Visionary Plates: Advancing License Plate Detection Models

Team AI Avengers

- Andrew
- Abdelrahman
- Saifullah
- Kashaf Jamil
- Mohit
- Wasay

Project Overview

This project aims to design and implement a real-time license plate detection system capable of accurately detecting and localizing license plates in images. The system is robust and can handle various environmental conditions, including different lighting conditions, vehicle orientations, and background clutter. The goal is to provide reliable results for further processing or use in applications such as traffic monitoring, parking management, or law enforcement.

Detail of Hackathon

Dataset Creation using Label Studio for License Plate Detection

In our quest to improve license plate recognition through object detection, we carefully curated and labeled a diverse dataset using Label Studio. This dataset is instrumental in fine-tuning our models to achieve optimal performance.

Data Collection

We began by collecting a wide range of images depicting various real-world scenarios in which license plates might be encountered. These scenarios include different lighting conditions, vehicle orientations, and environmental backgrounds.

Labeling Process

- **Image Selection:** Relevant images were chosen to ensure a comprehensive representation of different scenarios, including daytime, nighttime, bad weather, etc.
- **Labeling with Label-Studio:**
  - We utilized the "Object Detection with Bounding Boxes" template in Label Studio.
  - Each selected image was loaded into Label Studio's interface.
  - Annotators marked the license plates in the images by drawing bounding boxes around them, providing precise annotations

Review and Quality Assurance:

Labeled data went through a review process to ensure accuracy and consistency. Any discrepancies or inaccuracies were addressed and corrected.

Dataset Statistics

- Total images annotated: 367
- Average number of license plates per image: 1-2
- Dataset diversity:
  - Day and night scenarios
  - Varied weather (rain, fog)
  - Frontal, side, and rear views of vehicles
  - Different country license plates and fonts

Dataset Samples

Fine-Tuning the Model

Leveraging Lambda Cloud for Enhanced Model Training

Infrastructure Optimization

We utilized Lambda Cloud's powerful infrastructure to expedite the model training process

Model Fine-Tuning and Output Analysis

- Pretrained Model: Fine-tuned the YOLOv8 model using the COCO dataset as a starting point.
- Output Format: The model generates outputs in the form of [confidence, position0, 1, 2, 3].

Training Insights

Training Duration: Approximately 12 hours for model training, demonstrating efficiency in model optimization and fine-tuning.

Result Packaging

Result File: Encapsulated the trained model and its output in the 'run.zip' file.

Training the Model

We used the YOLOv8 architecture for license plate detection. The model was fine-tuned based on a single class (license plates), and the 80 classes from the COCO dataset were removed in the YOLOv8 configuration file (yml)

Training Parameters

- **Image Size:** 512×512 pixels
- **Number of Epochs:** 10 (adjust based on dataset size and convergence)
- **Learning Rate:** lr_scheduler
- **Optimizer:** smart_optimizer

Training Process

The training process involved iterating through the dataset, feeding images and labels to the model, calculating loss, and optimizing the model using backpropagation. Training metrics were recorded for evaluation.

Hurdles and Solutions

Problem 1: Dataset Preparation

Problem: Acquiring and annotating the dataset with license plate bounding boxes was time-consuming and challenging.

Solution: Careful data collection, labeling, and quality control processes were implemented to create a reliable dataset.

Problem 2: Model Fine-Tuning

Problem: Fine-tuning YOLOv8 for a single class, license plates, required selecting suitable hyperparameters and managing overfitting.

Solution: Extensive experimentation, hyperparameter tuning, and continuous monitoring of the training process helped achieve the desired model accuracy.

Problem 3: Metric Interpretation

Problem: Interpreting and optimizing evaluation metrics for object detection tasks, such as precision and recall, was complex.

Solution: In-depth understanding of evaluation metrics and iterative model improvements were employed to balance precision and recall for real-world scenarios.

Problem 4: Initial Model Performance

Problem : Overcoming challenges related to night vision scenarios. Addressed suboptimal performance by implementing Sharpening and Gamma Control methods.

Solution: Experimented with YOLOv5 and traditional computer vision methods, eventually transitioning to YOLOv8 **for improved performance.**

Pitch

Want to make a presentation like this one?

Start with a fully customizable template, create a beautiful deck in minutes, then easily share it with anyone.

Create a presentation (It's free)