Data distillation for Automatic License Plate Recognition System

This file

Dataset Overview

We used a dataset composed of European (EU) license plate images. Before training, significant preprocessing was required, including fixing label formats, standardizing filenames, and reorganizing the directory structure to facilitate proper splitting and model training.

The initial dataset consisted of:

- **Images:** 108 `.jpg` files
- **Labels:** 108 `.txt` files (one per image)

It was split with the following ratio:

- **Training:** 80%
- **Validation:** 20%

Resulting dataset sizes:

- **Training set:** 86 image (`.jpg`) and label (`.txt`) pairs
- **Validation set:** 22 image (`.jpg`) and label (`.txt`) pairs

File Standardization

All image filenames were standardized to follow the pattern `eu_001.jpg`, `eu_002.jpg`, …, `eu_108.jpg`. Correspondingly, each image has a matching annotation file named `eu_001.txt`, `eu_002.txt`, etc.

Label Conversion (to YOLO Format)

Since we utilized a `YOLO` model, the bounding box annotations needed to conform to its specific format. The original annotations were provided in a format representing pixel coordinates `(x_min, y_min, width, height)`.

These were converted to the `YOLO` format: `(class_index, x_center, y_center, width, height)`, where:

- `class_index` is 0 (as we only have one class: license plate).
- `x_center`, `y_center`, `width`, and `height` are normalized values between 0 and 1, relative to the image's total width and height.

A `python` script was used to iterate through each original `.txt` annotation file. For each annotation, the script loaded the corresponding image to obtain its dimensions. It then calculated the center coordinates (`x_center`, `y_center`) for the bounding box, normalized all four coordinate values, and wrote the results back into the `.txt` file in the required `YOLO` format.

Train/Validation Split

The standardized image and label pairs were randomly shuffled and then split into training and validation sets. 80% of the data was allocated for training the model, and the remaining 20% was reserved for validation.

After splitting:

- **Training pairs:** 86
- **Validation pairs:** 22