Traffic Sign Detection

Rehan Sherawat
230123077
Mathematics Department, IITG
s.rehan@iitg.ac.in

Vansh Gandharva
230102102
EEE Department, IITG
g.vansh@iitg.ac.in

Mit Narodia
230123039
Mathematics Department, IITG
m.narodia@iitg.ac.in

***Abstract*—YOLO (You Look Only Once) is an algorithm based on deep neural networks with real-time object detection capabilities. This state-of-the-art technology is widely available, mainly due to its speed and precision. Since its conception, YOLO has been applied to detect and recognize traffic signs, pedestrians, traffic lights, vehicles, and so on. The goal of this project is to systematically use the YOLO object detection algorithm, and apply it to traffic sign detection and recognition systems.**

***Index Terms*—YOLO; traffic sign detection and recognition; road accidents; systematic literature review; object detection; computer vision**

I. Introduction

Automatic driving technology, integrating artificial intelligence, the Internet of Things, and automotive innovations, is pivotal in advancing intelligent transportation and enhancing global traffic interconnectivity. It not only boosts convenience and safety by preemptively alerting drivers of deviations or obstacles but also reduces the human error responsible for 90% of traffic accidents. Automatic driving capabilities, including traffic efficiency improvements and autonomous parking, mark significant progress in the field.

Traffic signs, crucial to safe driving and effective road navigation, play a vital role in this ecosystem. Traffic Sign Recognition (TSR) systems enable advanced applications in automatic driving, ADAS (Advanced Driving Assistance Systems), and Urban Intelligent Transportation Systems. They provide valuable information on road conditions, regulate traffic flow, and standardize driver behavior. This research addresses TSR by dividing it into two main processes: detection and classification. Key sign categories—prohibitive, warning, and informative—feature distinct shapes and colors but face recognition challenges due to environmental factors like lighting and weather conditions.

To tackle these challenges, this report presents a systematic literature review (SLR) on the detection and recognition of traffic signs using the YOLO object detection algorithm. In this context, a traffic sign serves as a visual guide to convey information about road conditions, potential hazards, and other essential details for safe road navigation

II. Literature Review

A. Related Work

Traffic sign detection methods are primarily divided into classical models and deep learning (DL) approaches. Classical models use manually selected features like color and shape, often in the HSV color space, and employ algorithms such as HOG with SVM for reliable detection and classification. These methods, while precise, struggle with scene variability. DL methods, however, improve robustness and can be split into two types: region-based and regression-based. Region-based models, like R-CNN and Faster R-CNN, excel in accuracy by generating regions of interest but are computationally intensive. Regression-based models, such as YOLO and SSD, offer faster, real-time detection by integrating localization and classification into a single network, though they may sacrifice some precision. While two-stage models provide better accuracy, one-stage models like YOLO are ideal for real-time traffic sign detection due to their efficiency.

B. Problem Statement

The general objective is to obtain a YOLO based deep CNN model for Traffic Sign Detection and Classification which can detect and classify traffic signs simultaneously in real time with performance comparable to that of R-CNNs and fast R-CNNs

III. Methodology and Framework

Works such as [1] have given remarkable results on traffic sign detection tasks. However, they are not real time which is a key factor to be considered while evaluating the performance. Our model is built with YOLO Architecture as base focusing on fine-tuning the existing model based on the traffic signs.

A. Architecture

YOLO (You Only Look Once) is an advanced object detection system that operates with high speed and efficiency by employing a single convolutional neural network. The architecture allows for predicting bounding boxes and class probabilities in one forward pass through the network, making it particularly suitable for real-time applications. Now, we provide a more detailed breakdown of the YOLO approach:

- 1) Grid Division: The input image is divided into an S × S grid. Each grid cell is responsible for predicting bounding boxes for objects whose centers fall within that cell.
- 2) Bounding Box Predictions: Each grid cell predicts k bounding boxes, which are defined by a quintuple (x, y, w, h, cf d):
  - (x, y): The coordinates represent the center of the bounding box, normalized relative to the grid cell.

- w and h: These values denote the width and height of the bounding box relative to the entire image, allowing for flexible object sizes.
- cf d: The confidence score for each bounding box, defined as the product of two factors: Pr(Object), which indicates the probability of an object being present in the bounding box, and IOUtruth pred, which measures the intersection over union between the predicted box and the ground truth box. The value of Pr(Object) is 1 if a grid cell contains part of a ground truth box and 0 otherwise.

- 3) Intersection over Union (IOU): This metric quantifies the overlap between the predicted bounding box and the ground truth box, ranging from 0 (no overlap) to 1 (perfect overlap).
- 4) Class Scores: Each grid cell predicts a single set of class scores (C) for all bounding boxes in that cell, indicating the likelihood of each class being present. This is different from other methods that might predict class scores for each bounding box individually.
- 5) Output Vector: The output of the YOLO network for an image is a tensor of dimensions S × S × (5B + C), where B is the number of bounding boxes predicted by each grid cell and C is the number of classes. The terms in this tensor provide the bounding box coordinates, confidence scores, and class probabilities.

B. Data Set

This traffic sign dataset includes a total of 4,969 samples divided into three parts: Training, Validation, and Test sets. The dataset is organized to support efficient model development and evaluation for traffic sign recognition tasks. Each sample corresponds to one of 15 distinct traffic sign classes:

Green Light Red Light Speed Limit 10 to 120(interval of 10) Stop Dataset Characteristics and Preprocessing This dataset includes images captured from various viewpoints in diverse urban settings, introducing challenges typical of real-world scenarios, such as lighting variations, partial occlusions, and background clutter. To standardize input, each image is resized to 448x448 pixels before being fed into the neural network, ensuring uniformity across training, validation, and testing phases. Each image is also annotated with precise bounding box coordinates (x-coordinate, y-coordinate, width, and height), allowing the model to accurately localize and classify traffic signs.

Potential Use Cases Autonomous Vehicle Navigation: The model can be integrated into autonomous driving systems to identify and interpret traffic signs accurately, enabling vehicles to follow speed limits, stop signs, and traffic lights. This is essential for safe, rule-abiding navigation on public roads, enhancing overall road safety.

Traffic Rule Compliance: Driver assistance systems can leverage this model to alert drivers of traffic rule violations, such as exceeding speed limits or running a red light. This proactive approach encourages compliance, reduces accident risks, and promotes safer driving practices.

Road Safety Training Programs: Driving schools and automotive companies can use this model to create simulations and educational tools. These resources can train new drivers to recognize and correctly respond to various traffic signs, improving their road safety knowledge.

Smart City Infrastructure: City authorities could implement this model in connected IoT systems, such as CCTV networks, to monitor traffic in real-time. The model could help identify areas prone to rule violations, allowing for targeted improvements in road safety infrastructure.

Road Network Analysis: Transportation researchers can use this model to analyze the distribution and visibility of traffic signs across different locations. Insights from this analysis can guide decisions on where to add or enhance signage to ensure clarity and safety for road users.

IV. Evaluation Metric - Mean Average Precision (mAP)

Mean Average Precision is a metric used to evaluate object detection and classification models. To calculate the mAP for a set of detections, the interpolated average preci- sion is calculated for each class, and a mean is calculated over it. For each class, the Average Precision is calculated using the area under the PR(Precision-Recall) curve for the predictions. The PR curve is constructed by associating each detection to its most overlapping ground truth object instance. Detections whose IOU with the ground truth above the threshold are considered as True Positives while the others are said to be False PositivesNext, two metrics called Precision and Recall are calculated as follows.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

$$\text{Recall} = \frac{TP}{\text{No. of Ground Truth Boxes}} \tag{2}$$

The IOU or Intersection over Union of predicted bounding box B_p and ground truth bounding box B_{gt} is defined as follows.

$$IOU = \frac{\text{Area of overlap of } B_p \text{ and } B_{gt}}{\text{Area of union of } B_p \text{ and } B_{gt}} \tag{3}$$

The average precision (AP) is computed by averaging the precision values on the Precision Recall curve where recall is in the range $[0, 0.1 \dots 1]$.

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, \dots, 1\}} p_{\text{interp}}(r) \tag{4}$$

The precision at each recall level r is interpolated by determining the maximum precision measured for a method for which the corresponding recall value exceeds r.

$$p_{\text{interp}}(r) = \max_{\tilde{r}:\tilde{r} \geq r} p(\tilde{r}) \tag{5}$$

where $p(\tilde{r})$ is the measured precision at recall $\tilde{r}$.

Now the mAP is calculated as follows.

$$mAP = \frac{\sum_{i \in \text{classes}} AP_i}{\text{Total no. of classes}} \qquad (6)$$

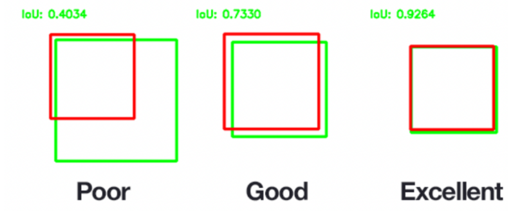

Fig. 1. Comparison of different IOUs

V. Work Done

This section describes the work done by us during the course of the project. We first imported the the existing YOLOv8n architecture and trained it on our data set to define weights and parameters for our model. During the training period, the model also generated results and analysis in form of graphs that are discussed below.

A. Results And Analysis

This section describes the results we have got up till now.

1) Precision analysis: The table describes the various metrics values for various traffic signs in our dataset.

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95
all	801	944	0.943	0.900	0.960	0.834
Green Light	87	122	0.901	0.747	0.867	0.538
Red Light	74	108	0.858	0.704	0.823	0.501
Speed Limit 100	52	52	0.933	0.942	0.984	0.892
Speed Limit 110	17	17	0.788	0.941	0.964	0.893
Speed Limit 120	60	60	0.973	0.950	0.993	0.917
Speed Limit 20	56	56	0.986	0.982	0.986	0.874
Speed Limit 30	71	74	0.963	0.959	0.983	0.921
Speed Limit 40	53	55	0.931	0.964	0.989	0.888
Speed Limit 50	68	71	0.960	0.859	0.970	0.865
Speed Limit 60	76	76	0.958	0.900	0.965	0.877
Speed Limit 70	78	78	0.981	0.962	0.988	0.909
Speed Limit 80	56	56	0.981	0.916	0.979	0.862
Speed Limit 90	38	38	1.000	0.783	0.949	0.802
Stop	81	81	0.983	0.988	0.993	0.936

TABLE I
PERFORMANCE METRICS FOR DIFFERENT CLASSES.

2) Precision-Recall Curve: For visual analysis, we use the following Precision-Recall curve: The Precision-Recall (PR) Curve shown in the graph illustrates the performance of a traffic sign detection model across various traffic sign classes. Each curve represents the precision and recall relationship for a particular class of traffic signs, such as different speed limits, stop signs, and traffic lights. The precision-recall trade-off is an important indicator of how well the model balances false positives and false negatives as thresholds change.

For each class, the model's performance is summarized by the area under its PR curve, with higher areas indicating better precision and recall balance. The mean Average Precision (mAP) at a threshold of 0.5 for all classes is 0.959, showing high overall accuracy. The ”Stop” sign class achieves the highest precision-recall area, indicating exceptional accuracy in detecting stop signs, while other classes, such as ”Green Light” and ”Red Light,” have lower values, showing potential areas for improvement.

Overall, this PR Curve provides a detailed analysis of the model's ability to detect each traffic sign category, helping to identify classes where precision and recall could be further optimized.

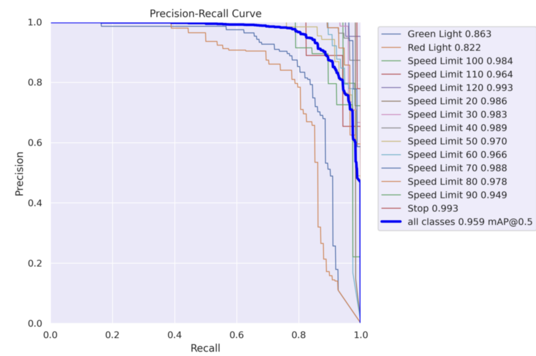

Fig. 2. Precision-Recall Curve

3) Summary Of Overall Results: The overall precision (B) for the model is approximately 0.9426. The recall (B) is around 0.8998. The mean Average Precision at 0.5 IoU threshold (mAP50(B)) is about 0.9595, showing high detection accuracy. The mean Average Precision across IoU thresholds from 0.5 to 0.95 (mAP50-95(B)) is 0.8340, suggesting strong generalization across varying detection thresholds.

In this context, ”(B)” likely stands for ”Box,” referring to bounding box metrics. Bounding box metrics assess the precision, recall, and mean Average Precision (mAP) specifically in relation to the detection and localization of objects (in this case, traffic signs) within bounding boxes drawn around each detected object.

References

[1] M. Flores-Calero, C. A. Astudillo, D. Guevara, J. Maza, B. S. Lita, B. Defaz, J. S. Ante, D. Zabala-Blanco, and J. M. Armingol Moreno, “Traffic Sign Detection and Recognition Using YOLO Object Detection Algorithm: A Systematic Review,” Published: 17 January 2024

[2] R. K. Megalingam, K. Thanigundala, S. R. Musani, H. Nidamanuru, and L. Gadde, “Indian traffic sign detection and recognition using deep learning,” Published: 26 June 2022