1. Project Background

1.1. General Background

For most people, losing their automobile is a very distressing situation. These situations are often exacerbated when the victims face the frustration of long investigation periods and, as in most cases, the law enforcement authorities being unable to retrieve the stolen vehicle.

Motor vehicle theft, also known as car theft and grand theft auto, refers to the criminal act of stealing or attempting to steal a motor vehicle [1]. Statistically speaking, there were 352 cases of "Missing Motor Vehicle" cases in Hong Kong during the first six months of 2022 [2]. This is a common phenomenon in urban spaces; in Europe, motor vehicle theft accounted for approximately 7.4 billion Dollars worth of damages in 2020 [1].

The prevalence of this issue can mostly be attributed to the fact that most countries still rely on old methods for identifying and tracking down stolen vehicles, for example in the United Kingdom, law enforcement relies on methods such as random street checks, license plate checking of stolen vehicles and tips from crowd-sourced websites [1].

1.2. Research Gap

Despite recent advances in technology, the current method of identifying and spotting stolen vehicles relies heavily on manual labor. Law enforcement agencies (LEA) watch several hundreds of hours of surveillance videos to try and identify potential suspects of stolen vehicles, and they invest even more time and energy investigating each potential suspect.

The danger of this method is that, due to its time-consuming nature, by the time the stolen vehicle is identified it might be too late to efficiently intercept it. This issue is compounded when it is taken into account the fact that there are several cases of motor vehicle theft each year and many of these cases have to be resolved in concurrency with one another.

This project tries to address this problem by introducing a new method of identifying and tracking stolen motor vehicles. The project aims to automate the process of watching surveillance footage by creating an Artificial intelligence (AI) model which can identify cars and classify their features such as color, type, and license plate number. By comparing passing vehicles against a list of stolen vehicles law enforcers can be alerted whenever a potential suspect is identified, thus saving both time and energy.

1.3. Project Objective, Deliverable and Scope

The goal of this project is to create a Machine Learning (ML) Model that can be integrated into Hong Kong's existing network of surveillance cameras. The model aims at matching each vehicle it sees against the description of stolen vehicles, notifying the LEA in the event of a potential suspect.

The project can be defined by three distinct deliverables: an ML model that analyzes surveillance videos, a database server to store descriptions of stolen vehicles, and a mobile app interface for authorities to view the results of the model's analysis.

This three-part system allows great efficiency in deploying the project. It also makes for easy maintenance and upgrades in case of developments in the future; only the part which has to be upgraded can be turned offline and replaced while the rest of the system remains functional.

It also provides LEA with real-time information and updates as the notification system removes the need to manually request analysis results.

The mobile app interface enables the LEA to view the results of the ML model and make the active decision of whether or not the potential suspect is correctly identified. This interface allows for efficient human intervention to prevent false positives from wasting precious time and resources.

1.4. Justification of Work as a Solution

This project would greatly help in reducing, and possibly resolving, the issues faced by LEA in cases related to motor vehicle thefts.

Firstly, since the ML model is used to handle the analysis of surveillance videos, it becomes a simple task to run multiple instances of the ML model. In this manner, multiple video feeds are analyzed simultaneously allowing LEA to save on labor. costs. It also helps save time as the model can be left running continuously without any breaks.

Secondly, the model can be optimized to help LEA reduce the number of false positives; this helps prevent innocent civilians from being mistakenly labeled as potential suspects. This is highly beneficial as the reduced number of potential suspects translates to lesser amounts of time and labor required to resolve a case.

Finally, the model outputs can be analyzed to trace the path of stolen vehicles. Since all the surveillance camera's video feed is being watched simultaneously the moments a potential suspect passes through a camera are time-stamped. Multiple time stamps could then be plotted on a map to trace the path taken by potential suspects.

1.5. Outline

In the upcoming sections, this paper will explore the methodologies intended to implement the project. Section 2 will elaborate on the software, and hardware implementations as well as provide details on the projects’ plan for deployment.

Section 3 will discuss the project status and will evaluate the progress achieved as well as the schedule for future milestones. This section will also discuss the difficulties encountered as well as the solutions implemented to overcome these difficulties.

Finally, this paper concludes by giving a summary of the project status and highlights the limitations of the project; the final section also provides recommendations for future research.

2. Project Methodology

2.1. Overview

In this section, this paper will first discuss the design principles, before examining the hardware architecture, software architecture, mobile app interface, and model deployment. The paper will expand on each section while providing justification for the decisions.

2.2. Design Principles

The design of the project was made taking three principles into account: scalability, modularity and ease of maintenance.

In this paper, scalability refers to the projects’ ability to be modified in future. The project should be easy to upgrade to accommodate new technologies and techniques.

The modularity of the project refers to the division of the project deliverable into modules. By achieving this principle the project will have multiple parts running independently, this in turn enhances the project's ability to run on a variety of different hardware with little need for additional configuration.

Modularity also enables the project to carry out its third design principle, ease of maintenance. If achieved, this design principle will give LEA the freedom to hire independent teams for their project maintenance and remove any reliance on third-party services.

2.3. Hardware Architecture

2.3.a Hardware Selection

The primary hardware required for this project is the camera through which the ML model will receive the video feed. For this project, it was chosen to use the inbuilt camera of an OPPO A83 smartphone. While other, more suitable, options exist such as the digital single lens reflex (DSLR) camera or the traffic enforcement camera, the OPPO A83 camera was cheap and immediately available to use with no learning curve. The cost of a DSLR camera had to be taken into consideration as it exceeds the projects' operating budget and traffic enforcement cameras were ruled out as they were extremely difficult to procure.

Another candidate for the input camera was the Insta360 camera. While the Insta360 camera was freely available for rent from the University library and was capable of taking high-resolution videos, the project had to consider the format of input available. The Insta360 camera could only take panoramic videos, and thus to avoid data mismatch problems during the training and deployment of the ML model it was necessary to edit each frame of the video feed and convert them to an equirectangular format. Since Insta360 videos could only be edited by proprietary software there was no means for the project to automate the process. On account of time restrictions, the project chose not to move forward with the Insta360 camera.

The second hardware component crucial to the project is the computational system for training and testing the ML model. For this project, it was chosen to work with Hong Kong University's GPU cluster. The Hong Kong University GPU cluster is an array of servers built with high-end GPUs. This cluster allows staff and students to train complex ML models without the need for an expensive computing system. The project had the option of choosing to work with cloud computing services such as Google Cloud or Amazon Web Services, which were not only faster than the University cluster, but also came with pre-built functionalities that aided in AI development. Cloud computing companies provide the service of taking computationally expensive ML models and training them on behalf of the client; this service is commonly availed in academic projects as it is cheaper than investing in a private system. However due to the high costs associated with cloud computing services, the project move in the direction of Hong Kong University's GPU cluster as it was freely available for university staff and students.

2.3.b Hardware Layout

The project requires the cameras to be placed in an elevated position adjacent to the road with high visibility. Since the project is intended to be integrated into Hong Kong's traffic camera network, its placement must mimic the existing positions of traffic enforcement cameras. The goal of the camera positioning is to attempt at capturing the target vehicle while ensuring that the visibility of other vehicles is not obscured.

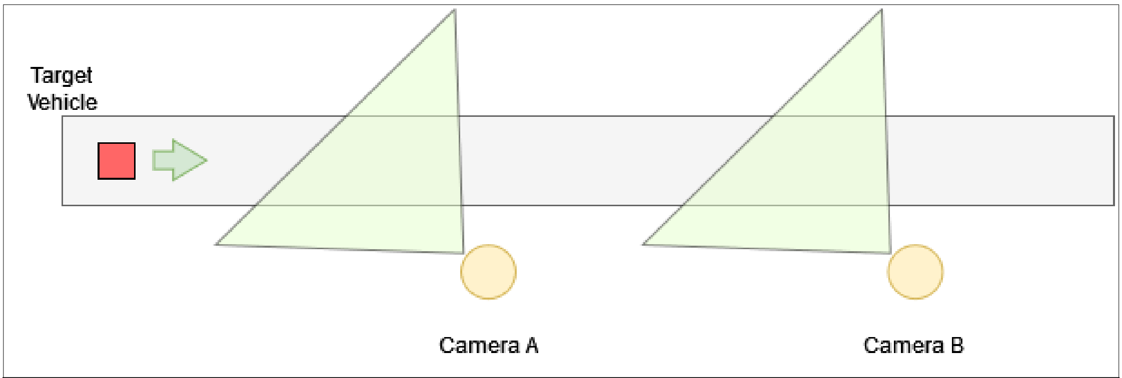

Figure 1: Visualization of the expected position of the cameras, relative to a road.

Figure 1 shows a visualization of how the project intends to place the cameras adjacent to a busy road. Each camera, upon detecting a suspect, sends a unique time-stamped notification to the LEA; a series of such operations generate the necessary data to infer the path taken by a stolen vehicle. This also allows for the LEA to estimate the best location to intercept the stolen vehicle. The layout intends to mimic the existing layout of Hong Kong's traffic enforcement cameras. However, this figure only provides a birds-eye view of the layout and may not properly convey other aspects of the layout such as camera height and angle.

2.4. Software Architecture

2.4.a. Video Stream Input

The video stream taken for the project will be pre-processed to resemble the video quality of traffic enforcement cameras. This pre-processing method will involve reducing the Frame-Per-Second (FPS) of the input video, as well as the pixel resolution of the input. The FPS of a video is dictated by the camera's hardware specification; it refers to the number of video frames taken per second. While high FPS is almost always ideal for creating powerful machine learning (ML) models, the performance of the model deteriorates if it is not trained using data similar to what it would receive on the field.

The pixel resolution of a video refers to the number of pixels used in each video frame; the pixel resolution of a video directly correlates to the sharpness of the videos taken. Similar to FPS the pixel resolution must be adjusted to match the specifications of traffic enforcement cameras.

2.4.b. ML Model Architecture

The ML model in this project makes use of Object Detection and Classification. Object Detection in ML refers to the statistical technique of drawing a bounding box around identified targets in the image frame, while classification in ML refers to the statistical technique of identifying and classifying the objects in an image. In this context, classification is used to identify the target captured by the object detection system.

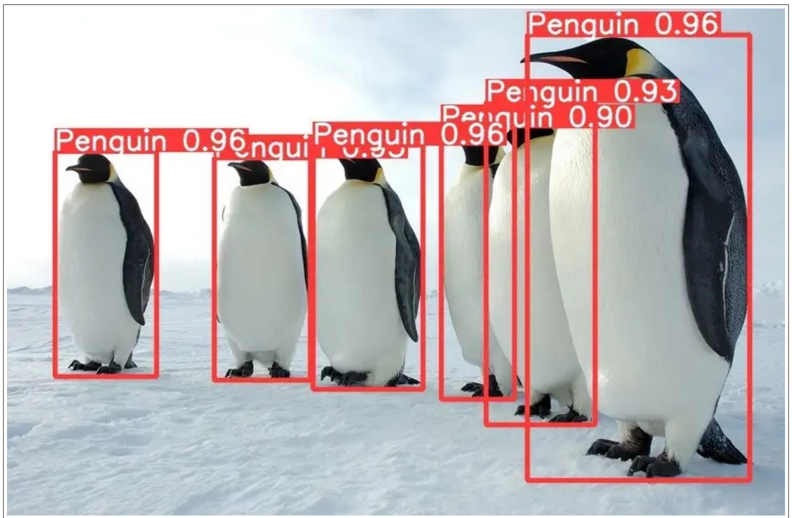

Figure 2: Example of Object Detection output. Bounding boxes drawn against an image of penguins.

In Figure 2, an example of the object detection and classification model's output is shown. Bounding boxes are drawn around targets and a classifier labels each box with the object's identity. In this particular example, the confidence score of the classifier is included in the label. The confidence score of a classifier is a numerical representation of how sure the model is of its result, for example, a confidence score of 0.96 translates to the model being 96 per cent sure of its answer. However this specific example was made using a high-contrast image with a plain background, this is a very ideal condition for an ML model and may not reflect its performance in real-life scenarios.

The project will also deploy a text recognition model. Similar to the classification model, a text recognition model is an ML model that is able to extract textual data from images. This second model is intended to run in parallel with the object detection model to extract information such as the license plate number of vehicles.

2.4.c. Database Architecture

The database of the project will be made using SQLite3. SQLite3 is an open-sourced Structured Query Language (SQL) database engine with a python wrapper. SQL is a programming language for handling data in a database. A database is a file which stores data; the data can then be retrieved efficiently based on specified properties.

SQLite3 will allow the project to seamlessly integrate database management capabilities into the existing framework written in python. The database is designed to store data in the third normal form. The third normal form refers to the database schema which aims to reduce the duplication of data, avoid data anomalies and simplify data management while protecting referential integrity [3]. Critical information such as properties of stolen vehicles, details of incident reports, and geographic locations of cameras, etc. is stored in the database.

2.5. Mobile App Interface

The project includes two mobile app interfaces among its final deliverables.

The first mobile app aims at allowing LEA to efficiently respond to potential suspects by receiving a notification through their mobile app. The app allows the LEA to view the camera footage and make an assessment of whether or not the potential suspect discovered is a false positive. It also allows the LEA to manually correct any wrong predictions such as misread license plate numbers.

The second mobile app interface is the camera interface. This mobile app converts mobile phones into impromptu camera modules. The app can integrate the ML model and surveil traffic for potential suspects. This architecture feature brings about two benefits: multiple instances of the ML model can be run in parallel in a distributed network; the cost of development and deployment is reduced as this removes the need for specialized equipment.

The project plans to create both mobile app interfaces in Kotlin using Android Studio IDE. This would allow the project better flexibility in testing the apps before deployment as the Android Studio IDE has the capability to emulate a large variety of Android Devices.

2.6. Model Deployment

This project plans to deploy the completed model on a cloud-based service such as Amazon Web Services. The goal is to demonstrate the flexibility of the ML model as well as its modularity. This also serves the purpose of marketing to LEA, as this demonstrates the model's capability to be deployed on virtual machines. A virtual machine is a software which emulates an independent computer, it allows for the efficient sharing and distribution of hardware resources.

3. Project Status

This section of the paper explores the current status of the project and provides insight into the implementation of the processes mentioned in the previous section.

3.1. Evaluation of Progress

The project has completed its data collection stage. In this stage, several hours of street traffic video footage were captured and broken into individual frames. The cameras were set up in a configuration similar to Hong Kong's traffic enforcement cameras (see section 2.3.b).

The data labeling stage was initiated immediately after the completion of the data collection; the project makes use of a naive model to automate the process of data labeling. The project uses the tool, ROBOFlow, to integrate the naive model into our data labeling process. Unfortunately, the team was unable to complete this stage within the expected deadline as the team had grossly underestimated the number of man-hours necessary to label the images. This was also exacerbated due to the increase in University course workload towards the end of the semester and the resulting conflicts in time schedules between the team members.

In parallel with data labeling, the project initiated the development of the mobile app interfaces and the centralized database server. The centralized database server is a web server written in python flask and uses SQLite3 as the database engine. The web server was also tested using Pythons' unittest module to adhere to the principle of test-driven development.

3.2. Expected Outcomes

At current there are two expected outcomes at this stage of the project: the first is a set of labels for each image in the training and testing pair, and the second is an ML model which can detect and locate stolen vehicles in images.

The labels are necessary, as it provides the correct answers for the ML model to learn from, and it also helps the project verify the accuracy and performance of the ML model.

3.3. Project Schedule

The project schedule for the first stage includes nine milestone.

Milestone	Original Due Date	Project status	Revised Due Date	Task Allotment
Data Collection	28/10/2022	Completed	-	Asim
Data Labeling	30/12/2022	In Progress	28/02/2023	Young
Web Server	25/12/2022	Completed	-	Asim
Client App Interface	18/02/2023	In Progress	-	Asim
Camera App Interface	25/02/2023	In Progress	-	Asim
Model Creation	20/01/2023	Postponed	15/03/2023	Young
Model Testing	23/01/2023	Postponed	18/03/2023	Asim & Young
Model Ensemble Testing	04/02/2023	Postponed	30/03/2023	Asim & Young
Deployment and Field Test	02/04/2023	-	-	Asim & Young

Table 1: Milestones for the first phase of model development and their expected completion dates.

Table 1 shows the milestones for this stage of the project. The first two milestones were explained in the previous section and are associated with data preparation, while the third, fourth and fifth milestone is responsible for the creation of a user interface. The sixth, seventh and eighth milestone is responsible for the creation and testing of the ML model. While the table gives a simple representation of the project schedule, it does not show all the smaller tasks associated with each milestone. The table also shows the current status of each milestone as well as their revised completion dates.

Due to the increased workload in the project teams' academic courses and scheduling conflicts, the project had to be postponed. In reflection of this, the deadlines had to be extended to accommodate the academic requirements of the team.

3.4. Remaining Work

The project has yet to complete its labeling of image frames. This process, despite being assisted by a naive model, is still time intensive.

The project also has to repeat the experimentation of the ML model built with third-party data (see section 3.1) and examine if the same results can be reproduced with the data collected for the project.

After the completion of the postponed milestones, the next step of the project would be the final testing and fine-tuning of the model. At this stage, a video feed collected from the street camera (see section 2.3.b) will be fed to the model to examine its performance in a simulated environment. Any incorrect parameters will be fine-tuned before finally testing the model against a live feed of a street camera.

The final milestone will be the integration of all the independent pieces of the project. The deployment of the web server and the assimilation of the ML model into the Camera App interface. At this stage, the project may be released for beta testing to various LEA.

In addition to the previously proposed milestones, an additional goal has been added to test and experiment with the behavior of the ML model when it is used as a part of a larger ensemble model. In ensemble models, multiple similar models compete on the same input data and the output selected is based on a chosen metric. The models are each trained in a different environment with varying hyper-parameters. For this project, it is proposed to base the ensemble output selection on the confidence scores of each model.

3.5. Difficulties Encountered

The project has faced three main obstacles up til the writing of this paper.

Firstly, the project was suffering from label imbalance in the data collected. Label imbalance refers to the problem where a certain label occurs much more frequently than other labels. In this particular context, since the data collection relies on using image frames from traffic videos (see section 2.4.a for more details), there were more images of empty streets than there were of motor vehicles. This difference in label frequencies can harm the accuracy of the model.

To address this issue the project used data augmentation to synthesize images of motor vehicles to reduce the difference in label frequencies. Data augmentation refers to techniques where simple changes can be applied to images to create two different images. These changes may include flipping the image horizontally or skewing the image at an angle to create different images of the same view. While this measure is usually simple to implement, it has to be carefully calibrated to prevent the model from over-fitting. Over-fitting refers to a problem in ML where a model becomes too accustomed to a set of data during training that it performs poorly on new data provided during testing.

The second problem the project faced was data mismatch for the naive model implementation. Since the project relies on the naive model to automate the data labeling process a data mismatch problem would impair the accuracy of the naive model and require more human intervention. Data mismatch is caused when a model is trained by data taken using different hardware configurations.

A third problem faced by the project was the occurrence of polluted data in the collected datasets. Since the camera is left to record videos outdoors, it is not uncommon for birds to perch on the camera stand causing the camera to tilt and change the orientation of the recorded video. A change in video orientation can greatly affect the accuracy of the model and might influence the viability of the project after deployment. This problem was resolved by having a team member monitor and protect the camera while it was recording.

4. Conclusion

This project aims at addressing the issue of Hong Kong's missing motor vehicle problem by creating an ML model to watch busy highways for stolen vehicles, and an accompanying notification system and mobile app to alert LEAs of the stolen vehicles.

This paper concludes by reporting on the project status. The project has successfully managed to collect and label a large amount of data to train an effective object detection model. As explained in section 2.4.a the collected data will enable the project to train for very specific use cases and will adapt the model to suit Hong Kong's unique terrain.

At current the project has also successfully created naive implementations to automate the data labeling process; the information and experience from creating the naive model can be translated to the main project and improvements to the main ML model can be made. As it stands, the project is almost complete with the data labeling stage and, afterwards, will progress to complete the first variation of the ML model.

One of the project's limitations is that it does not have the ability to distinguish if a vehicle is stolen; the model matches the description of a stolen vehicle against a database (see section 2.4.c) and does not independently judge whether a vehicle is stolen or not. It is important to distinguish this limitation, as the model will not be able to alert LEA if no missing vehicle report has been filed.

A second limitation of this project is that it is sensitive to terrain and lighting. The data collected for the project were all taken outdoors with natural lighting. Due to this, the project may not be able to perform well in unknown situations such as underpass tunnels.

A recommendation for future projects could be an extension of the current model to accommodate unknown terrains and lighting. This will widen the field application for this model and will allow more traffic enforcement cameras to watch for stolen vehicles.

A third limitation of the project is that it is specifically calibrated to Hong Kong's traffic. The model may perform very poorly in other countries. This is especially true for countries with different road networks, traffic enforcement camera orientations, and car model distributions. For example, in Hong Kong, it is very rare to see a vehicle of the Toyota Prado model, while the Toyota Prado is relatively common in urban areas of Thailand and Singapore. This may cause issues where the model would not recognize the Toyota Prado as a vehicle.

A recommendation would be to create a model trained on data collected from multiple countries. This would help the model to learn the necessary parameters for adaptation in different environments.

A fourth limitation is that the mobile app interfaces are written in Kotlin for Android phones. While writing the apps in Kotlin allowed for better control over the phone peripherals and resources the apps are not compatible with iOS phones or Windows phones.

A recommendation would be to recreate different versions of the mobile app interfaces to be able to run on multiple devices as this would ensure a greater range of hardware compatibility.