

|  |  |
| --- | --- |
| Simon Chaisouang  shchaiso@uark.edu  010878218 | Venkata Kumari  vg021@uark.edu 011088021 |

CSCE 50103: Full Stack Deep Learning

Spring 2025

Homework #4

Late Days: 2

**License Plate Detection Using DETR**

**1.Introduction:**

The objective of this assignment is to integrate the two models into a seamless pipeline, allowing for automated processing of images to extract and recognize license plate characters. To make the system accessible and user-friendly, a web-based interface using either Streamlit or Gradio will be developed. By the end of this assignment, we aim to achieve an optimized, fully functional web application with real-time processing capabilities.

**2. Objectives and Requirements**

**Objectives**

1. Integrate license plate detection and character recognition models into a unified pipeline.
2. Develop an interactive web application using Streamlit or Gradio.
3. Ensure the system completes processing within 2 seconds per image on CPU mode.
4. Maintain at least 90% of the performance metrics achieved in previous assignments.

**Requirements**

1. **Efficiency:** Optimize the pipeline for real-time performance.
2. **Usability:** Create an intuitive and user-friendly web interface.
3. **Robustness:** Test the system on diverse images to ensure reliable performance.

**3. End-to-End Pipeline Design**

**Pipeline Overview**

The pipeline consists of the following steps:

1. **Input Image Handling:** Upload an image through the web interface.
2. **License Plate Detection:** Use the trained detection model to localize the license plate in the image.
3. **Character Segmentation:** Segment characters from the detected license plate region.
4. **Character Recognition:** Recognize each character using the trained recognition model.
5. **Result Display:** Return the detected license plate number and display the annotated image.

**Optimization Techniques**

* **Model Selection:** Use lightweight architectures like MobileNet or small ResNet variants.
* **Pruning:** Reduce unnecessary layers and parameters in the model.
* **Quantization:** Apply 8-bit quantization to reduce computational load.
* **Model Distillation:** Train a smaller model to mimic the performance of a larger one.

**4. Web Application Development**

**Framework Selection**

Gradio was selected for its simplicity and flexibility in building user-friendly interfaces.

**Interface Design**

* **Input:** Image upload feature.
* **Output:** Detected license plate number and annotated image.

**5. Testing and Results**

**Test Scenarios**

1. **Diverse Input Images:** Include images with varying lighting, angles, and plate designs.
2. **Error Cases:** Test on images without license plates or with distorted plates.
3. **Performance Metrics:** Measure inference time, accuracy, precision, recall, and mAP.

A screenshot of a computer

Description automatically generated

A close up of a sign

Description automatically generated

A screenshot of a computer

Description automatically generated

****

From the results, it can be shown that the pipeline was successful in detecting the license plate but fell very short in classifying characters. This showcases the limitations of our trained model being solely trained on the MNIST dataset. A more diverse dataset should be considered for further experimentation. A dataset with more diverse perspectives and augmentations can be considered to enhance the robustness of the model’s performance in detecting plates under varying conditions. The pipeline was successful in generating moderate results within the 2 second constraint.

**Challenges and Improvements**

* **Challenge:** Handling low-quality images. **Solution:** Apply preprocessing techniques such as contrast enhancement and noise reduction.
* **Challenge:** Improving inference time. **Solution:** Optimize model parameters and use quantization.

**6. Conclusion**

This project successfully integrates two deep learning models into an end-to-end pipeline for automated license plate recognition. By optimizing the models and employing Gradio for deployment, the system achieves real-time performance and high accuracy, making it suitable for real-world applications. Further improvements, such as incorporating additional training data and advanced optimization techniques, could enhance robustness and efficiency.