BỘ THÔNG TIN VÀ TRUYỀN THÔNG

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

7th 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**: FastAPI (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 6

1. **Weekly goals**

- Complete the connection between frontend and backend
- Deploy the end-to-end system test: from uploading photos → recognizing license plates → displaying results
- Evaluate the accuracy and processing speed of the entire system on real photos

2. **Work done**

- Improve the frontend-backend connection
- Optimize the JavaScript to handle error situations: no photos, photos in wrong format
- Add the "processing" status so users know the system is running
- Smooth the display effect of results after recognition is complete
- Process the returned photos from the backend
- 3. **Results achieved**
- Complete the full processing process from user input photos → AI analysis → displaying results
- The system is stable enough
- Intuitive interface, easy to use, suitable for ordinary users
- 4. **Difficulties encountered**
- Some heavy photos (>4MB) increase processing time → need to resize image before sending

- Debugging API errors is more difficult because it has to handle both frontend and backend at the same time