AI Model for Vehicle Identification and Tracking

Team: FYP22059
Presenter: ASIM, Uzair Bin Colangoy [3035603071]
Presenter: Young, Chak Fung (John) [3035468160]
Supervisor: Dr T.W. Chim

- Cover the Background, Motivation and Objectives
- Look into Software Architecture and Methodology with Illustrations
- Go through Data Collection and Data Handling
- Expand on Model Implementation and Project Deployment
- Wrap up with Limitations and Suggestions

Background

- Losing an motor vehicle is a very expensive and stressful experience. Depending on the model of the vehicle the victims may suffer several thousands of dollars worth of damages.
- It is a common and growing phenomenon in urban spaces. Especially countries experiencing rapid urbanization.
- Searching for stolen vehicles can take long periods of time without guarantee of success.

Motivation

- Searching for stolen motor vehicles relies on a manual system.
- Police rely on citizen reports and checking during traffic stop
- Where police have to watch several hours of street camera footage.
  - A very labor intensive process.
  - Is very time consuming.
  - Is prone to human errors.

Objectives

- Our objectives can be broken down into three items:

- Object Detection ML model
- Database server
- Mobile App Interface for Notifications

Project Deployment

- Database Server will be deployed on a web server.
  Where it can interact with http requests

- The Camera App Interface and Notifications App Interface will be deployed on Google Play Store.
  For easy distribution.

Project Deployment - Camera App Interface

- Kotlin Mobile App for Android
- Applies Object Detection model on camera input stream
- Sends information to database server

Project Deployment - ML Model

- Object Detection Model and Optical Character Recognition model will be wrapped in the Camera App Interface
- Models converted using TFLite to run using mobile phones hardware
- Characteristics of vehicles is recognized and sent to database server for testing.
  - Allows for parallelization of ML instances.
  - Reduces the need for power computational severs.

Project Deployment - Database Server

- Holds data about stolen vehicles
- Receives input from cameras and tests for positive match
- Creates a list of positive matches which can be watched by notification apps.

Project Deployment - Notifications App Interface

- Kotlin Mobile App
- Pings database server periodically for updates
- Can notify law enforcement of stolen vehicles spotted
- It can retrieve essential data from database server:
  - Location of camera that spotted the vehicle
  - Time the vehicle was spotted

Limitations

1. Model cannot make independent decisions.
   - Must refer to database server to determine if vehicle is stolen
2. Cannot predict path stolen vehicle will take
3. Model trained on data obtained in Hong Kong
   - Might perform poorly if deployed overseas
4. Mobile App Interface written in Kotlin
   - Incompatible iOS phones.

Recommendations

1. An ML model to predict if vehicle is stolen
   - Watches for suspicious behaviour such as swerving and reckless driving.
2. An ML model to predict paths taken by stolen vehicles.
   - Can use google maps api to analyze traffic and make decisions
3. Collect training data from other cities to supplement training data collected in Hong Kong.
4. Recreate the Mobile App Interface in Swift for iOS compatibility.