Licence Plate Recognition

Pre-processing and Data Analysis

License Plate Detection Data

- This set includes 900 images, each containing a single car with its corresponding license plate.
- Labels consist of bounding **box coordinates** (xmin, ymin, xmax, ymax).
- These annotations help **YOLO** detect where the license plate is.

License Plate OCR Data

- This set also comprises 900 images, but it exclusively features license plates.
- After detecting license plates, the **extracted cropped plate images** are used for character recognition.
- Each cropped plate has **text labels** corresponding to the number plate.

Pre-processing Steps

- **Image resizing**: YOLO uses **416×416 input images**.
- **Normalization**: Images are **scaled to [0,1]** (YOLO) or **standardized** for the **OCR model**.
- **Grayscale conversion**: The OCR model often benefits from **grayscale images**.

Data Exploration

Exploring License Plate Detection Data

- **Bounding Box Visualizations.**
- Plot images with **YOLO-detected license plates** to verify correctness.
- Overlay bounding boxes on images to confirm if the model detects plates correctly.

Exploring License Plate OCR Data

Note - We have observed that all licensed number plates contain a fixed set of characters, where the numeric characters include 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9, along with an additional character denoted as "T".

Character Distribution

- **Count occurrences** of each character (0-9 and T).
- Check for **class imbalance** (e.g., some numeric characters appearing more frequently).

Character Segmentation

- Visualize how **characters are segmented** from plates.
- Ensure the characters are properly extracted before training OCR.

Model Building

You have two models working together :

YOLO for License Plate Detection

Model: YOLO
Training Process:

- Clone **YOLOv5** repository and install dependencies.
- **Prepare dataset** in YOLO format.
- Specifying image size, batch size, epochs, dataset path, and pre-trained weights.
- After training, the **best model is saved** for inference.
- Then **load the YOLO model**.
- Convert input images into **YOLO blob format**.
- Forward pass through the network to get **bounding boxes**.
- Apply **Non-Maximum Suppression (NMS)** to **remove overlapping boxes**.
- **Extract the detected license plate** as a cropped image.
- IPYNB File

OCR Model for Character Recognition

Model: CNN-based character recognition model.

Training Process:

- **Input:** Cropped license plate images (**28×28** for each character).
- **CNN** extracts **character features** and **classifies** them into **0-9 and T**.
- The final output is a **sequence of predicted characters**.
- IPYNB File

Final Pipeline Flow

- **YOLO detects** the **license plate** in a car image.
- Extracted plates are **cropped** and **saved**.
- The **OCR model recognizes characters** from the cropped plate.
- Final output is a **readable license plate number**.
- IPYNB File

Results

- We have executed the combined License Plate Detection and OCR model on the provided test dataset
- The output has been saved in a CSV file.
- CSV File