Name: Vidhi khandelwal

GWID: G30891152

**Read License Plates from Video Sequences**

With the rapid development of deep learning models, computer vision tasks have become more accessible than ever. One exciting project that showcases the power of computer vision is building an automated system to read license plates from video sequences, such as those from surveillance cameras or dashcams. This project combines object detection, Optical Character Recognition (OCR), and tracking algorithms to deliver a robust, end-to-end license plate recognition system.

In this blog, we’ll walk through the essential components required to build this project, including the YOLO object detection model, the PaddleOCR library for text recognition.

The primary objective is to read license plates from vehicles in video sequences. To achieve this, we’ll:

* Detect vehicles and locate license plates within each frame.
* Use OCR to extract the license plate number.
* Track each vehicle to associate the plate numbers accurately across frames.

The final product will output annotated video footage with each vehicle’s license plate information, providing real-time monitoring for applications like traffic management, parking systems, or toll booths.

**1. Dataset**

For this project, I have used the following dataset:

<https://universe.roboflow.com/roboflow-universe-projects/license-plate-recognition-rxg4e/dataset/4>

Vehicle detection data: Images of vehicles with bounding boxes labelling vehicle classes such as cars, trucks, and motorbikes.

License plate data: Images of license plates annotated with bounding boxes and corresponding text labels.

Train set: 21174

Valid set: 2048 images

Test set: 1020 images

**2. Pretrained Model YoloV8:**

For the project, I employed a pretrained YOLOv8 model, renowned for its accuracy and efficiency in real-time object detection tasks. This choice not only accelerates development but also ensures a high level of performance out of the box.

YOLOv8, an advancement of the YOLO architecture, excels in real-time object detection with its deep convolutional backbone network, dynamic scaling, focal loss optimization, and multi-scale prediction strategy. These technical innovations enable precise detection of objects of varying sizes and aspect ratios while maintaining high speed and accuracy, making YOLOv8 a leading solution in computer vision tasks.

The primary YOLOv8 model was configured to track vehicles within each video frame, filtering by vehicle classes (like cars, trucks). The second model detected license plates within these vehicle regions, which improved detection accuracy by narrowing the search area within each frame.

**3. Model Training and Optimization**

Training Parameters: Both YOLOv8 models were fine-tuned using custom datasets with relevant classes (vehicles and license plates). Hyperparameters like learning rate, confidence thresholds, and intersection-over-union (IoU) thresholds were optimized for the best performance.

Model Fusion: To optimize runtime, model fusion was applied so that the vehicle and license plate detection models could run in parallel, efficiently analysing each frame without bottlenecking the system.

**4. Optical Character Recognition (OCR) using PaddleOCR:**

The license plate was detected inside a vehicle bounding box. PaddleOCR was then used to process the extracted plate picture in order to identify and decipher the text characters on the plate.  
**Character Conversion and Formatting:** A dictionary-based mapping was utilized to differentiate visually identical characters (such as "O" vs. "0" and "I" vs. "1") in order to address common misinterpretations. Following recognition, the text was converted to adhere to standard license plate structures.

**5. Workflow in the System and Processing in Real Time  
Frame Processing Pipeline:** The YOLOv8 vehicle and plate detection models were applied one after the other to each frame of the video feed. The discovered plate region was then cropped by the pipeline before being sent into the OCR model.  
**Result Overlay:** The data was superimposed on the video frames following the extraction of the license plate text. To improve interpretability, bounding boxes were drawn around cars and license plates, with plate numbers shown above each vehicle.

**6. Results**

****



The model achieved high accuracy in detecting license plates across various frames, even in challenging conditions. Test results showed that YOLOv8 accurately tracked vehicles and isolated license plates with minimal false positives.

PaddleOCR performed well in reading license plate characters, reaching approximately 90% accuracy in varied lighting and angles. The character mapping system further improved OCR accuracy.

**7. Challenges and Lessons Learned**

**Lighting and Reflection:** Bright sunlight or nighttime reflections affected detection. This challenge was partially addressed by data augmentation and enhancing model generalizability.

**Character Similarity in OCR:** Distinguishing between similar-looking characters was a frequent issue. The customized character dictionary mapping was instrumental in mitigating this problem.

**Real-Time Constraints:** Achieving low latency while processing high-resolution frames was challenging. Model optimization techniques like fusion and efficient batch processing were essential in overcoming this.

**Conclusion**

In conclusion, our journey into vehicle detection using the YOLOv8 model has shown both the challenges and the potential of this solution. By leveraging advanced computer vision techniques and datasets like the license-plate-recognition dataset, the blog demonstrates the transformative impact of precise object detection. Despite the complexities inherent in detecting vehicles accurately, the exploration has showcased the effectiveness of thoughtful parameter tuning and optimizer selection in improving model performance.