BỘ THÔNG TIN VÀ TRUYỀN THÔNG

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Third Report

Foundation Internship

Project Title: Traffic License Recognition

Instructor:	Kim Ngoc Bach
Student Name :	Bui Xuan Hai
Student ID :	B22DCAT105
Lớp :	E22CQCN05-B

Hà Nội - 2025

INTERNSHIP BASE REPORT

I. Project Introduction

This project aims to develop a web application that supports automatic recognition of vehicle license plates from uploaded images or videos. The system uses a deep learning model (trained by myself) to detect and recognize license plate numbers. This can be applied in real-life scenarios such as entry/exit management in parking lots or garages.

Users can interact with the system via a web interface, upload photos or videos, and the system will display the detected license plates. The backend is implemented using Flask, while the frontend is designed using HTML, CSS, and JavaScript (with optional integration of Bootstrap or Tailwind CSS).

II. Key Features

- Allow users to upload vehicle images or videos.
- Automatically detect and recognize license plate numbers.
- Display recognition results in a user-friendly format.
- Possibility to store recognition history or export results.
- Designed for application in garage or parking management systems.

III. Technologies Used

- **Frontend**: HTML, CSS, JavaScript (Bootstrap/Tailwind CSS).
- **Backend**: Flask (Python).
- **AI Model**: YOLO-based license plate detection + OCR model (e.g., PaddleOCR or custom model).
- **Storage**: Local server or database (SQLite/MySQL)

IV. Week 3

Work done

- Convert best.pt,last.pt models into elements in the Python system
- Clone YOLOv5 repo and import manually
- Write a function that receives an image as input → returns the coordinates of the bounding box containing the license plate
- Display the input image along with the drawn bounding box
- Save the output image to the /outputs folder for easy re-checking
- Print out the location and confidence of each detected license plate
- Use about 10 never-before-seen images to test the effectiveness
- Successful detection rate: 80%, mainly missing blurred images or license plates that are too small

Results achieved

- YOLOv5 model can operate stably on images outside the training set
-
- Preprocessing pipeline → detect license plates → return coordinates that work well
-
- Create a solid foundation for the next step of character recognition similar to PaddleOCR

Difficulties encountered

- Some input images are broken, tilted or overexposed, causing the model to miss
-
- YOLOv5 still has errors in images with many noisy objects, e.g. car logos, signs near license plates
-
- Debugging bounding boxes requires manual verification, which is quite time-consuming