YOLO) algorithms demonstrate superior performance in terms of high frames-per-second (FPS) rates. YOLO, as a single-stage detector, exhibits relatively lower complexity compared to second-stage detectors such as RefineNet and CornerNet. This characteristic enables YOLO to deliver faster detection outcomes, making it well-suited for real-time detection. This aligns with the requirements of our project, which aims to provide real-time localization and navigation assistance.

Therefore, YOLO has been chosen as the preferred algorithm for implementing the object detection layer of our model.

2.5.2.1 YOLO

YOLO is a real-time object detection model that utilizes a single neural network to directly predict class labels. This unique approach allows the model to swiftly recognize multiple objects within an image. The real-time nature of YOLO facilitates rapid navigation, reducing the computation time required for determining the location result. This computational efficiency serves as a strong justification for selecting YOLO, as it guarantees a fast detection and consequently a responsive user experience during navigation tasks.

In the project, the latest version of YOLO, specifically YOLOv8 provided by Ultralytics, is used as the object detection model. Ultralytics is a software company that shares open-source YOLO models with the public. Their application interface (API) offers extensive and efficient functionality, empowering users to build customized object detection models. The availability of guidelines and online support contributes to the broad community of YOLO users. Furthermore, according to official documents released by Ultralytics, YOLOv8 has low hardware requirements, with a minimum of 8GB of memory on a GPU. This favorable hardware requirement makes YOLOv8 well-suited for implementation within the scope and limitation of the project.

2.5.2.2 Optimizer

API provided by Ultralytics allows users to select suitable optimizer algorithms, such as Stochastic Gradient Descent (SGD), Adaptive Momentum Estimation (Adam), Adam with Infinity Norm (Adamax), Adam with Weight Decay (AdamW), Nesterov-accelerated Adam (Nadam), Rectified Adam and Root-mean-square Propagation (RMSprop). To evaluate the effect of the corresponding optimizer algorithm, testing with the same hyperparameters (listed in Appendix F) and the same dataset is used in the model training process with 300 epochs.

An experiment has been conducted to investigate the impact of optimizer algorithms on the accuracy of object detection during model training. Table 2 below shows the overall class accuracy (Precision) using different optimizer algorithms.

Optimizer	Precision
SGD	0.965
AdamW	0.958
Adamax	0.956
RAdam	0.944
NAdam	0.942
Adam	0.934
RMSProp	0.517

Table 2 Overall class precision for YOLOv8 training with different optimizer algorithms

Among all the optimizer algorithms tested, RMSProp exhibits the poorest overall class detection accuracy. With a precision score of 0.517, the model trained using RMSProp under the given conditions falls short in providing accurate and efficient object detection. On the other hand, the remaining optimizer algorithms perform better in terms of overall class accuracy, ranging from 0.934 to 0.965. Notably, SGD outperforms the other optimizer algorithms and achieves the highest accuracy. Appendix G provides a detailed evaluation for different optimizer algorithms.

2.5.3 Filtering and Matching Layer

To minimize the occurrence of high detection error rates and gather comprehensive environmental information, videos instead of individual images are captured and passed to the object detection model. The model processes the video frame by frame, resulting in multiple object detections with varying confidence scores. The real-time detection capability of YOLO enables swift predictions. However, this speed may lead to repetitive and misleading detections, primarily due to low confidence scores. Additionally, solely identifying objects does not provide precise location information, necessitating further adjustments to enhance accuracy and reliability.

To address these challenges, an additional layer is incorporated into the system architecture. The objects detected in the first layer serve as location anchors that give general idea of the user locations. The additional layer serves to filter out detection results with lower confidence scores and establish correlations between the identified object patterns and their corresponding locations. The filtering layer effectively screens out detections with low confidence scores. Subsequently, the filtered detection results undergo processing to compare them against predefined patterns, aiming to identify potential matches. By implementing this approach, the scale of detection results is reduced, significantly improving detection precision and minimizing misleading results.

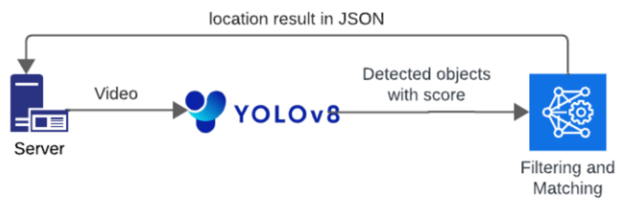


Figure 8 2-layer structure of the detection model

2.5.4 Filtering Score Adjustment

As discussed in Section 2.5.3, the implementation of a filtering layer aims to remove results with low confidence scores. However, there is a potential drawback in mistakenly blocking correct detections which have low confidence scores due to external factors, such as lower image quality in phone cameras or detection during night-time conditions. To address this issue, a mechanism is introduced to dynamically adjust the filtering score based on the failure time, where a lack of valid results from the filtering and matching layer is considered as a failure.

Upon receiving each video, the algorithm checks the failure time of the corresponding detection request. If there have been multiple instances of failure, the filtering score is gradually decreased, allowing results with slightly lower confidence scores to pass through the filter. This adjustment acknowledges that repeated failures may indicate challenging or unfavourable conditions for detection. Once a successful detection occurs, the failure time is reset, ensuring the subsequent filtering process operates based on the most recent detection outcomes. This adaptive mechanism enhances the system's resilience to external factors and improves the overall accuracy and effectiveness of the filtering layer.

2.6 Data Flow

In order to ensure high-quality data for training the CV model, each data sample needs to undergo four essential phases: data collection, data processing, data labeling, and data augmentation, prior to being utilized in the model training process.

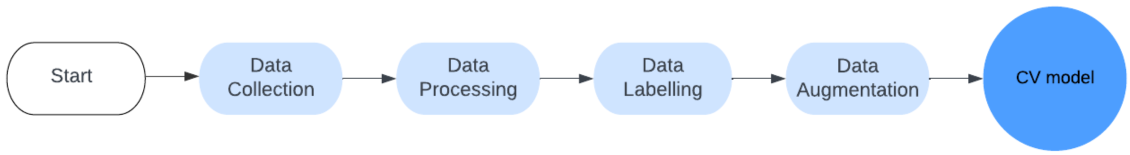


Figure 9 Demonstration of the 4 phases data workflow

2.6.1 Data Collection

In order to gather data for the project, multiple sources were considered, including online sources, past research, internal sources, and primary data sources. Upon evaluation, it was determined that the image quality from online sources did not meet the project's requirements. Additionally, access to up-to-date information from past research internal sources was limited, making it challenging to obtain current data. As a result, the primary data source was identified as the most suitable option, providing better control over both the quantity and quality of the images.

During the initial phase of data collection, images were manually captured one by one using phone cameras. However, it became evident that this method was inefficient After conducting the data experiment (Section 4.2 Inefficiency of initial Data Collection). It revealed the need for a more streamlined data collection method.

To enhance the effectiveness of data collection, a more efficient method was proposed and implemented. Videos were recorded, and images were extracted frame by frame. This approach proved to be highly efficient, enabling the rapid generation of hundreds of images and significantly reducing the overall time required for data collection.

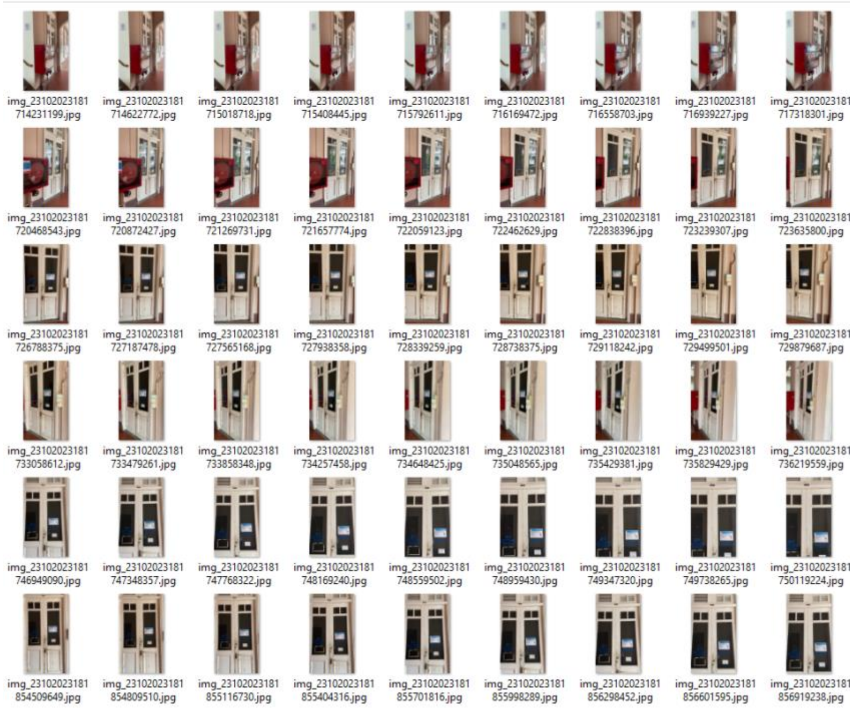


Figure 10 Examples of training dataset images

2.6.2 Data Processing

The data processing phase encompasses several essential steps, including data cleaning, image compression, and image formatting. Data cleaning involves eliminating any noise, outliers, or artifacts present in the acquired data to ensure its quality and integrity. Image compression techniques are applied to reduce storage and computational requirements while maintaining an acceptable level of image quality. Additionally, image formatting is performed to standardize the images, enabling seamless integration into the model and ensuring consistent results throughout the training process.

2.6.3 Data Labelling

The process of data labeling typically demands significant human effort and time. To expedite the development process, the decision was made to utilize RoboFlow, a contemporary data workflow management tool [8], for labeling the data.

This tool offers the capability to annotate images and provides a convenient overview of the annotated dataset, allowing for updates whenever necessary. By incorporating RoboFlow into the data labeling process, the annotation workflow is streamlined, resulting in improved efficiency and ensuring the quality and consistency of the labeled data. The integration of RoboFlow into the data labeling process also ensures consistency and maintains the quality of the labeled data. By utilizing a centralized tool, it becomes easier to enforce labeling standards and guidelines, reducing the risk of inconsistent or erroneous annotations.

2.6.4 Data Augmentation

Data augmentation is utilized to enhance the diversity and robustness of the training data. By implementing transformations such as cropping, flipping, rotation, scaling, and adjusting brightness, the augmented dataset introduces variations that closely resemble real-world conditions.

The primary objective of data augmentation is to improve the models' capacity to generalize and effectively classify objects in diverse scenarios. This technique plays a vital role in enhancing the overall reliability and robustness of the navigation system.

By introducing these variations through data augmentation, the models become better equipped to handle various environmental factors and accurately interpret objects in different contexts. This, in turn, leads to improved performance and increased confidence in the navigation system's reliability, further bolstering its robustness.

2.7 Database Management System

A database plays a crucial role in this project. A database enables efficient data manipulation, retrieval, and analysis, supporting the functionality and operations of the system. Firebase is selected as the database management system of the project. It serves as an excellent choice for storing spatial data and other relevant information for both the client side and server.

One key advantage of utilizing Firebase is its real-time database feature. This ensures that users accessing the application from different devices or platforms can access the most up-to-date information without delay. This real-time functionality is particularly advantageous for the usage of dynamic spatial data, as it allows for seamless synchronization across various devices.

Firebase also offers strong data security measures. It provides built-in authentication and authorization services, allowing developers to easily manage user access and permissions. By leveraging Firebase's security features, developers can ensure that sensitive data are securely stored, accessed only by authorized users, and protected against unauthorized access.

Furthermore, Firebase's scalability and reliability make it a suitable choice for the project that may experience scope expansion in the future. It is a cloud-based platform, which allows handling large amounts of data without compromising performance. Firebase's infrastructure enables scaling dynamically, ensuring that the application remains responsive even as the user base grows.

The implementation of Firebase for storing spatial data and other relevant information involves utilizing Firefoo for data manipulation. Developers can establish a secure and efficient connection between the application and the Firebase platform. This enables seamless data synchronization, real-time updates, and efficient storage and retrieval of spatial data.

3 Results

This chapter provides an overview of the final results in data collection, the model, the implementation of the mobile application, the development of the indoor map, a comparative analysis between the selected model and GPT-4 (Generative Pre-trained Transformer 4), and some of the project findings.

3.1 Data Collection

After recording videos at 42 different locations within the HKU main building, a large number of images were extracted from the videos. To ensure high-quality data, images with low quality were filtered out, and the remaining images were labeled with corresponding location tags, such as "room201," "room202," "room203," and so on.

Initially, a dataset of 5978 images was generated from 2264 raw images by applying data augmentation techniques, specifically rotation variation and brightness variation. Subsequently, the dataset expanded further to include 16886 images derived from 6474 raw images, covering a total of 56 distinct classes.

To facilitate model training and evaluation, the dataset was divided into three subsets: the training dataset, the validation dataset, and the testing dataset. The split ratio was 8:1:1, ensuring that each subset contained an appropriate representation of the overall dataset.

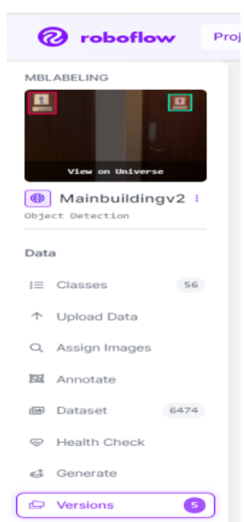


Figure 11 Information of Training set used

3.2 Model Tuning and Training

The section discusses the result of the model after model tuning and training.

3.2.1 Model Tuning

The Ultralytics YOLOv8 API is equipped with a well-designed interface that facilitates efficient and speedy model tuning. Considering the limitations of resources and time, the model tuning process involved 15 iterations, with each iteration comprising 50 epochs. The fitness of the model for each iteration is illustrated in Figure 12.

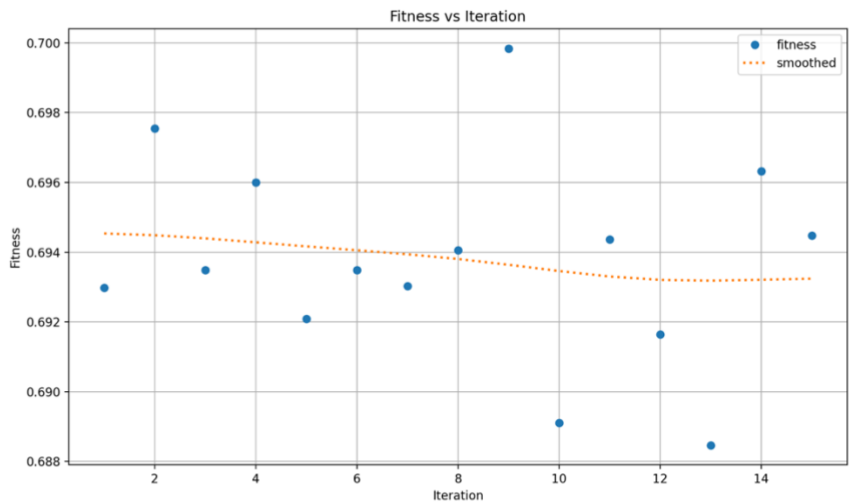


Figure 12 Graph of model fitness in each iteration during model tuning process

By analyzing the graph, it can be observed that the highest fitness score of 0.69984 was achieved at iteration 9. The corresponding hyperparameter values (refer to Appendix F) were extracted and subsequently utilized in the model training process.

3.2.2 Model Training

Using the tuned parameters and the expanded dataset, the YOLOv8 model was trained by the group. For the final training phase, the approach involved utilizing 1000 epochs and employing the SGD optimizer algorithm. The training process was conducted on a local desktop equipped with a graphics processing unit, which is NVIDIA GeForce RTX 3060.

Outlined below are the graphs and evaluation results obtained from the trained model.

3.2.2.1 Performance evaluation

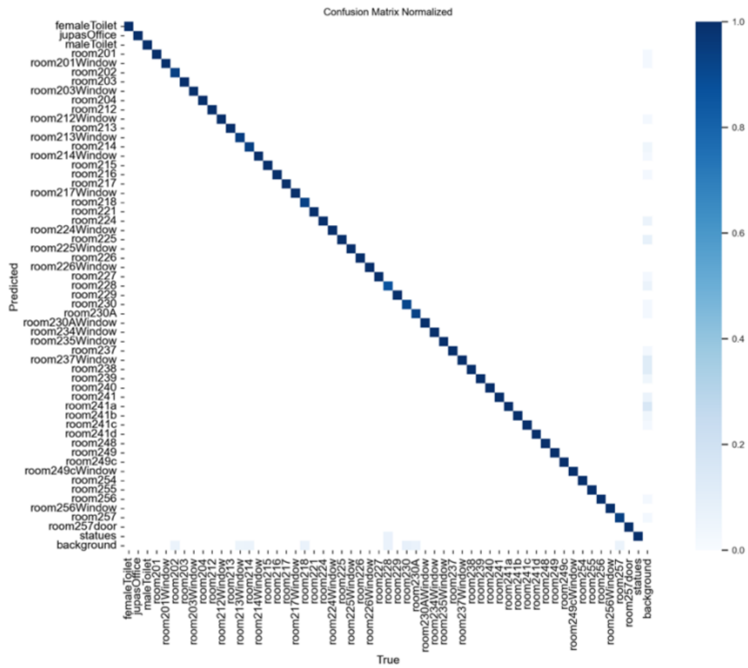


Figure 13 Confusion matrix of model after normalization

Figure 13 presents the precision of each class in the form of a confusion matrix. Following normalization, the precision values for each class demonstrate a consistently high performance, ranging from 0.85 to 1.0. This indicates that the model exhibits excellent precision across most classes, without any notable instances of low accuracy. Consequently, the model demonstrates accurate and well-balanced detection performance.

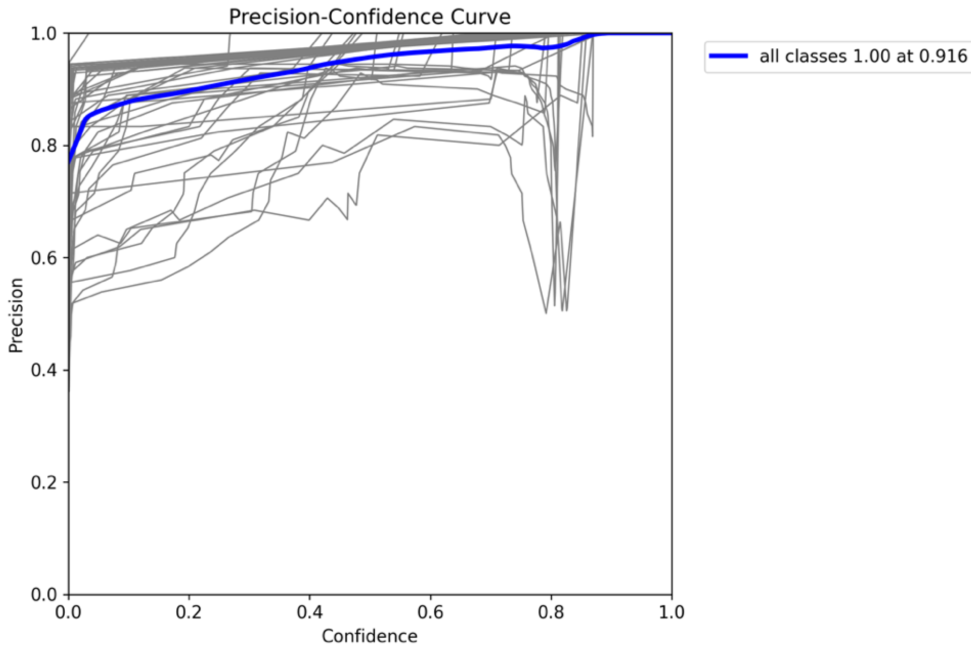


Figure 14 Precision-Confidence curve of trained model

In Figure 14, the relationship between precision and confidence is illustrated. The blue curve represents the overall class performance, while the grey curves represent the precision of individual classes. Upon analyzing the graph, it can be inferred that the model maintains a consistently high precision across various confidence levels.

At a confidence level of 0.916, the precision reaches a perfect score of 1.0. Moreover, there is a discernible increasing trend, indicating that the model's precision remains reliable and accurate across the entire range of confidence levels. This observation further solidifies the model's overall performance and its ability to deliver precise and dependable results.

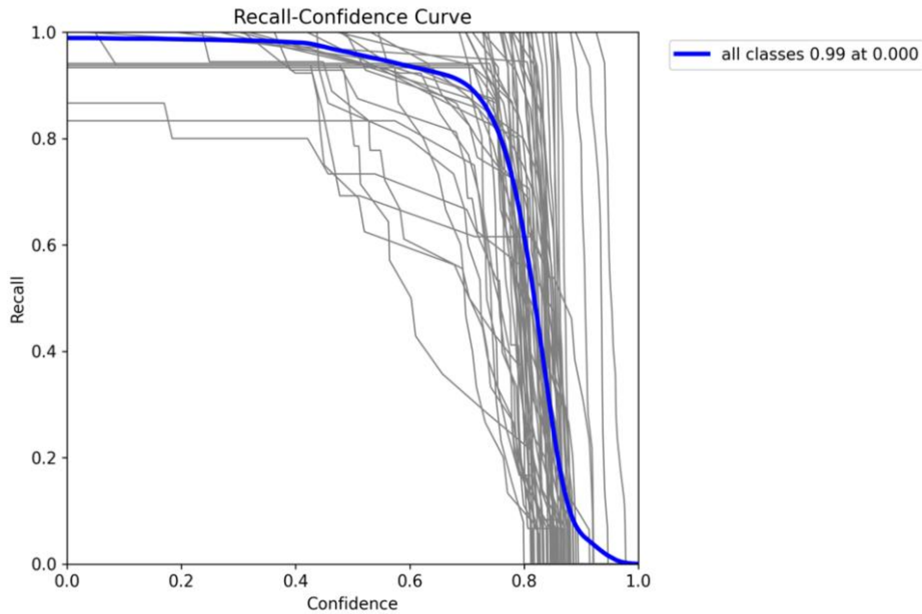


Figure 15 Recall-Confidence curve of trained model

Figure 15 illustrates the relationship between recall and confidence levels for the trained model. The curve representing recall remains relatively flat, suggesting consistent recall performance regardless of the confidence levels. This observation indicates that the model exhibits commendable performance in terms of recall. It reliably and consistently provides accurate predictions, ensuring the model's ability to recall relevant instances effectively.

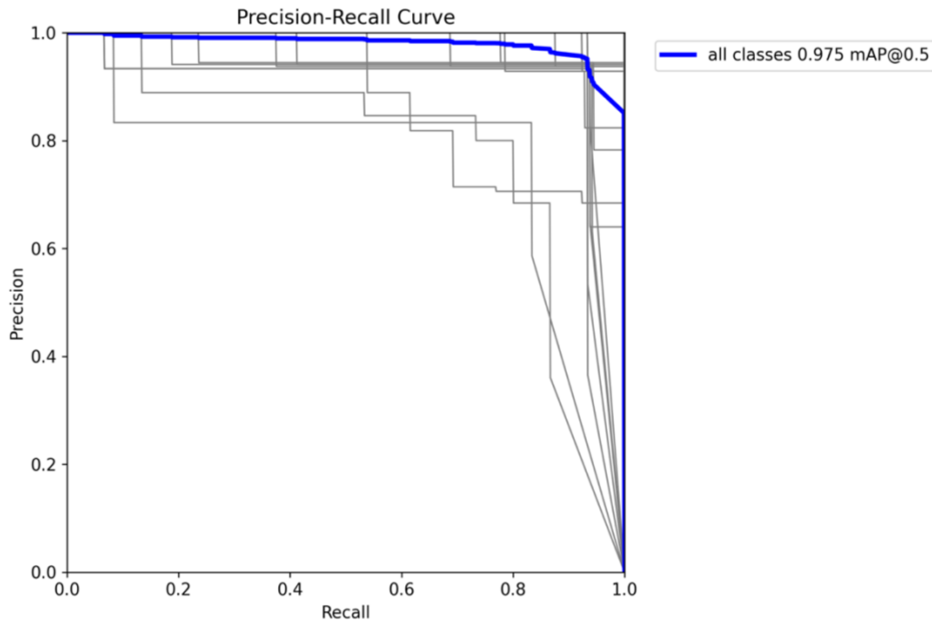


Figure 16 Precision-Recall curve of trained model

Figure 16 depicts the precision-recall trade-off for the model trained using the SGD optimizer algorithm. The precision-recall curve for the overall class demonstrates high precision and recall values across a majority of the threshold values. Additionally, the obtained mAP@0.5 value of 0.975 indicates exceptional performance in terms of the precision-recall trade-off.

The precision-recall curve for the model closely resembles that of an ideal detector, with consistently high precision and recall values at all confidence thresholds. This finding further emphasizes the model's ability to achieve both high precision and recall simultaneously, indicating its exceptional performance in balancing these two important evaluation metrics.

3.3 Mobile Application

The app, named MapU, consists of 3 pages, the Camera page, the Map page, and the Info page, implementing vision-based positioning, path finding, and accessing campus information respectively. Figure 17 shows the app logo.

Figure 17 MapU app Logo

3.3.1 Camera Page

As shown in Figure 18 to Figure 20, the app accesses the phone camera for image capturing. If the connection to the server fails, an error message will be prompted, as demonstrated in Figure 18. Figure 19 shows the reminder message to the users if the location detection continues to fail. Upon receipt of the location result from the server, as shown in Figure 20, a message will be prompted to indicate the detected location. Users can choose to keep capturing or go to the "Map" page with the detected location. For the former option, the app would start a new round of detection. For the latter option, the UI would direct to the "Map" page and the detected location would be automatically placed in the "Current Location" field.

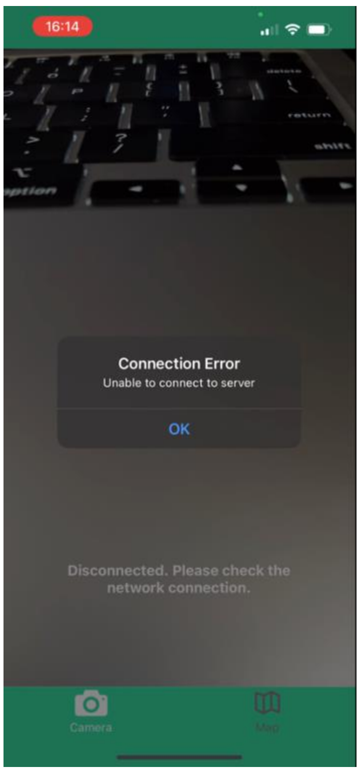


Figure 18 The "Camera" page – Connection error

Figure 19 The "Camera" page – Reminder message

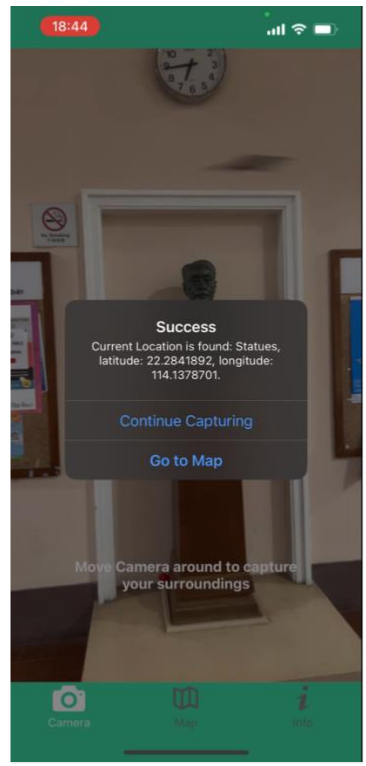


Figure 20 The "Camera" page – Successful detection

3.3.2 Map Page

As illustrated in Figure 21, the "Map" page allows users to input source location and destination. When users are inputting in the boxes, a suggestion list of venue names will be provided. Users have to choose from the suggestion list to finish inputting. This serves as a way of input validation. As the "Current Location" and the "Destination" are verified, corresponding location information would be displayed on the page, as shown in Figure 22.

Figure 21 The "Map" page – User Input

Figure 22 The "Map" page -- Display information

As users click the “Navigate” icon button, the app would validate the inputs. If both fields are valid, the path from the “Current Location” to the “Destination” would be optimized and displayed on the map. Figure 23 shows the path optimized for outdoors, which is obtained from Google Maps Routes API. Figure 24 demonstrates the indoor path generated by FengMap and accessed by the FengMap API. To switch between the outdoor map and the indoor map, users can click the button “View Indoor Map” / “View Outdoor Map” to seamlessly navigate to another map.

Figure 23 The "Map" page -- Outdoor Map Path Visualisation

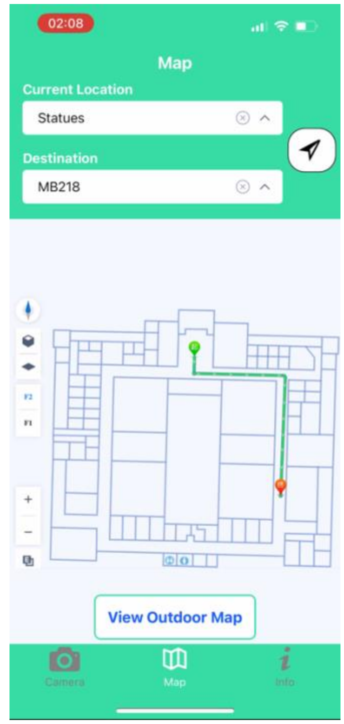


Figure 24 The "Map" page -- Indoor Map Path Visualisation

3.3.3 Info Page

Figure 25 and Figure 26 show the “Info” page that provides users with various information about the campus of HKU. Basic information about the campus, including locations of buildings, departments, facilities, and transport, can be accessed through this page. For example, users can look for the facility information and location on the “Facilities” list as shown in Figure 25. Users can also check for department or building information as illustrated in Figure 26. Additional information, such as the driver’s route and transport, is available as well.

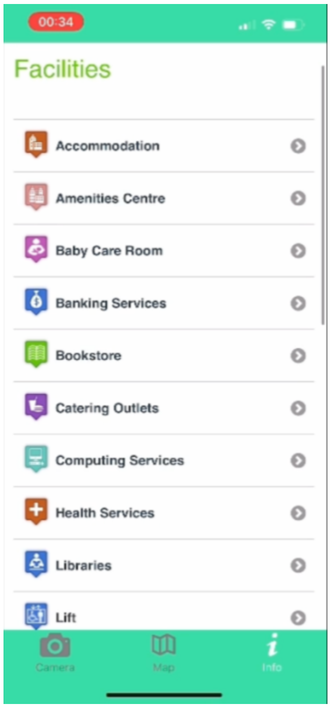


Figure 25 The "Info" page – Facilities list

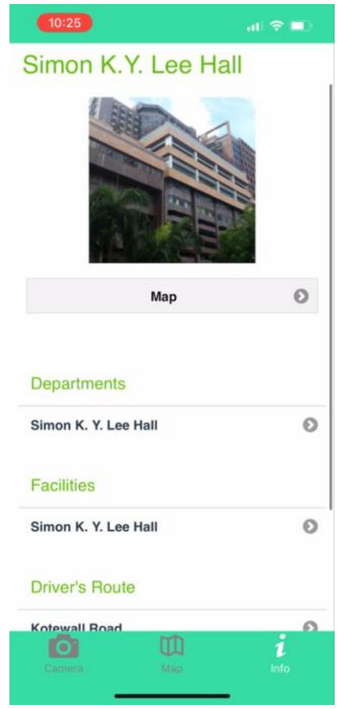


Figure 26 The "Info" page -- Building Information

3.4 Campus Map

The indoor map for the main building has been structured. A floor selector is provided for selecting the floor to display. Routes are added so that indoor navigation powered by FengMap can be used. The remaining works are labelling room names, optimizing layouts, and integrating the mapping and navigation service into the app.

Figure 27 Demonstration of an indoor map built using FengMap (2/F, Main Building)

3.5 User Testing

To evaluate the performance of the indoor positioning functionality of the application and gather valuable insights for system enhancement, user testing was conducted. The project group invited two individuals, referred to as A and B, who had no prior knowledge of the detection mechanism. This approach simulated real user scenarios and provided unbiased feedback for evaluating the system's performance.

3.5.1 Testing Details

Environment:	2/F, Main Building
Number of participants:	2
Testing device:	Participant's own mobile phone
Time:	
	- Individual A: 1700 - 1730
	- Individual B: 1830 - 1900

Test Procedures:

1. Participant, accompanied by the project group, walked around the 2/F of the Main Building, randomly detected the location using the app
2. Then the project group compared the detection result with the actual location, and recorded the test data

3.5.2 Testing Results

3.5.2.1 Result Table

Individual	No. of detection	Max.	Min.	Mean	Hit Rate
A	21	21	4	10	95.24%
B	33	26	5	11.1	87.9%
Total	54	26	4	10.55	91.56%

Table 3 User Testing Results

3.5.2.2 Other Observations

3.5.2.2.1 Accuracy drops during nights

During the user testing, it was observed that the accuracy of the indoor positioning system decreased during nighttime. This observation can be reflected by the testing accuracy difference between A and B, which are 95.24% and 87.9% respectively. This decline in accuracy could be attributed to two possible reasons. Firstly, the image quality was significantly affected due to the lack of adequate lighting during night-time testing sessions. This diminished image quality might have made it more challenging for the system to precisely detect and recognize the surroundings. Secondly, there may have been differences in darkness levels between the training data set and the actual images captured during the nighttime testing. These variations in lighting conditions might have introduced discrepancies between the expected and actual image patterns, leading to a decrease in accuracy.

3.5.2.2.2 Fewer time is needed when users get familiar

Both participants, A and B, demonstrated a decreasing trend in the time required to achieve successful location detection as the testing progressed. This observation indicated that participants became more familiar with the location detection process and started to understand the techniques to obtain quicker results. With repeated use and experience, they likely developed a better understanding of the system's functionality and optimized their approach to obtain accurate location detections more efficiently. This finding suggests that user familiarity and experience play a significant role in improving the overall efficiency and effectiveness of the indoor positioning system.

3.5.3 Result Evaluation

The evaluation of the system's performance yielded several important findings. The time required for location detection ranged from 4 seconds to 26 seconds, indicating that the system is capable of providing relatively fast results when the captured videos contain sufficient features for accurate detection. Even if users initially do not capture informative clips, the application offers clear instructions to guide them towards acquiring a location result within half a minute. The average detection time falls within a reasonable range, demonstrating the system's ability to provide timely results. Additionally, it was observed that experienced users tend to achieve faster detection times, highlighting the impact of user familiarity and proficiency on the system's performance.

The average hit rate of 91.56% from 54 instances of detection is considered acceptable. However, it should be noted that the original approach of using data augmentation to handle darkness differences did not effectively address all situations for vision positioning at night. As a result, the system occasionally generates false location results under such conditions. To improve the vision positioning system further, it is crucial to expand the training data set to include images captured at night. By incorporating nighttime images into the training data, the system can learn to better handle low-light scenarios and reduce false detections. This step is essential for enhancing the accuracy and reliability of the vision positioning system in various lighting conditions.

3.6 GPT-4 Analysis

Given the widespread popularity of GPT-4 and its recent announcement on September 25, 2023, stating its compatibility with image inputs [9], it becomes an intriguing candidate for comparison with the CV model of the project. To evaluate the image capabilities of the GPT-4 model, a few experiments were conducted. Subsequently, a comprehensive comparative analysis was performed between the GPT-4 model and the model of the project.

3.6.1 Traditional train-test experiment

The experimental setup closely resembles the typical procedure for training and testing a supervised machine learning model. It involves exposing the model to a significant amount of training data, consisting of pairs of input images and their corresponding expected outputs. During the training phase, the model is trained on groups of images from various locations. In the subsequent testing phase, a single image is presented to the model, accompanied by a question regarding its specific location.

The results of the experiment yield intriguing observations. The GPT-4 model exhibits the ability to recognize the distinctive features present in the given test image. However, it faces challenges in explicitly mentioning the exact location of the image. For instance, it can accurately describe elements like a door sign within a room, which can serve as valuable clues for inferring the location. However, without additional guidance or explicit instructions, the model cannot provide a precise answer regarding the location.

This indicates that while GPT-4 demonstrates proficiency in understanding and describing visual features, it still requires supplementary information or context to accurately pinpoint the specific location of an image.

3.6.2 Zero-shot Experiment

The objective of the experiment is to assess the image capabilities of GPT-4 by challenging its ability to determine the location of an unseen image without any prior training. Each model input consists of a test image and a query requesting information about its location. The model is subjected to testing across a sufficient number of locations to evaluate its performance comprehensively.

The experimental results bear resemblance to those obtained from traditional train-test experiments, as described in Section 3.6.1. Notably, the model successfully captures features present in the image but fails to provide an exact location answer. These strikingly similar outcomes lead us to question the effectiveness of training for GPT-4. It appears that the training process does not significantly enhance the model's ability to respond accurately during testing. Consequently, we argue that zero-shot learning, with its notion of achieving high-quality results with minimal training effort, represents a more favorable approach.

This observation prompts us to reconsider the training paradigm for GPT-4 and explore alternative methodologies that leverage zero-shot learning to achieve improved performance in similar image recognition tasks.

3.6.3 Zero-shot Experiment with Location Domain

To address the limitation observed in the previous zero-shot experiment, namely the inability to explicitly mention the location of a given image, a modification is made to incorporate location options as part of the model input query. Consequently, the modified model input query consists of three components: the location query request, a list of location options, and the query image, as illustrated in Figure 28.

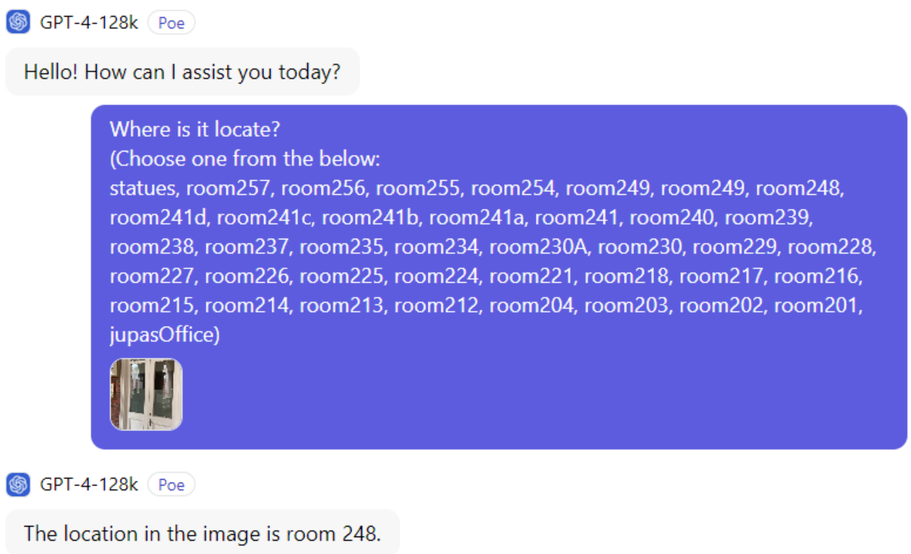


Figure 28 room 248 zero-shot experiment w/ location domain

The response from GPT-4 following this modification proves to be quite satisfactory. It now demonstrates the ability to accurately deduce the specific location of the given image. For instance, as depicted in Figure Figure 28, when presented with an image of room 248 under the modified query format, GPT-4 explicitly mentions the location as room 248 in its response. Furthermore, numerous locations are tested, and in most cases, GPT-4 successfully predicts the correct location. Notably, GPT-4 exhibits proficiency in recognizing non-textual features, as evidenced by its identification of statues, as shown in Figure 29. This highlights the robust object detection capabilities of GPT-4.

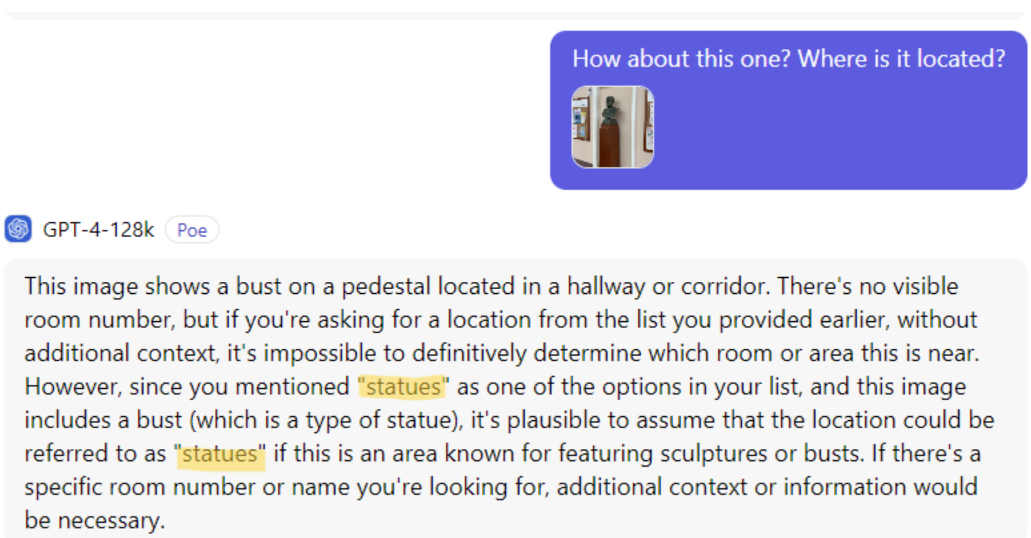


Figure 29 The statues zero-shot experiment w/ location domain

However, it is worth noting that GPT-4 may occasionally produce incorrect predictions. For example, in the case of predicting room 227, GPT-4 accurately predicts the location for the two images captured from the left side of room 227. However, it misleadingly predicts two images taken from the right side of room 227 as "room 202" or "room 202A," as depicted in Figure 30 and Figure 31. This suggests that images taken from certain angles or perspectives of a location can significantly impact the location prediction accuracy of GPT-4.

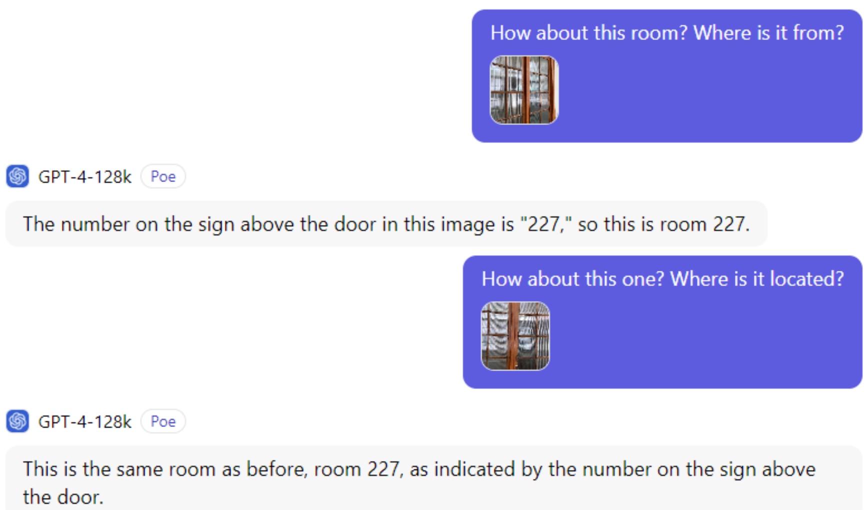


Figure 30 room 227 (left side) zero-shot experiment with location domain

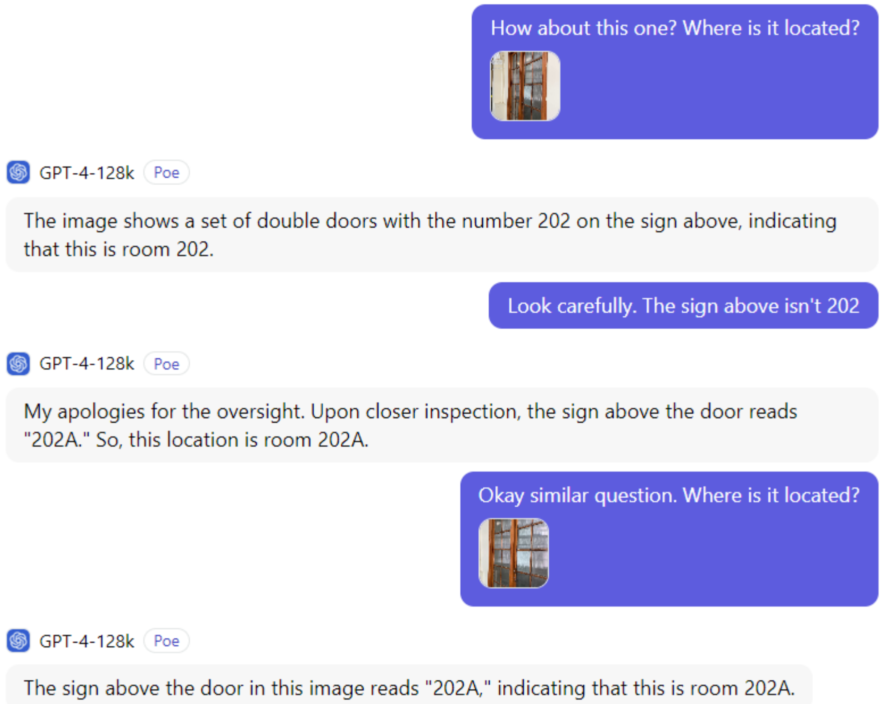


Figure 31 room 227 (right side) zero-shot experiment with location domain

3.6.4. GPT-4 experiments conclusion

Overall, GPT-4 exhibits exceptional performance in image recognition, particularly in mapping images to their corresponding locations. This achievement is noteworthy considering the minimal preparation effort required and the broad scope of applicability. Initially, a traditional train-test experiment was conducted, revealing GPT-4's impressive ability to recognize image features accurately. Subsequently, a zero-shot experiment was implemented, demonstrating similar results while significantly reducing the preparation effort. Finally, the zero-shot experiment was further enhanced by incorporating the location domain. With this additional guidance, the GPT-4 model consistently generates desirable and reliable results.

3.6.5 GPT-4 vs YOLO

As shown in Table 4, both YOLO and GPT-4 demonstrate successful capture of image features. However, there are notable differences in the process.

	Capture image feature?	Output correct location?	Training effort	Scalability
YOLO	✓	✓	Huge (data labelling)	✗
GPT4 w/o location domain	✓	✗	No	✓
GPT4 w/ location domain	✓	✓	No / A bit (providing location domain)	✗

Table 4 Comparisons between GPT-4 and YOLO

3.6.5.1 Feature Preparation

YOLO requires manual selection and labeling of image features before being fed into the model. This approach is cumbersome, demanding significant human intelligence and time. On the other hand, GPT-4 automates the process of capturing image features and identifies a broader range of features. However, the selection of features is non-deterministic, similar to its response to open-ended questions.

3.6.5.2 Detection Quality

Regarding the model output, both YOLO and GPT-4 with the location domain are capable of providing the correct location answer. In contrast, GPT-4 without the location domain fails to explicitly output the correct location. This indicates that GPT-4's output can be shaped into an ideal format when provided with appropriate guidance.

3.6.5.3 Training Effort

In terms of training effort, YOLO requires a significant investment. It involves data labeling and model training, which can be time-consuming and resource-intensive. In contrast, GPT-4 can accomplish the same task using zero-shot learning, eliminating the need for extensive training.

3.6.5.4 Scalability

When considering scalability, both YOLO and GPT-4 with the location domain face limitations. YOLO requires extensive labeling effort, which becomes increasingly challenging as the dataset grows. Similarly, GPT-4 with the location domain necessitates a pre-defined list of location options, which can restrict its scalability. The only scalable model is GPT-4 without the location domain, as it benefits from zero-shot learning, allowing it to adapt to new tasks without additional training efforts.

3.7 Project Findings

In this section, we will discuss additional findings that emerged from the project. Specifically, we will delve into two key areas: the selection of objects for labelling (Section 3.7.1) and the emergence of new technologies (Section 3.7.2). These findings shed light on important aspects related to the project's objectives and provide valuable insights for future research and development in the field of vision positioning and location mapping.

3.7.1 Selection of Object for Labelling

The selection of objects to be labeled plays a crucial role in determining the performance of the object detection model. The case of the 4 fire hose wheels on each corner of the 2/F of the Main Building serves as a pertinent example. While these wheels may possess slight differences, they still appear similar. If these wheels are chosen as image features, it can lead to confusion within the model.

In essence, the selection of objects with significant uniqueness is of paramount importance. By choosing objects that possess distinct characteristics, we enable the CV model to differentiate between them accurately. This, in turn, enhances the model's ability to identify and classify different objects or locations with a higher level of confidence.

The process of selecting objects for labeling requires a high degree of human intelligence. It necessitates a thorough understanding of the specific task at hand, as well as a keen eye for identifying objects that exhibit noteworthy distinguishing features. When undertaking this process, it is crucial to consider factors such as shape, color, texture, and any other discernible attributes that can aid in the proper identification and differentiation of objects.

The quality and quantity of the selected objects also have a direct impact on the overall performance of the vision positioning system. A small number of well-chosen objects may yield satisfactory results, provided they possess clear and distinctive characteristics. Conversely, a larger set of poorly chosen objects can introduce confusion and ambiguity into the system, leading to a decrease in accuracy and reliability.

Achieving an optimal balance between quantity and quality is essential. Including a sufficiently diverse range of objects in the dataset ensures robustness and generalization of the CV model. However, it is equally important to avoid including objects that are too similar or exhibit minimal distinguishing features, as this can lead to misclassification or ambiguous results.

Furthermore, the selection process should also consider the practical aspects of labelling. It is crucial to strike a balance between the complexity of the labeling task and the available resources, such as time and human effort. Careful consideration should be given to selecting objects that provide meaningful information for the vision positioning system while optimizing the efficiency of the labelling process.

In summary, the correct selection of objects for labeling significantly impacts the performance of the CV model in vision positioning systems. Objects with significant uniqueness enable accurate differentiation and classification, enhancing the model's confidence in identifying various locations or objects. The quality and quantity of selected objects should be carefully balanced to ensure both accuracy and efficiency in the vision positioning system.

3.7.2 Emerging New Technologies

Continuous evaluation of new technologies and emerging advancements is crucial for enhancing a vision positioning system. The analysis of the GPT-4-Turbo model in Section 3.6 provided valuable insights into the development landscape of OpenAI. It was announced on September 25, 2023 [9], that the GPT-4 model now supports voice and image inputs, allowing the general public to access and contribute to its vision and voice capabilities.

However, it is important to note that the current version of the GPT-4 model does not support fine-tuning of its vision capabilities [10]. This limitation prevents us from specializing in the GPT-4 model specifically for the objectives of this project. While this may present a temporary constraint, it is encouraging to observe that OpenAI is actively developing various vision models in the preview phase, such as "gpt-4-turbo", "gpt-4-vision-preview", and "gpt-4-1106-vision-preview”, as shown in Figure 32.

gpt-4-vision-preview	GPT-4 model with the ability to understand images, in addition to all other GPT-4 Turbo capabilities. This is a preview model, we recommend developers to now use gpt-4-turbo which includes vision capabilities. Currently points to gpt-4-1106-vision-preview.	128,000 tokens	Up to Apr 2023
gpt-4-1106-vision-preview	GPT-4 model with the ability to understand images, in addition to all other GPT-4 Turbo capabilities. This is a preview model, we recommend developers to now use gpt-4-turbo which includes vision capabilities. Returns a maximum of 4,096 output tokens. Learn more.	128,000 tokens	Up to Apr 2023

Figure 32 Information of GPT4 preview versions

The research on GPT-4's vision capabilities is expected to undergo significant advancements shortly, given the trend of generative AIs. By leveraging the capabilities of GPT-4, the vision positioning system could benefit from enhanced accuracy, efficiency, and adaptability. The ability to carry out auto-data labelling would streamline the data preparation process, reducing the manual labeling effort and minimizing human error. Moreover, high-accuracy image positioning would enable precise localization and tracking of objects or locations, enhancing the overall performance and reliability of the vision positioning system.

It is crucial to continuously evaluate and monitor emerging technologies in the field of computer vision. As new advancements and breakthroughs occur, the vision positioning system can be updated and adapted to incorporate the latest developments. This iterative process allows for ongoing improvement, ensuring that the system remains at the forefront of technological advancements and delivers optimal results.

In conclusion, continuous evaluation of emerging technologies, such as the advancements in GPT-4's vision capabilities, is essential for enhancing a vision positioning system. As these technologies mature, they have the potential to revolutionize data labelling, image positioning, and overall system performance. By staying abreast of the latest developments and incorporating them into the system, we can ensure that the vision positioning system remains accurate, efficient, and adaptable to evolving requirements and challenges.

4. Limitations and Difficulties

This chapter delves into the dataset size overload issue identified in the early stages, data collection inefficiency problem, and the location matching issue, examining both the nature of the challenges and the responses devised to address them.

4.1 Dataset Size Overload

4.1.1 Problem of Dataset Size Overload

The large data size required for the project imposes significant challenges on its development. In particular, the capturing of images from various angles, heights, and times at the same node on the map adds to the complexity. Furthermore, the quest for higher accuracy necessitates a substantial number of images for model training. Consequently, the development of an accurate positioning model for the entire HKU campus may require an extensive dataset comprising millions of images. The magnitude of data involved imposes rigorous demands on computational power, storage capabilities, and human resources. Notably, these requirements surpass the scope of a typical final year project, warranting careful consideration and resource allocation.

4.1.2 Response to Dataset Size Overload: Limiting Scope

The primary mitigation strategy is to limit the scope of the project. Instead of encompassing the entire HKU campus, the project will concentrate on a specific section. The Main Building of HKU has been selected as the designated testing site for the project. By focusing on the Main Building, the project aims to achieve the goal of effectively navigating visitors within its premises. By adopting a focused approach and considering potential future expansions, the project aims to strike a balance between the limited resources available and the aspiration to deliver an effective navigation solution within the Main Building of HKU.

4.2 Inefficiency of initial Data Collection

4.2.1 Problem of Initial Data Collection

An experiment was conducted to examine the efficiency of the initial data flow design. Table 5 below shows the times used for different numbers of images in the 4 phases.

	No. of Image	Time used			
		Data Collection	Data Processing	Data Labelling	Data Augmentation
Set I	40	5	5	10	3
Set II	117	14	7	15	4
Coefficient	2.93	2.80	1.40	1.50	1.33

Table 5 Time used for 40 images and 117 images in the 4 data flow phases

The findings demonstrate that the time spent on the data collection phase exhibits the most substantial growth. The coefficients associated with the number of images and data collection, highlighted in Table 5, are closely aligned, suggesting that the time required for data collection gradually increases as the dataset size expands. Leveraging existing tools aids in mitigating the proportional growth of time in data processing, data labelling, and data augmentation. However, due to the manual image acquisition on an individual basis, a linear relationship emerges between the number of images and the time spent on data collection. The experiment highlights the inefficiency of the original data collection method, consequently diminishing the value of the vision-based approach.

4.2.2 Response to Inefficient Data Collection: Videos instead of Photos

The initial data collection process encountered inefficiency as images were gathered on an individual basis. However, given the importance of both image quality and quantity, it became evident that obtaining data from primary sources was necessary. Consequently, a new and improved method was implemented, which involved capturing videos and subsequently extracting frames from these videos. This approach ensures a more efficient and comprehensive collection of data for the project, allowing for a greater variety of images to be obtained while maintaining the desired level of quality.

Using the new data collection method, the time used for capturing training data gradually reduced. Table 6 shows the improved time used for data collection.

	No. of Image	Time used in Data Collection	Image per minute
Initial Method	117	14	8.36
New Method	165	3	55

Table 6 Time cost comparison of the initial and new data collection methods

4.3 Location Matching

4.3.1 Problem of Location Generalization

To identify locations using images, a model for object detection is utilized to identify distinctive features within the images. These detected objects serve as reference points (like anchors) to give a general indication of the user's location. However, this method has a drawback when it comes to directly predicting the precise location. The presence of multiple objects in the same frame can lead to misleading detection results that cannot be directly linked to the exact location.

4.3.2 Response to Location Matching: Additional layer

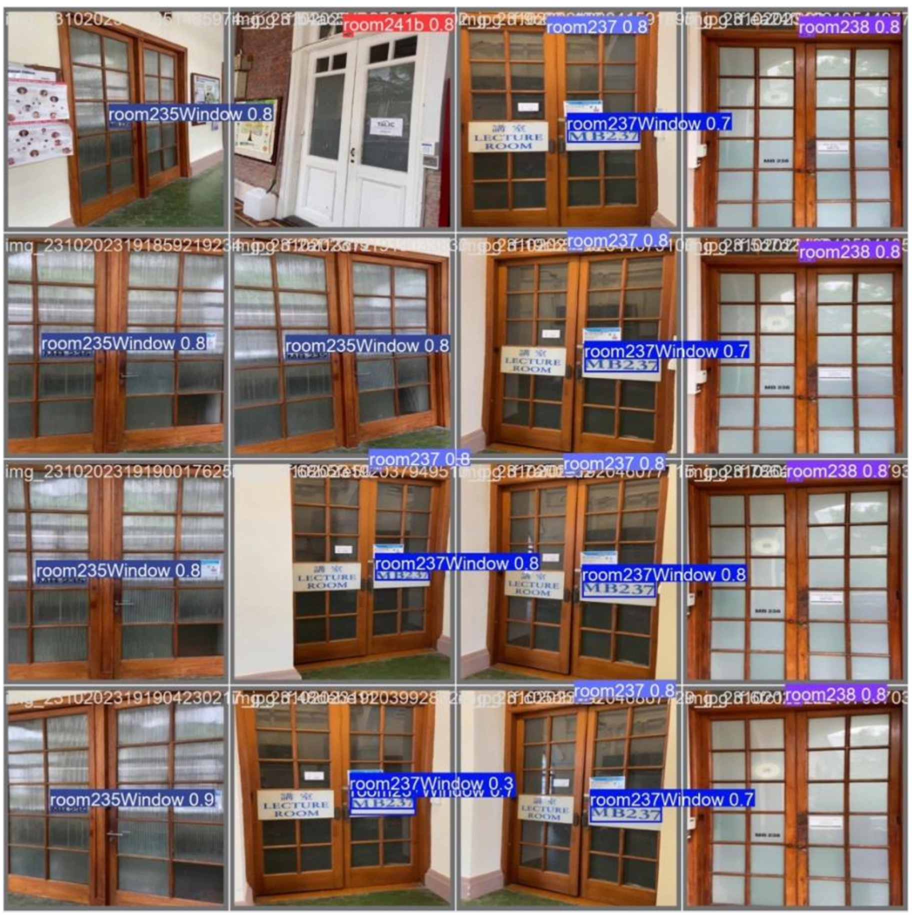


Figure 33 Detection results example of using trained model

To address this issue, one approach taken in the project involved implementing confidence score filtering on the object detection model. As depicted in Figure 33, the object detection model can detect multiple objects in the same image, each with varying confidence scores. By applying confidence score filtering, predictions with low confidence scores can be disregarded, reducing misleading detections and simplifying the understanding of the user's location. Another mitigation strategy involves developing a specialized function that incorporates multiple factors, including confidence scores, detected class names, and appearance in frames. This function consolidates the information about the user's location that is detected and generalizes it to provide an overview of their whereabouts.

In the project, a filtering and matching layer is implemented to compare the detected results with record patterns stored in a database. Once the results have undergone filtering, the program searches for a matching pattern, thereby narrowing down the potential location.

However, this method is not efficient and lacks the ability to provide additional information such as the distance between objects and precise longitude and latitude coordinates on a map.

4.3.3 Proposed Response to Location Matching: Extra technologies

With the aim of expanding the project's scope and enhancing its capabilities, the possibility of integrating a machine learning layer and camera calibration is being considered. These additional layers have the potential to refine the detection results and provide more precise location information.

One approach being explored involves categorizing the detected class IDs obtained from the YOLO model predictions. By collecting data and leveraging it to train an additional machine learning layer, the spatial relationships between the detected class IDs and their corresponding locations can be learned. This trained layer would act as a bridge between the output of the YOLO model and the desired precise location information.

In parallel, camera calibration can significantly contribute to improving detection precision. Accurately calibrating the camera parameters, such as intrinsic and extrinsic properties, enhances the accuracy and reliability of location estimation. The calibration process entails determining intrinsic properties like focal length and principal point, as well as extrinsic properties such as the position and orientation in the world coordinate system. Combining the calibration results with the object detection information enables the retrieval and display of users' exact locations on the map.

By integrating a machine learning layer and employing camera calibration techniques, the project can benefit from refined detection results and provide more accurate location information for enhanced performance.

4.4 Dependence of third-party indoor map solutions

4.4.1 Problem of Dependence of third-party indoor map solutions

The problem of dependence on third-party indoor map solutions becomes apparent when considering the limitations of the two main solutions explored in the project.

Firstly, Google Maps, while widely used for outdoor navigation, does not provide indoor navigation functionality for the HKU campus. This means that users can only navigate to buildings on the campus, but not to specific rooms or lecture venues within those buildings. Furthermore, the availability of indoor maps on Google Maps is limited, with only a few buildings on the campus having an indoor version available. This severely restricts the usefulness of Google Maps for the project's indoor positioning requirements.

Secondly, FengMap, a potential alternative, offers limited functionality in its free version. This limitation hampers the ability to fully leverage the potential of FengMap for indoor navigation within the campus. Additionally, the API patch provided for React developers does not keep up with version changes, making it challenging to integrate FengMap smoothly into the project's development workflow. These limitations make FengMap a less well-supported option, creating uncertainties and potential difficulties in its implementation.

Furthermore, exploring other indoor map solutions, such as Mapbox, IndoorAtlas, and ArcGIS, revealed that many of them are prohibitively expensive. The high costs associated with these solutions make them impractical for use within the project's scope and budget.

4.4.2 Proposed solutions to Dependence of third-party indoor map solutions

In this project, two different mapping services, Google Maps and FengMap, are combined to feature the indoor map and indoor navigation in the app. This approach met the goal of providing the primary functionality of the app. However, the reliance on third-party agents still leaves unignorable potential risks to ensuring the availability of the mapping functionality and providing seamless navigation experiences to the users.

While the issue of dependence on third-party indoor map solutions remains unresolved in the project due to various limitations, several potential solutions are proposed for future developments.

4.4.2.1 Self-Development of Campus Map

One proposed solution is to self-develop a campus map with the aid of some HKU-owned technology. This approach would involve creating a custom map specifically tailored to the campus's layout and indoor navigation requirements. A remarkable example of owned technology for developing a map is the 3D scanning reconstruction offered by ManiFold Tech Ltd. By developing the map in-house, the project team would have full control over the customization and implementation of the map, eliminating dependence on third-party providers.

However, it is important to acknowledge that the self-development of a campus map would require a significant allocation of human resources, including skilled developers and designers. Additionally, the process could incur substantial costs, including the acquisition of necessary tools, technologies, and data sources. Therefore, careful consideration of the project's budget and available resources would be crucial when evaluating this solution.

4.4.2.2 Cooperation with Google Maps

Another potential solution is to explore a collaboration with the Google Maps Team. By actively engaging with the Google Maps Team and providing them with the required information and motivation, it may be possible to obtain indoor maps and navigation functionality for the campus. Google Maps is a widely recognized and reliable third-party agent, offering a familiar interface for users. Collaborating with Google Maps would leverage their expertise in mapping and navigation, reducing the need for extensive self-development efforts.

However, this solution still entails a level of dependence on a third-party provider. Customization options may be limited to the features and functionalities provided by Google Maps. The extent of cooperation and the specific terms of the collaboration would need to be carefully negotiated to ensure that the project's indoor positioning requirements are adequately met.

5 Milestones

The project milestones are outlined in Table 7. While some minor adjustments were made to the schedule to account for the actual limitations and unforeseen challenges, the overall project trajectory remained consistent with the original plan.

Over the course of 7 months, the project team successfully accomplished the primary objectives by developing a comprehensive navigation system. This entailed creating a server service, establishing a database, constructing the mapping component, designing the detection model, and developing the mobile application. These efforts collectively enabled the delivery of an enhanced navigation experience within the HKU campus.

Throughout the project duration, extensive studies and experiments were conducted to gain valuable insights and gather comprehensive data. These insights were instrumental in the development of dedicated navigational maps and vision-positioning systems, further augmenting the project's capabilities and potential impact.

Period	Work Description	Progress
Sep - Oct	Analyse different proposed ideas' feasibility and effectiveness	Done
	Research and literature review of computer vision-based positioning	Done
	Design interface for mobile app	Done
Oct – Nov	Collect data in Main Building (2nd Floor)	Done
	Research in different computer vision models	Done
	Develop mobile app (both frontend & backend)	Done
	Data processing	Done
Nov – Dec	Build and train shortlisted machine model	Done
	Evaluate and select model with best performance	Done
Dec - Jan	Prepare interim report and presentation	Done
Jan - Feb	Compare and select suitable navigation mechanism	Done
	Expand model dataset	Done
	Evaluate extensive use of GPT4 on vision-based localization	Done
	Fine tuning the model	Done
Feb - Mar	Debug and improve mobile application and model	Done
	Integrate model and map mechanism to mobile application	Done
	Retrain model with expanded dataset	Done
	Perform user testing	Done
Mar - Apr	Prepare final report and final presentation	Done
	Source code cleanup	Done
	Prepare for the project exhibition and project competition	Done

Table 7 Project Schedule Table

6 Future Works

Future works of the project should focus on scope expansion, collaboration with established platforms, the incorporation of innovative technologies for vision-based indoor positioning to further enhance the project's capabilities and deliver an exceptional indoor navigation solution.

6.1 Scope Expansion

To fulfill the project's objective of providing indoor navigation assistance to users, the next phase should focus on expanding the scope to encompass the entire campus of HKU. Currently, the project group has managed to record and provide indoor navigation services for facilities located only on 2/F of the Main Building. However, it is important to note that as the coverage expands to include more buildings, the effort and resources required will increase significantly.

6.2 Collaborations for Indoor Map Construction

As emphasized earlier, the project has a high dependence on third-party map services. In this regard, ManiFold Tech Limited, a startup specializing in three-dimensional reconstruction technology, emerges as a strong candidate for collaboration. Leveraging the tools provided by ManiFold Tech Limited, the project team can scan the surroundings and create customized and detailed maps for navigation purposes. Furthermore, the incorporation of virtual reality, augmented reality, and mixed reality into the navigation guidance system would significantly enhance the user experience, and ManiFold Tech Limited can provide extensive support in this area.

Additionally, during the development process, the project team noticed that the Google Maps Team is continuously integrating their indoor navigation system, leading to improved accuracy and robustness of their services. Exploring partnerships with established navigation platforms, such as Google Maps, presents an opportunity for collaboration and mutual learning. By leveraging the expertise and advancements of these platforms, the project can benefit from improved accuracy, updated mapping data, and wider user accessibility.

6.3 Continuous Emerging New Technologies

Looking ahead, the detection mechanism will be further enhanced through the integration of innovative technologies. For example, the integration of the machine learning layer and camera calibration techniques can be pursued. This integration holds the potential to refine the detection results even further and provide increasingly precise location information.

Additionally, as the field of natural language processing continues to advance, future work should also consider leveraging the potential of more advanced language models, such as GPT-4. By leveraging the vision power of GPT-4, the project may benefit from a more scalable and cost-effective solution for indoor positioning.

7 Contribution

This chapter focuses on the labor division and individual contributions within the project team. During the course of the project, all members of the team collaborated effectively, ensuring that assigned tasks were completed efficiently and within the designated timeframes.

Table 8 presents a comprehensive overview of the key contributors to significant tasks within the project. Despite the varying levels of effort and expertise required for these tasks, the project team members have consistently shown their dedication and commitment. Their unwavering commitment is evident through their valuable insights, active participation in discussions, and their ability to foster a productive environment for knowledge exchange among team members.

Task	**Contributor(s)**
Website building	Liu Kan Man, Wong Riley Hoi-kiu
Data preparation (i.e. photo taking, data labelling)	Wong Riley Hoi-kiu, Ng Enoch
Mobile Application Development	Liu Kan Man
Socket Connection setup	Liu Kan Man
Model tuning and training	Ng Enoch
Integration of object detection model with mobile application	Ng Enoch, Liu Kan Man
Model related function (i.e. Layer filtering mechanism)	Ng Enoch, Liu Kan Man
Database setup	Ng Enoch, Liu Kan Man
Research and experiment (GPT4)	Wong Riley Hoi-kiu
Application and model testing	Wong Riley Hoi-kiu, Ng Enoch, Liu Kan Man

Table 8 Table for recording contribution and labour of division in the project

8 Conclusion

The primary objective of this project is to develop an efficient vision-based navigation system and contribute to the research field of computer vision applications. The implemented navigation system has demonstrated its effectiveness in enhancing the visitor experience by providing accurate and efficient indoor navigation within the HKU campus. The system caters to the needs of individuals who are unfamiliar with the building layout or require specialized navigation assistance. Furthermore, this project contributes to the advancement of indoor navigation technologies, particularly within complex building structures, serving as a valuable reference for future endeavors in this domain.

The paper provides a comprehensive analysis of the methodologies employed and the outcomes achieved in designing the real-time system, application, and model. The implementation of the mobile application and backend service has been completed, establishing a solid foundation for the project. The object detection model has been trained using the SGD optimizer, utilizing a dataset comprising 6474 collected images. To optimize accuracy for navigation purposes, a two-layer structured algorithm, comprising an object detection model and a filtering and matching algorithm, has been implemented. For the object detection model, the quantity and the selection of objects are significant factors affecting the location detection accuracy.

Considerable attention has been given to factors such as data acquisition, dataset size, and model selection to ensure the cost-effectiveness of the project. As the project scope increases, it becomes crucial to manage the time spent on data collection to ensure scalability. Additionally, the real-time capability of the positioning system is a critical factor in developing a location detection mechanism, as the complexity and speed of the algorithm directly impact the user experience.

It is important to note that the rapid evolution of technology brings forth new possibilities and potential advancements for vision positioning systems. Despite its current limitations and relatively immature state, GPT-4 holds promise for reducing data acquisition time through its future potential for zero-shot generalization. OpenAI has allocated significant resources to enhance the vision capabilities of GPT-4, making it a viable option for further exploration in addressing indoor positioning challenges.

References

[1]Communications and Public Affairs Office, “HKU Quick Stats 2022”. https://www.cpao.hku.hk/qstats/files/Archive/2022.pdf (accessed Sep 19, 2023).

[2] H. Motte, J. Wyffels, L. De Strycker, and J. P. Goemaere, “Evaluating GPS data in indoor environments,”.https://www.researchgate.net/publication/269978666_Evaluating_GPS_Data_in_Ind oor_Environments. Advances in Electrical and Computer Engineering, vol. 11, no. 3, pp. 25–28, Jan. 2011, doi: 10.4316/aece.2011.03004. (accessed Sep 21, 2023).

[3] T. Wu, L.-K. Chen, and Y. Hong, A Vision-Based Indoor Positioning Method with High Accuracy and Efficiency Based on Self-Optimized Ordered Visual Vocabulary, https://lightweb.ie.cuhk.edu.hk/api/publication/1658736041110- 07479682.pdf (accessed Sep. 21, 2023).

[4] D. Khan, Z. Cheng, H. Uchiyama, S. Ali, M. Asshad, and K. Kiyokawa, “Recent advances in vision-based indoor navigation: A systematic literature review,”.https://www.sciencedirect.com/science/article/abs/pii/S0097849322000371. Computers & Graphics, vol. 104, pp. 24–45, May 2022, DOI: 10.1016/j.cag.2022.03.005. (accessed Sep 22, 2023).

[5] C. Wiegand, “Achieving Blue Dot: Best types of indoor positioning systems,” Mar. 22, 2023. https://www.inpixon.com/blog/what-is-the-best-system-for-achieving-blue-dot- indoors

(accessed Sep 22, 2023).

[6] S. Ahmad, “How is Mobile Computer Vision Changing the World?,” Mobisoft Infotech, Dec. 01, 2020. https://mobisoftinfotech.com/resources/blog/how-is-mobile- computer-vision-changing-the-world/ (accessed Sep 23, 2023).

[7] Grigorescu, Sorin & Trasnea, Bogdan & Cocias, Tiberiu & Macesanu, Gigel. (2019). A survey of deep learning techniques for autonomous driving. Journal of Field Robotics. 37. 10.1002/rob.21918. (accessed Dec 21, 2023)

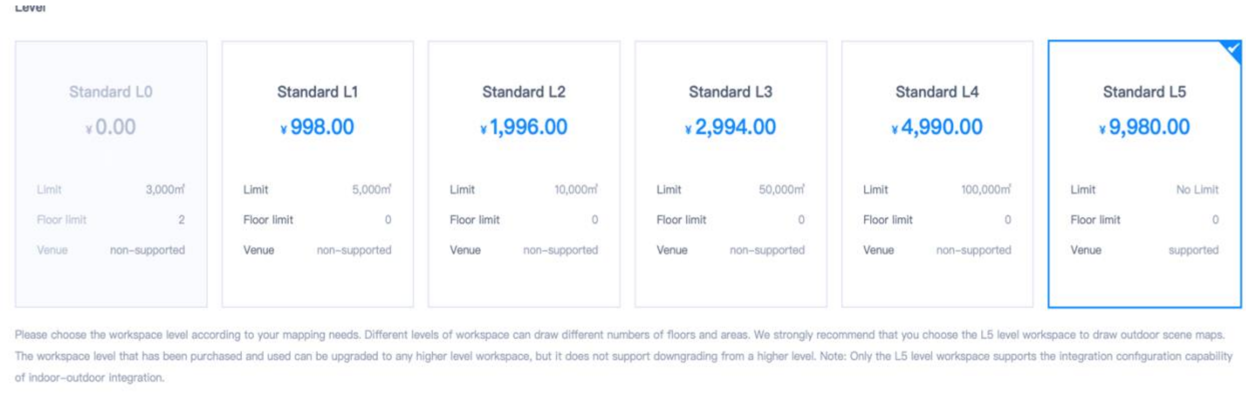
[8] Q. Lin, G. Ye, J. Wang, & H. Liu, "RoboFlow: a Data-centric Workflow Management System for Developing AI-enhanced Robots." Proceedings of the 5th Conference on Robot Learning, PMLR 164:1789-1794, 2022. https://proceedings.mlr.press/v164/lin22c.html. (accessed Nov 30, 2023).

[9] Chatgpt — release notes | openai help center, https://help.openai.com/en/articles/6825453-chatgpt-release-notes (accessed Apr. 24, 2024).

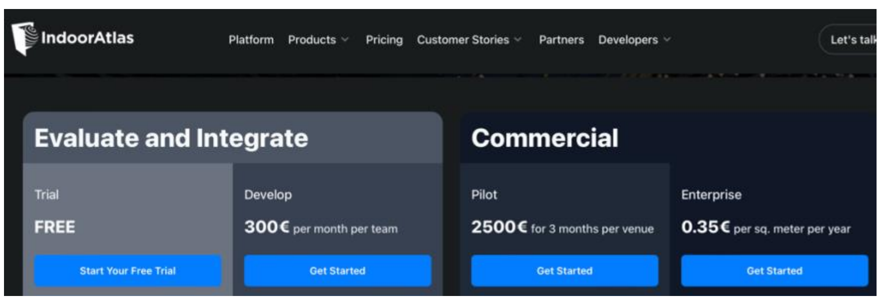
[10] "Documentation of GPT-4 Turbo," OpenAI. https://platform.openai.com/docs/guides/vision (accessed Apr. 02, 2024).

Appendices

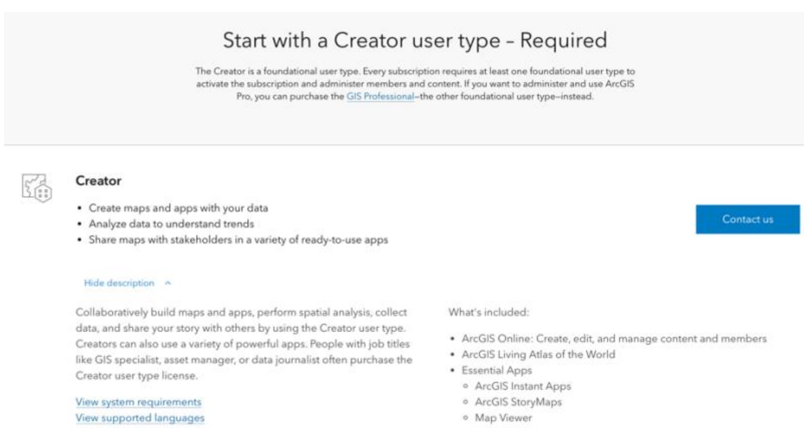
Appendix A – FengMap Pricing



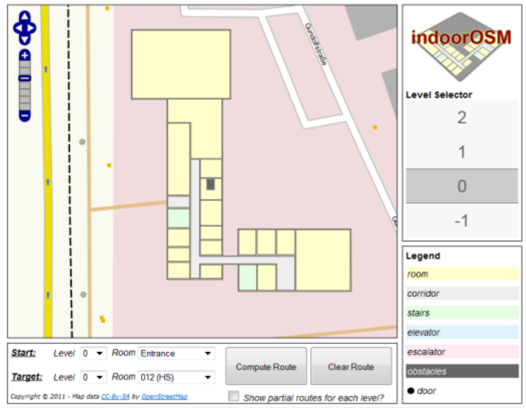
Appendix B – IndoorAtlas Pricing



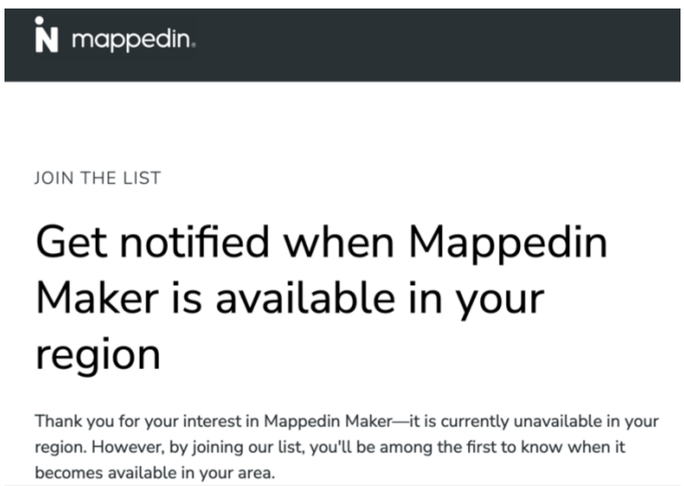
Appendix C – ArcGIS Pricing



Appendix D – Indoor Map of Demonstration of OpenStreetMap



Appendix E – Inaccessibility of MappedIn

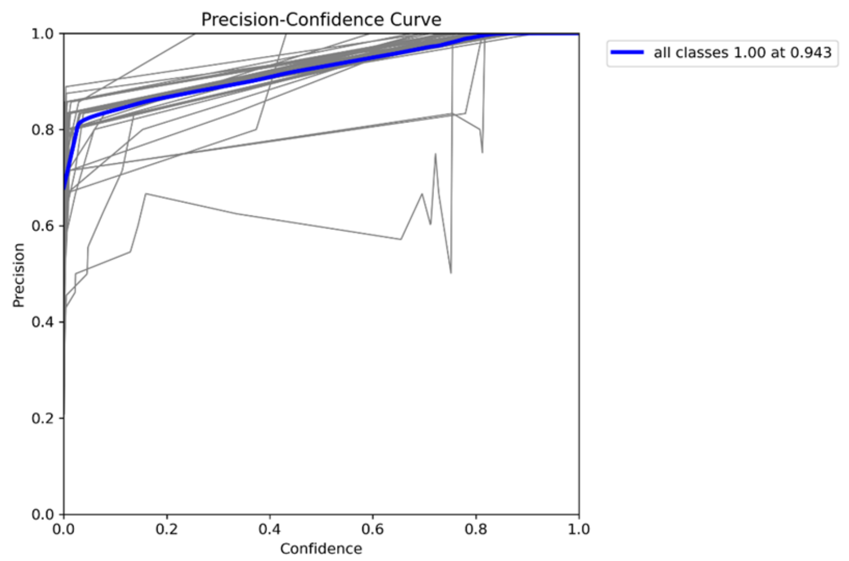


Appendix F – Used parameters of model training in optimizer algorithms selection


```yaml
task: detect
mode: train
model: yolov8n.pt
epochs: 300
patience: 20
batch: 16
imgsz: 640
save: true
save_period: -1
cache: false
device: 0
workers: 8
project: null
name: train
exist_ok: false
pretrained: true
verbose: true
seed: 0
deterministic: true
single_cls: false
rect: false
cos_lr: false
close_mosaic: 10
resume: false
amp: true
fraction: 1.0
profile: false
freeze: null
overlap_mask: true
mask_ratio: 4
dropout: 0.0
val: false
split: val
save_json: false
save_hybrid: false
conf: null
iou: 0.7
max_det: 300
half: false
dnn: false
plots: true
source: null
vid_stride: 1
stream_buffer: false
visualize: false
augment: false
agnostic_nms: false
classes: null
retina_masks: false
show: false
save_frames: false
save_txt: false
save_conf: false
save_crop: false
show_labels: true
show_conf: true
show_boxes: true
line_width: null
format: torchscript
keras: false
optimize: false
int8: false
dynamic: false
simplify: false
opset: null
workspace: 4
nms: false
lr0: 0.01
lrf: 0.01
momentum: 0.937
weight_decay: 0.0005
warmup_epochs: 3.0
warmup_momentum: 0.8
warmup_bias_lr: 0.1
box: 7.5
cls: 0.5
dfl: 1.5
pose: 12.0
kobj: 1.0
label_smoothing: 0.0
nbs: 64
hsv_h: 0.015
hsv_s: 0.7
hsv_v: 0.4
degrees: 0.0
translate: 0.1
scale: 0.5
shear: 0.0
perspective: 0.0
flipud: 0.0
fliplr: 0.5
mosaic: 1.0
mixup: 0.0
copy_paste: 0.0
cfg: null
tracker: botsort.yaml
save_dir:
runs\detect\train
```

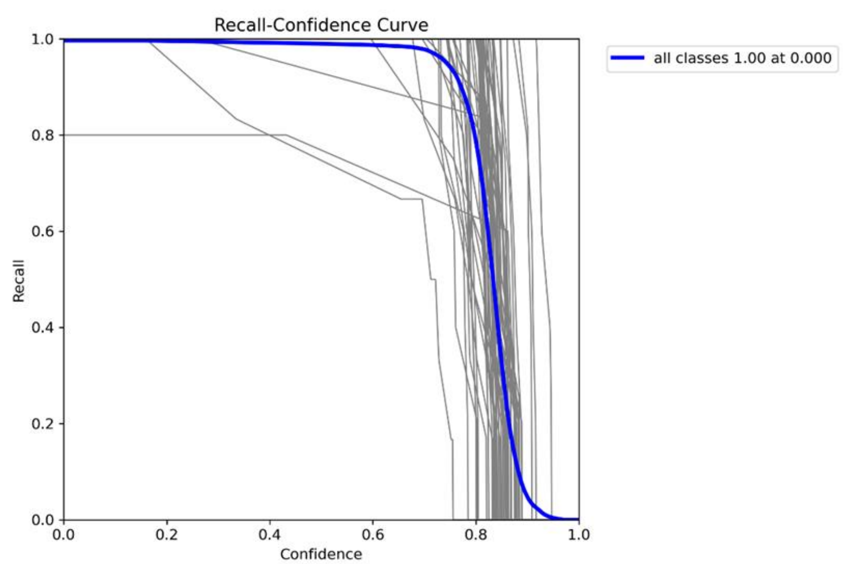
Appendix G – Evaluation on result using different optimizer algorithms

SGD



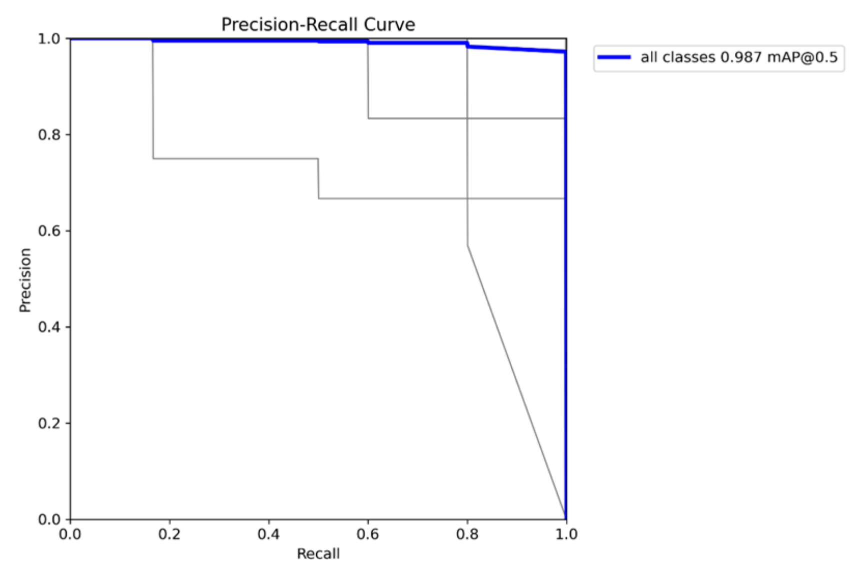
Precision-Confidence Curve for model using SGD

From above figure, precision curves for each class are establishing an overall upward trend at a high precision value. The blue curve indicates the overall class precision curve for the model. As the confidence threshold value increases, the precision of the model increases correspondingly and achieves maximum precision (precision = 1.0) when the confidence level is set as 0.943.



Recall-Confidence Curve for model using SGD

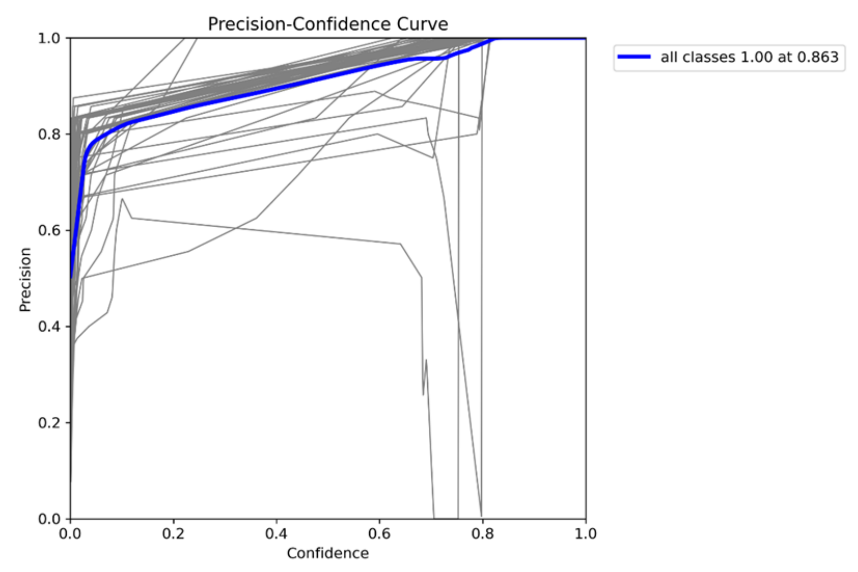
The presented recall curve, as depicted in above figure, corresponds to the performance of a model that has employed the stochastic gradient descent (SGD) optimizer algorithm. The curve shows a perfect recall value of 1.0 when the confidence threshold is set between 0.0 and 0.4, indicating that the model's performance is optimal when the threshold is lenient. As the confidence threshold value increases, a slight decline in the recall curve is observed. However, when the confidence threshold value reaches 0.8, the recall curve exhibits a gradual decline, indicating that the model's ability to make true positive predictions decreases as the threshold becomes stringent.



Precision-Recall Curve for model using SGD

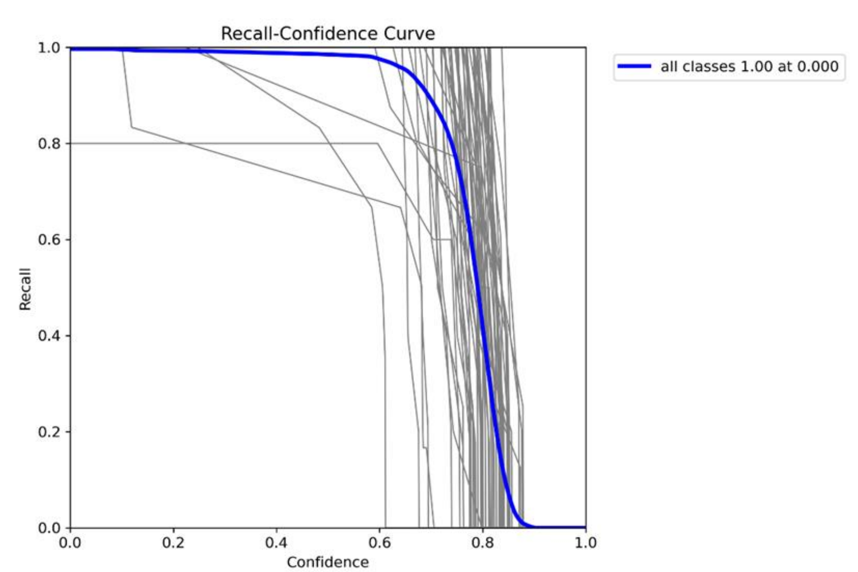
Above Figure illustrates the trade-off for precision and recall for the model trained with SGD as the optimizer algorithm. The overall class precision-recall curve indicated that the model has a high precision value and high recall value at most of the threshold values. From the graph, we obtained mAP@0.5 is equal to 0.987 which shows that the model has a very good performance in terms of the trade-off between precision and recall. The precision-recall curve for the model is almost identical to the precision-recall curve of an ideal detector, high precision value and high recall value at all confidence thresholds.

Adam



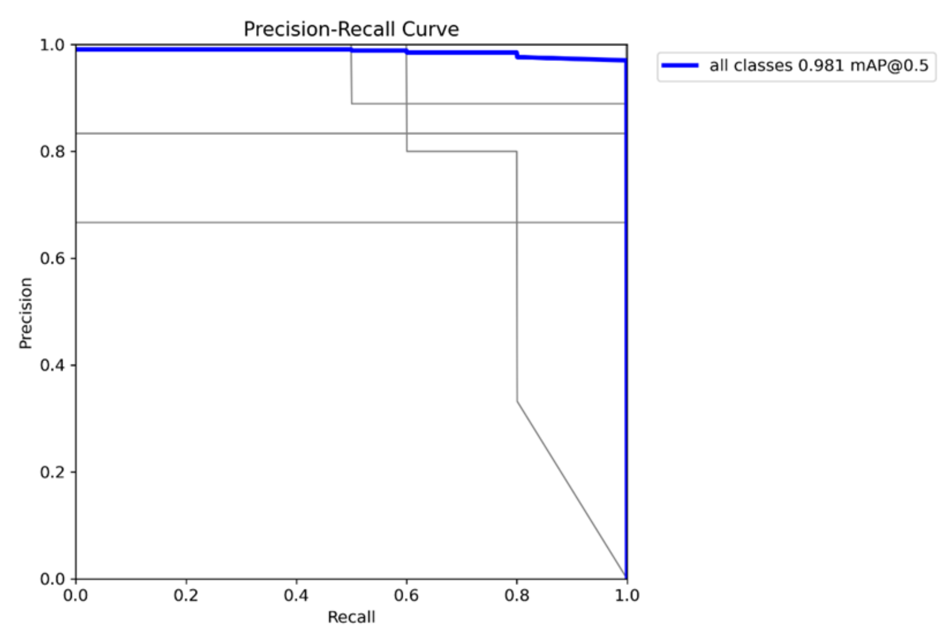
Precision-Confidence Curve for model using Adam

From the precision curve, the precision curve of the overall classes established a general upward trend and achieved 1.0 precision at a 0.863 confidence level. By looking at the precision curve for individual classes, a fluctuating and inconsistent pattern is observed. Precision curve of individuals classes reaches 0.0 precision at confidence score threshold ranged from 0.7-0.8. It indicates that the model performance in terms of precision for individual classes is not consistently reliable and precise.



Recall-Confidence Curve for model using Adam

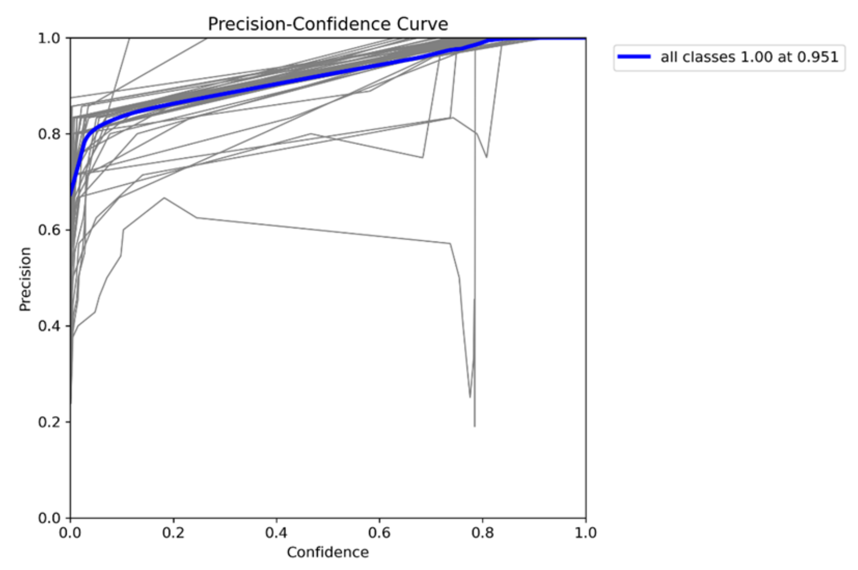
Above is the Recall curve obtained after the model training using Adam as the optimizer algorithm. It has a very similar trend to the recall curve for the model using SGD. However, the perfect recall value is only maintained when the confidence is set between 0.0 – 0.3. The recall curve declined as the confidence threshold value increased and drastically dropped when the confidence threshold was set between 0.75 - 0.85. From the graph, some classes even result in 0.0 recall at a 0.62 confidence threshold. It shows that the model may not be able to be detected in a stringent confidence threshold.



Precision-Recall Curve for model using Adam

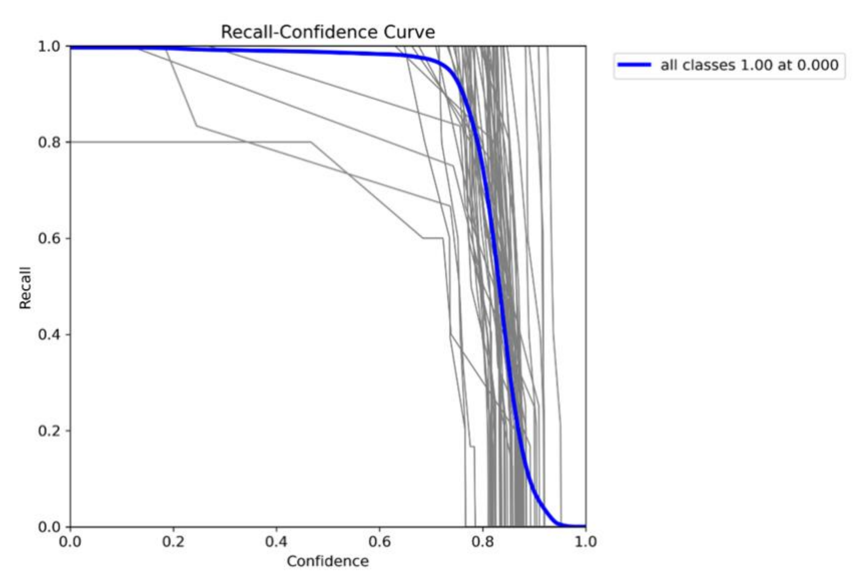
Above Figure depicts a similar precision-recall curve pattern as SGD. The overall class precision-recall curve indicated that the model has a high precision value and high recall value at most of the threshold values. The mAP@0.5 value obtained from the precision-recall curve is 0.981, indicating the model has an acceptably good performance in terms of precision-recall trade-off.

AdamW



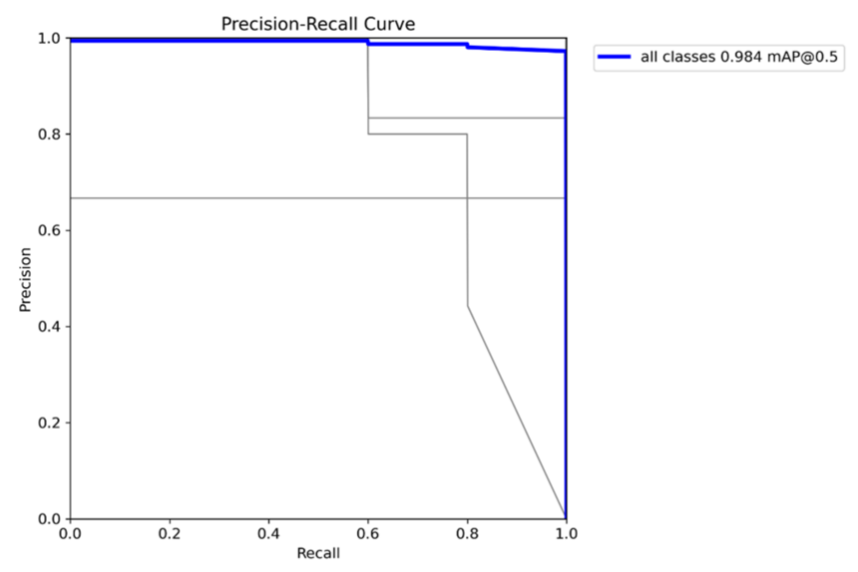
Precision-Confidence Curve for model using AdamW

The blue curve representing the precision curve for overall classes establishes an upward trend. The model obtained a precision score of 1.0 when the confidence score threshold was set to 0.951. However, for some classes, the precision fluctuated as the confidence score threshold increased.



Recall-Confidence Curve for model using AdamW

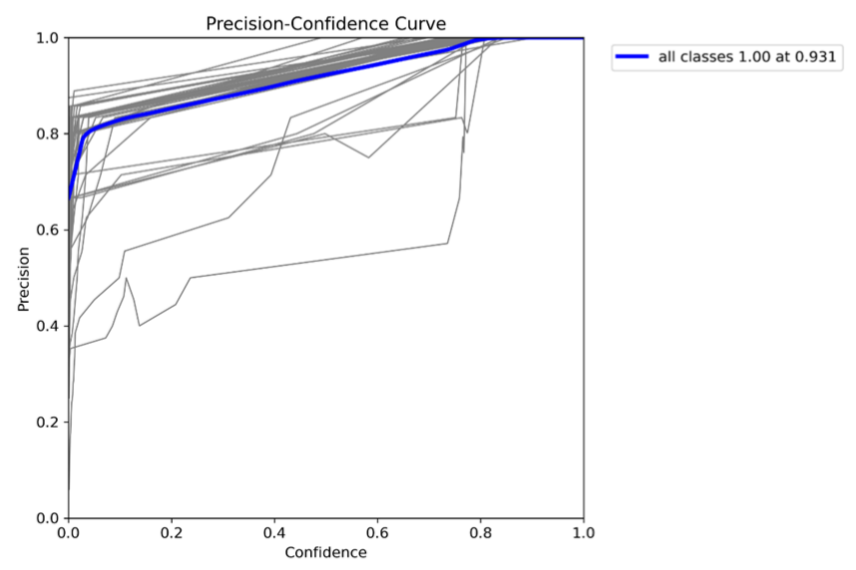
By comparing the above figure with the recall curve for the model using SGD, a similar pattern is observed. Both curves display comparatively high recall values at lenient confidence threshold levels. As the confidence threshold becomes stricter, the recall value drops drastically from the 0.75-0.9 confidence threshold level. The recall curve demonstrates that the model has an overall good performance in terms of recall at different confidence thresholds.



Precision-Recall Curve for model using AdamW

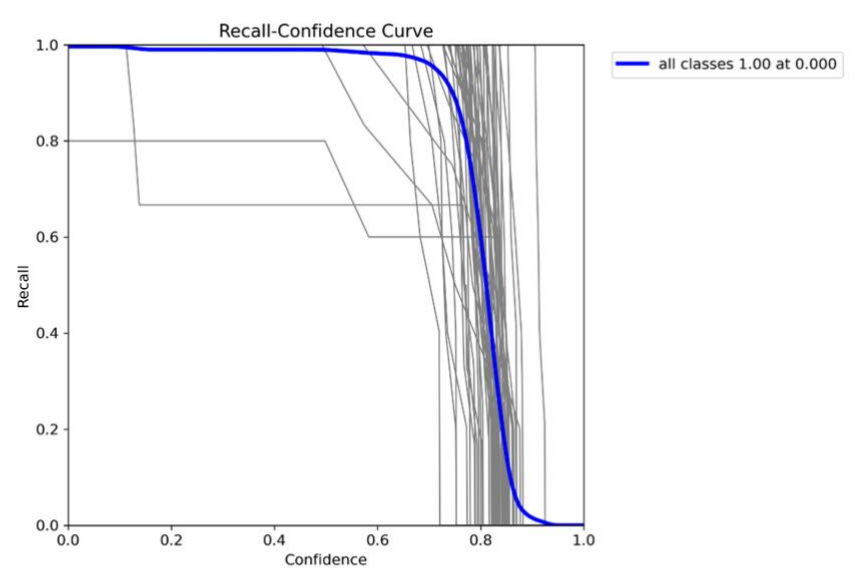
The presented analysis depicts a precision-recall curve pattern that is analogous to the one shown by the SGD model. The overall precision-recall curve reveals high precision and recall values for the model at most threshold values. The mAP@0.5 value obtained from the precision-recall curve is 0.984, indicating the model has an acceptably good performance in terms of precision-recall trade-off.

Adamax



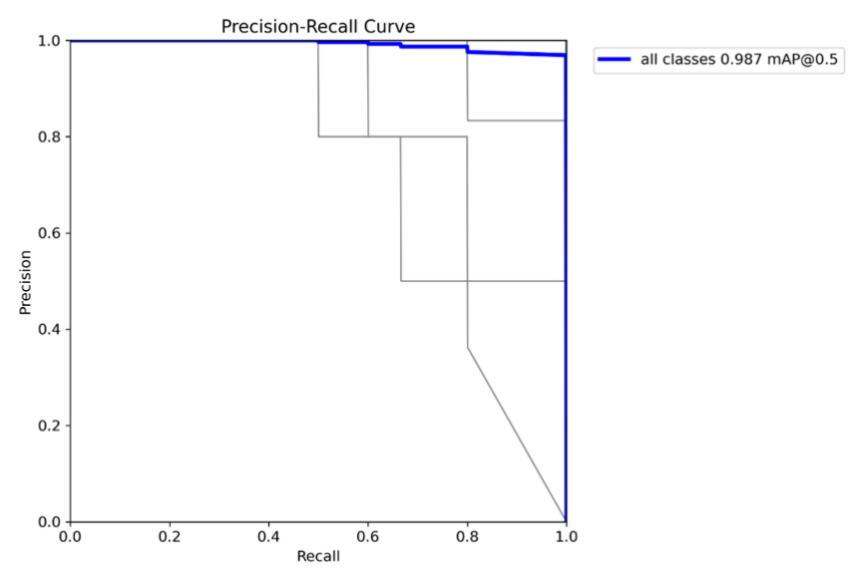
Precision-Confidence Curve for model using Adamax

The blue curve demonstrates an upward trend for the overall class precision. Moreover, the perfect value of the overall class precision is retained when the confidence score threshold is equal to or greater than 0.931. It is noteworthy that as the confidence score threshold value increases, there is a steady improvement in precision observed for each class.



Recall-Confidence Curve for model using Adamax

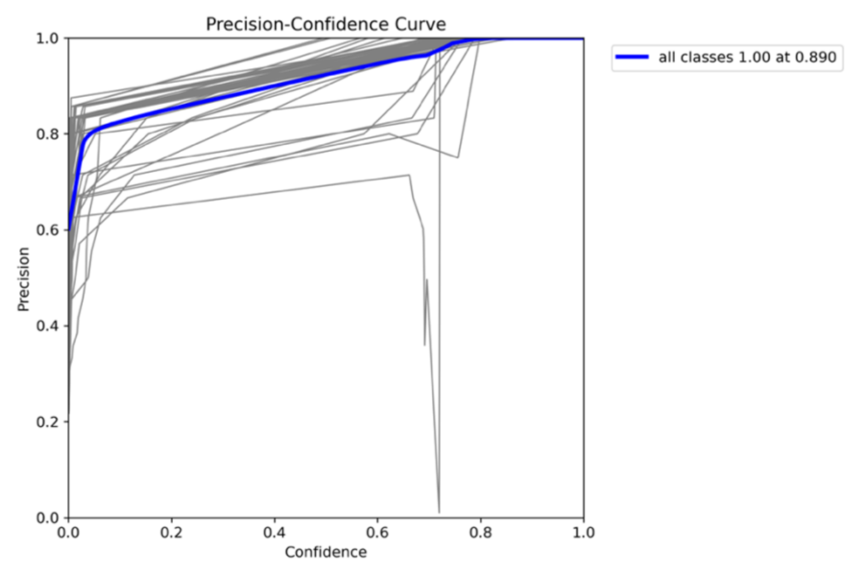
Above, high recall values are obtained when the confidence threshold value is lenient (0.0 – 0.6). The recall curve gradually declined when the confidence threshold value became stringent, decreasing when the confidence was set between 0.75 - 0.9. Moreover, some classes, represented by the grey curve, result in a relatively low recall value at lenient confidence threshold values. It indicates that, for some of the classes, the detection may not be able to discover targeted features successfully.



Precision-Recall Curve for model using Adamax

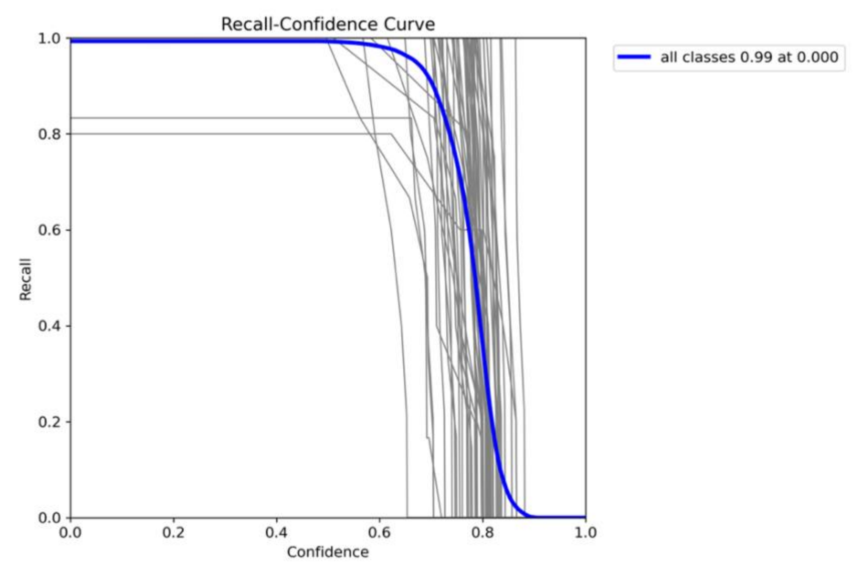
Above depicts the precision-recall curve for the model using Adamax. The mAP@0.5 value is 0.987, which is the same as the mAP@0.5 score for the model using SGD, for the overall class performance. Moreover, by comparing the pattern of the individual class's precision-recall curve from both figures, the model using Adamax as an optimizer algorithm demonstrates a better performance in terms of the trade-off between precision and recall.

RAdam



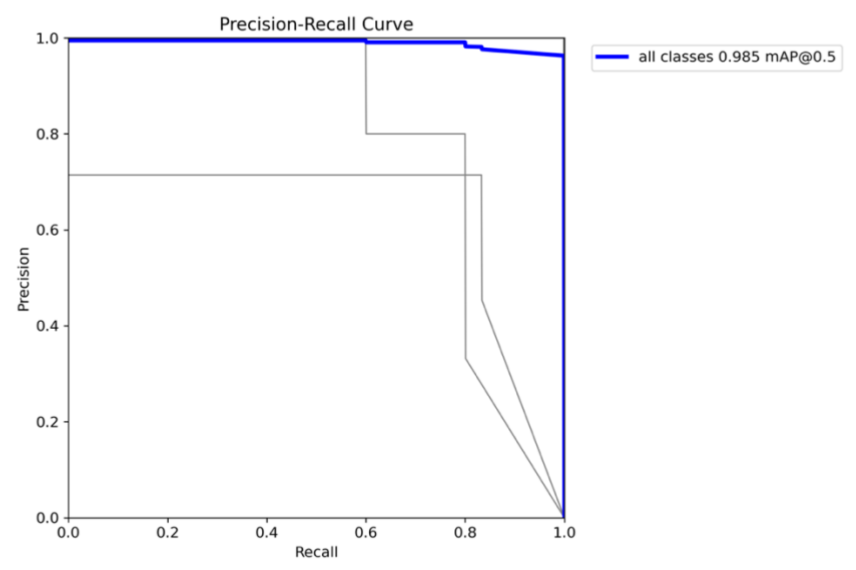
Precision-Confidence Curve for model using RAdam

The precision curve shown illustrates an upward trend. The precision for all classes achieves 1.0 when the confidence threshold value is equal to or greater than 0.89. Further looking into the precision performance of individual classes, abnormal cases, where the precision fluctuated drastically, are observed. The presence of the pit in the precision-confidence curve suggests that the model is struggling with making correct predictions in a certain range of confidence score threshold for some of the classes. Therefore, there is still room for improvement in terms of prediction accuracy.



Recall-Confidence Curve of model using RAdam

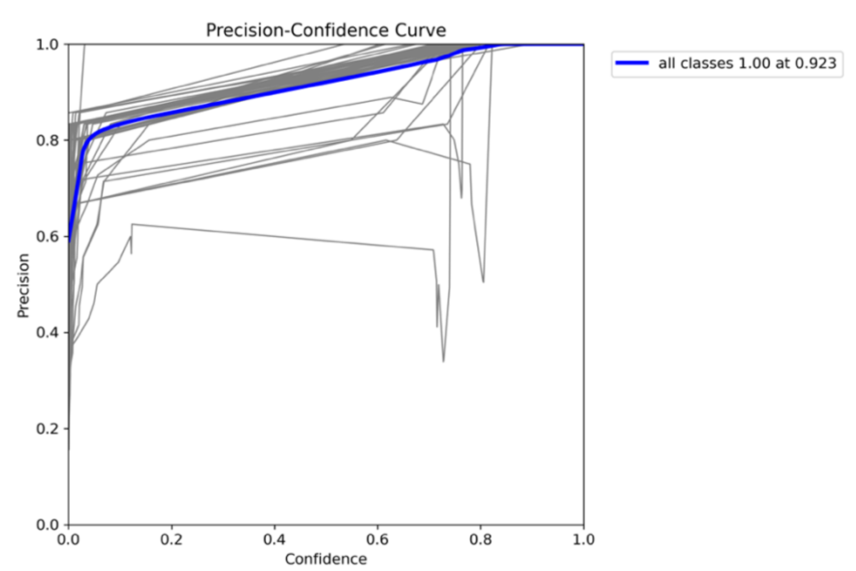
The recall-confidence curve shown follows a similar pattern as the ones obtained from models that use different optimizers. The performance, measured in terms of overall class recall, is good for lenient confidence threshold levels. As the confidence level becomes more stringent, the recall value decreases. Notably, the overall class recall value is not as high as that of other optimizers, where maximum recall value is 1.0, whereas the model has a maximum recall value of 0.99. The slight difference suggests that both the model and dataset may require further improvement to provide better results.



Recall-Confidence Curve of model using RAdam

The precision-recall curve provides an approximately ideal model performance in terms of the trade-off between precision and recall. The precision value remains high for most of the recall values. The mAP@0.5 value obtained from the figure is 0.985.

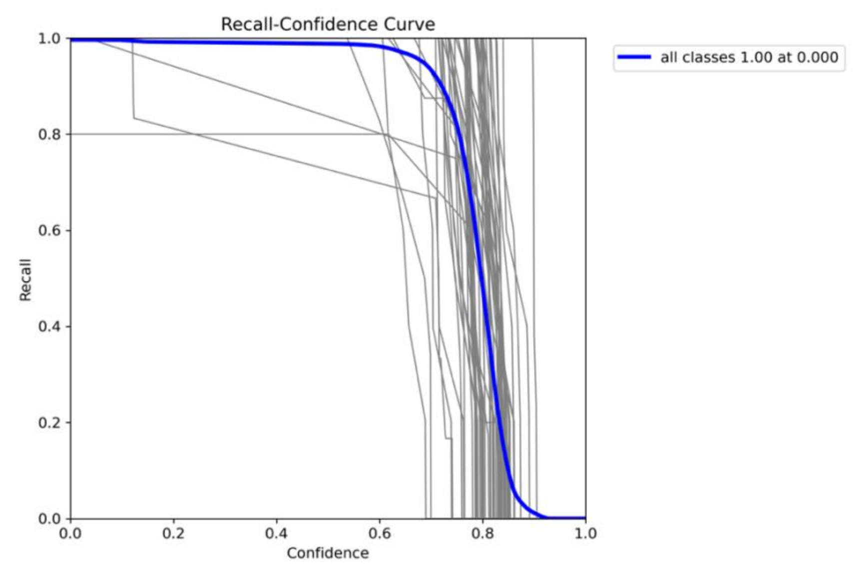
NAdam



Precision-Confidence Curve of model using NAdam

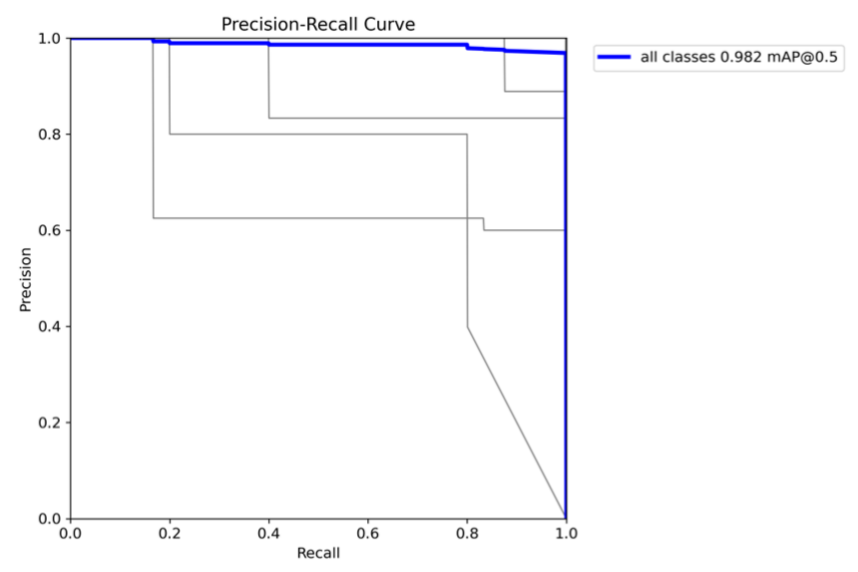
The overall class precision observed shows an upward trend as the confidence threshold value increases. The model achieved a precision value of 1.0 at a confidence threshold value equal to 0.923 for all classes. However, the precision-confidence curve for individual classes fluctuates.

This suggests that the accuracy of precision provided by the model in different classes can be inaccurate and unreliable.



Recall-Confidence Curve of model using NAdam

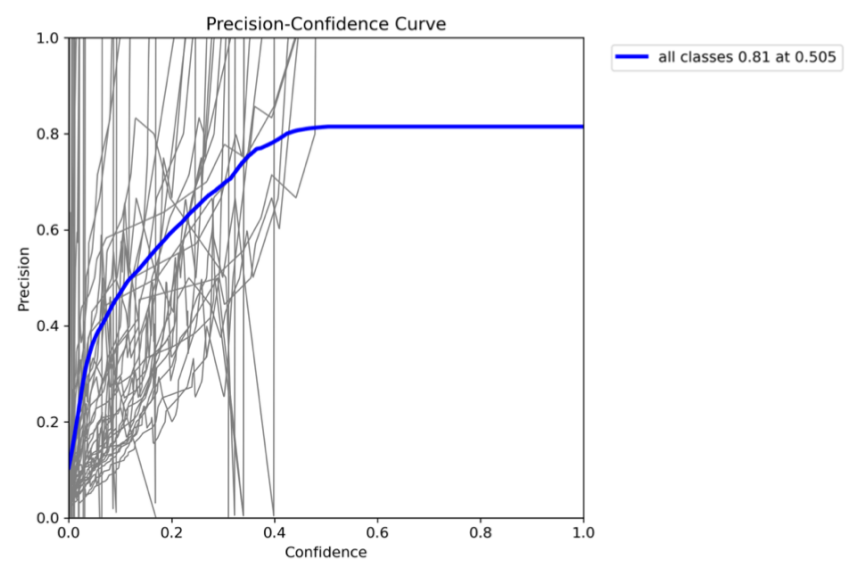
The recall-confidence curve drops slowly from confidence threshold 0.0 to confidence threshold 0.6. It declines drastically from confidence threshold 0.75 – 0.9. Additionally, the grey curves that represent the recall-confidence curves for individual classes display distinct fluctuations. Some of the classes exhibit a noteworthy decline, achieving only 0.8 recall at lenient confidence threshold level. These observations suggest that the machine learning model may not provide accurate predictions for some of the classes.



Precision-Recall Curve of model using NAdam

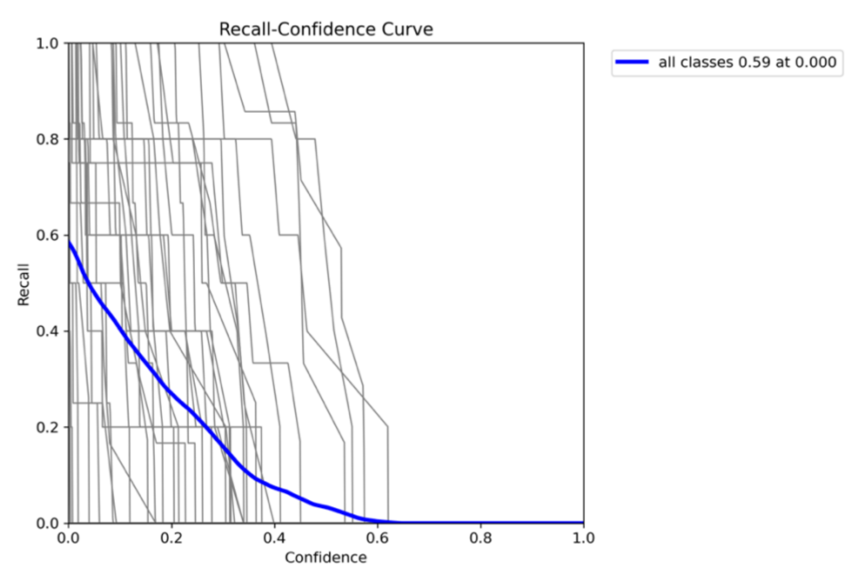
The above figure showcases the precision and recall trade-off of the model using NAdam. The representing precision-recall curve demonstrates a good performance: high precision value at different recall values. The mAP@0.5 obtained is equal to 0.982. However, the precision-recall curves for individual classes established a diversified and dispersed pattern. For some individuals, the precision value dropped at a recall level of 0.2. It implies that the trade-off between precision and recall for some of the classes may be dissatisfying. In other words, the detection results will be unreliable as the precision and recall is affected.

RMSProp



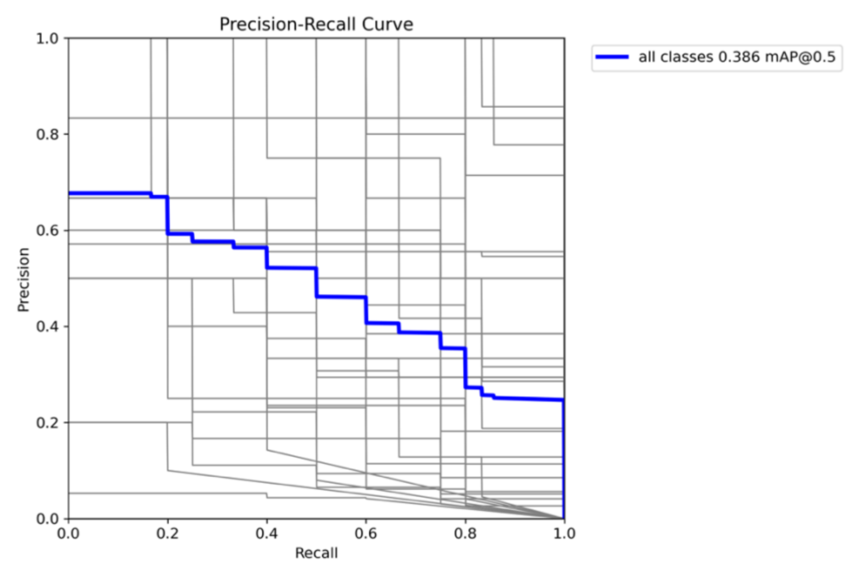
Precision-Recall Curve of model using NAdam

The overall precision performance is demonstrated. The overall precision achieves a maximum of 0.81 at the confidence threshold value of 0.505. It indicates that the model is struggling to make highly confident and precise results. The grey curves representing individual classes display an unstable and fluctuating precision-confidence pattern, further indicating that the model training with RMSProp under the given conditions may not be capable of delivering accurate predictions.



Recall-Confidence Curve of model using RMSProp

The Recall-Confidence Curve has an exponential decay pattern. The overall class recall value dropped from 0.59 to 0.0 across the confidence threshold value. This suggests that the model is incapable of identifying or capturing positive instances and making accurate predictions.



Precision-Recall Curve of mode using RMSProp

Above Figure displays a mAP@0.5 value of 0.386 and a precision-recall trade-off curve that is non-smooth and exhibits low values for precision and recall. This indicates a poor performance of the model, as reflected by the low mAP@0.5 value. The model's predictions do not meet the desired performance standards, being both inaccurate and incomplete.