1. Project Overview

The objective is to develop a specialized computer vision pipeline for detecting and classifying prohibitory traffic signs from real-world images.

2. Technical Pipeline Stages

2.1 Data Acquisition and Preprocessing

- **Source**: Images were sourced from Google Street View.
- **Color Space**: Initial segmentation used the **HSV** model to isolate color (Hue) from lighting (Value) for consistency.

2.2 Multi-Strategy Detection

Four approaches handle varying luminosity:

- **Standard**: For clear lighting.
- **Luminosity-Adjusted**: For overexposed images.
- **Dark Image**: Uses **CLAHE** for low-light contrast.
- **Hough Circle Transform**: Geometric fallback for color-unreliable cases.

2.3 Object Refinement and Segmentation

- **Dynamic Bounding Boxes**: A custom function "tightens" detection boxes around red pixels.
- **Vertical Signal Separation**: Automatically divides vertically stacked signals using aspect ratio analysis.

2.4 Data Augmentation Pipeline

To simulate real-world conditions, I implemented a comprehensive augmentation strategy including geometric transformations (random rotations (+-15º), scaling, and affine transformations). I also applied photometric transforms (brightness and contrast adjustments) and noise filters like bilateral blurring to mimic motion blur or low-resolution sensor data.

2.5 Classification via Transfer Learning

- **Architecture**: Utilized **MobileNetV2**, pre-trained on ImageNet.
- **Fine-tuning**: Frozen base model training followed by deep-layer adaptation to specific pictograms.

3. Numerical Classification: Challenges and Failures

3.1 CNN Failure (SVHN Model)

- **Problem**: A CNN trained on the SVHN dataset suffered from **"Digit Collapse"**.
- **Outcome**: Due to resolution loss at 32 x 32 pixels and dataset imbalance, the model simplified curves (like "3" or "0") into vertical strokes, predicting **"1"** for almost all inputs (e.g., "11" for a 30 km/h sign).

3.2 Template Matching Failure

- Problem: During the transition to template matching, the numbers extracted from the signals were of very bad quality, or in many instances, no numbers were detected at all.

- Outcome: The accuracy of the template matching was poor.

4. Challenges Encountered and Resolved

Challenge	Solution
Luminosity Variation	Specialized detection functions and CLAHE enhancement.
False Positives	Circularity Descriptors and white-center validation.
Dataset imbalance	Managed class weighting and targeted data augmentation for categories with fewer samples (e.g., Category F).
Digit Segmentation	Implemented circular masking to get the inner part of the sign.

5. Why not YOLO?

- **Small Dataset Performance**: This "Segment-then-Classify" approach achieves high precision with significantly fewer annotated images than YOLO requires.
- **Granular Control**: Allows manual tuning of mathematical filters (Hough, Circularity) for better transparency and control over signal detection.

Code used:

Hough circles: https://docs.opencv.org/4.x/da/d53/tutorial_py_houghcircles.html

Contours, area, perimeter:
https://docs.opencv.org/4.x/dd/d49/tutorial_py_contour_features.html

CLAHE: https://docs.opencv.org/4.x/d5/daf/tutorial_py_histogram_equalization.html

Bilateral filtering: https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html

Canny: https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html