Self-Driving Car Simulation Project – Reflection Report

Course: CVI620
Semester: Winter 2025
Instructor: Ellie Azizi
Team Members: Airo Nazemirad, Jie Xu, Xuesong Zhou

Project Overview

The goal of this project was to train a Convolutional Neural Network (CNN) to control the steering of a simulated self-driving car. The model takes in images from a front-facing camera and predicts the appropriate steering angle to keep the car on the road. The model was tested in real-time using a simulation environment developed by Udacity and Nvidia.

Approach

1. **Data Collection**
   We used the Udacity Self-Driving Car Simulator in "Training Mode" to manually drive the car and record the dataset. The images were stored along with the corresponding steering angle in a CSV file (driving_log.csv).
2. **Data Analysis & Augmentation**
   A histogram was plotted to understand the distribution of steering angles. To address imbalance and improve generalization, we applied data augmentation techniques such as image flipping (with angle inversion), brightness changes, and zooming.
3. **Data Preprocessing**
   Each image was cropped to remove irrelevant parts (sky, car hood), converted to YUV color space (like Nvidia's model), resized to 200x66 pixels, normalized, and smoothed using Gaussian Blur.
4. **Model Architecture**
   The model was inspired by Nvidia's end-to-end self-driving architecture. It consists of several convolutional layers followed by dense layers. The model was trained using mean squared error (MSE) loss and the Adam optimizer.
5. **Model Training & Evaluation**
   We split the dataset into training and validation sets, tracked performance using loss curves, and saved the final model as model.h5.

6. **Testing**
   The model was integrated into TestSimulation.py, which connects to the simulator using Flask and SocketIO. The car drives autonomously in the simulation based on the model's predictions.

Challenges & Solutions

Challenge	**Solution**
Errors in environment setup	Created a dedicated Anaconda virtual environment (sdcar_env) and installed dependencies one-by-one when the package_list.txt failed.
pip install failures due to incorrect package format	Replaced = with == in package list or installed missing packages individually using pip/conda.
Module errors (cv2, socketio, imgaug)	Installed missing modules manually.
Model not controlling car	Ensured the simulator and script were both running in the **same virtual environment**, and verified that model.h5 was trained and loaded correctly.
Silent failures	Added logging like print("[INFO] Model loaded successfully!") to verify script flow

The challenges were primarily related to **environment setup** and **running the simulator.exe** across different operating systems(Windows/ Mac / Linux). To resolve this, we had to use a **Windows virtual machine**. However, there is still an unresolved error on my device. For details, please refer to the **Error_report.pdf**.

How to Run the Project

1. **Create Virtual Environment:**

   conda create -n sdcar_env python=3.8

   conda activate sdcar_env

2. **Install Dependencies:**

   pip install -r requirements.txt

3. **Collect Data:**
   - Open Udacity simulator → Training Mode
   - Manually drive the car → Record the data
   - Data will be saved to IMG/ folder and driving_log.csv

4. **Train Model:**

   python TrainModel.py

5. **Test the Model:**
   - Launch the simulator in Autonomous Mode
   - In terminal: python TestSimulation.py
   - The car should now drive on its own using the trained model.

Conclusion

This project provided hands-on experience in training a CNN for real-time inference, handling data preprocessing and augmentation, and deploying a model into a simulation. It also taught us practical debugging and environment management skills, crucial for deep learning deployment. Additionally, collaboration and teamwork were crucial for this project.