**DEEP LEARNING**

**CSCI 5931**

**SPRING 2023**

PROJECT REPORT

ON

REAL TIME ROAD SIGN DETECTION, RECOGNITION AND DRIVER GUIDANCE SYSTEM

SUBMITTED BY

RAGHAVENDRA SHANTHAMRAJU

TARUN REDDY NERELLA

ANUDEEP KALITKAR

**ABSTRACT**

This project aimed to develop a real-time road sign detection, recognition, and driver guidance system to enhance road safety and provide accurate information to drivers. Utilizing a two-layer architecture, the first layer detects road signs using the YOLOv5 model trained on the LISA dataset, while the second layer classifies detected signs among 47 types using a custom dataset and computer vision techniques. Through hyperparameter tuning, architectural adjustments, and evaluation using specific metrics, the system demonstrates promising results in real-time road sign detection and recognition. The developed system has the potential to improve road safety and contribute to more efficient transportation systems.

**CONTENTS**

Abstract 2

1. Introduction 4

2. Literature Review 5

2.1 Related Work

2.3 Problem Statement

3. Methodology and Framework 8

3.1 Architecture

3.2 Dataset

3.3 YOLO

3.4 Model Optimization and Evaluation

4. Implementation 13

4.1 Google Colab and Google Drive Integration

4.2 Individual Model Development

4.3 Model Training Pipeline and Ensemble Solution

4.4 Hyperparameter Tuning and Model Selection

4.5 Model Evaluation and Performance Analysis

4.6 Modifying Loss Function

5. Results 15

6. Future Work 17

7. Conclusion 18

8. Reference 19

1. **INTRODUCTION**

In the realm of road sign detection and recognition, two primary phases exist: identifying potential traffic signs in images (detection) and assigning meaning to each detected region (classification). Various techniques exist for detection, such as color-based, shape-based, and feature-based methods. Classification can be performed using template matching, support vector machines, or k-nearest neighbors, among others. Traditional approaches, however, have limitations like sensitivity to noise, occlusion, and lighting changes, and the need for manual feature extraction.

To address these issues, deep neural networks (DNNs) have become popular in road sign detection and recognition. These networks can learn complex features from data and offer better accuracy and robustness. Convolutional neural networks (CNNs) and You Only Look Once (YOLO) are popular DNNs for this purpose. The latest version, YOLOv5, uses a PyTorch framework and a CSPNet backbone network, improving feature fusion and multi-scale prediction.

We utilize the LISA Traffic Signs Dataset for road sign detection and recognition, which contains 47 different US traffic sign types. To overcome limitations, we generate a synthetic dataset for the second layer of our system, which classifies candidate regions into traffic sign types using a CNN. We apply computer vision techniques for tasks like image augmentation and transformation.

For this project, we propose using YOLOv5 as the primary model due to its advantages in speed, accuracy, and scalability. Additionally, we plan to implement data augmentation techniques to enhance the quality and variety of our training data.

1. **LITERATURE REVIEW** 
   1. **Related Work**

The growth of autonomous vehicles and intelligent transportation systems is remarkable. A key component for these systems is the precise, real-time detection and classification of traffic signs. In this brief overview, we discuss two studies that contributed to traffic sign detection and classification using machine learning techniques, particularly deep learning.

Ciuntu and Ferdowsi (2020) looked into various techniques for real-time traffic sign detection and classification using a portable system. They concentrated on speed limit signs, crucial for road safety and following traffic regulations. They employed a convolutional neural network (CNN) for the detection and classification of these signs, showing deep learning's potential in tackling traffic sign recognition challenges.

Furthermore, they incorporated a unique optical character recognition (OCR) technique to improve their system's accuracy. By combining CNNs with OCR methods, they demonstrated the potential of a combined approach for traffic sign recognition tasks. The data used in this work were derived from multiple sources, emphasizing the importance of using diverse and representative data for training and testing machine learning models in traffic sign recognition.

Zhu and Yan (2022) investigated the challenges linked to traffic sign detection and recognition in the context of ITS and autonomous vehicles. They identified that traditional visual object recognition techniques, such as color and edge feature extraction, have limitations when dealing with the diverse conditions present in real-world environments. These limitations can lead to reduced accuracy and reliability of traffic sign recognition systems, which is undesirable for safety-critical applications like autonomous driving.

To tackle these challenges, the authors suggested using deep learning-based visual object recognition techniques, specifically convolutional neural networks (CNNs), to overcome traditional methods' limitations. CNNs have shown great success in various computer vision tasks, making them a promising approach for traffic sign detection and classification. In their study, Zhu and Yan conducted an experiment to evaluate YOLOv5's performance on a dataset for Traffic Sign Recognition (TSR). This experiment aimed to showcase the effectiveness of state-of-the-art deep learning models, like YOLOv5, for visual object recognition tasks, including traffic sign detection and classification.

Their experiment's results demonstrated the potential of deep learning techniques to achieve high levels of accuracy and efficiency in traffic sign recognition tasks. By using advanced deep learning models like YOLOv5, the researchers showed that it is possible to develop traffic sign recognition systems that are robust to various conditions and perform well in real-world scenarios.

In summary, these two studies show the increasing importance of deep learning techniques, particularly CNNs, in traffic sign detection and classification. Both studies highlight the benefits of using deep learning models for traffic sign recognition tasks, such as improved accuracy, robustness to diverse environmental conditions, and real-time processing capabilities. As research in this field continues to progress, we can expect new techniques and architectures to emerge, providing enhanced capabilities for traffic sign detection and classification in the rapidly evolving domain of autonomous vehicles and intelligent transportation systems.

* 1. **Problem Statement**

The increasing number of vehicles on the road in today's society leads to a higher number of accidents. These accidents can be attributed to various factors, such as insufficient infrastructure, extreme weather conditions, and carelessness. One contributing factor is that individuals often fail to notice or comply with traffic signs. Despite traffic signs being strategically placed and essential for safety, many drivers struggle to interpret them correctly. Furthermore, some drivers might not understand the meaning behind the majority of traffic signs. In response to this issue, the aim is to develop a driver assistance system capable of detecting, recognizing, and explaining the identified signs.

Developing a real-time road sign detection, recognition, and driver assistance system involves addressing several challenges related to dataset collection and augmentation, model selection and optimization, and system integration. Successfully addressing these challenges is crucial for ensuring the system's effectiveness in enhancing road safety, promoting smooth traffic flow, and providing drivers with timely information and guidance.

1. **Methodology and Framework**
   1. **Architecture**

The design of the real-time road sign detection, recognition, and driver assistance system is composed of two primary layers: detection and classification.

**Detection Layer (Layer 1):** The detection layer's primary function is to identify road signs' presence and location within images or video frames. For this purpose, it utilizes the YOLOv5 (You Only Look Once) object detection model, which boasts high accuracy and real-time performance. The YOLOv5 model features a backbone network (CSPDarknet53) for extracting features and a neck network (PANet) for fusing features and making predictions. This architecture is optimized to process input images in a single pass, making it highly efficient for real-time applications.

**Classification Layer (Layer 2):** The classification layer takes the cropped images of detected road signs from the detection layer and sorts them into one of 47 distinct road sign types. To achieve this, it employs a ResNet-based neural network, renowned for its excellent performance in image classification tasks. The ResNet architecture utilizes residual learning and shortcut connections to mitigate the vanishing gradient issue, allowing for deeper and more accurate networks.

* 1. **Dataset**

**Dataset for Layer 1 (Detection):** The detection layer makes use of the LISA (Laboratory for Intelligent and Safe Automobiles) dataset, which is a popular dataset for traffic sign detection and recognition tasks. The LISA dataset comprises annotated images of traffic signs captured under various lighting conditions, weather conditions, and perspectives, making it ideal for training sturdy road sign detection models. With its diverse range of traffic signs, the dataset allows the detection layer to generalize effectively across different traffic sign types.



Fig. Image from LISA Dataset

**Dataset for Layer 2 (Classification):** The classification layer employs a custom dataset that contains images of 47 distinct types of road signs. This dataset was created by gathering images from various sources, including web scraping from<http://www.trafficsign.us/>. However, the initial number of images in the dataset was inadequate for training a high-performing classification model. To address this issue, computer vision techniques were used to generate more images from the limited dataset. Image manipulations, such as image transformations, Gaussian blur, salt and pepper noise, and median filtering, were applied to augment the dataset and enhance the model's ability to accurately classify road signs.

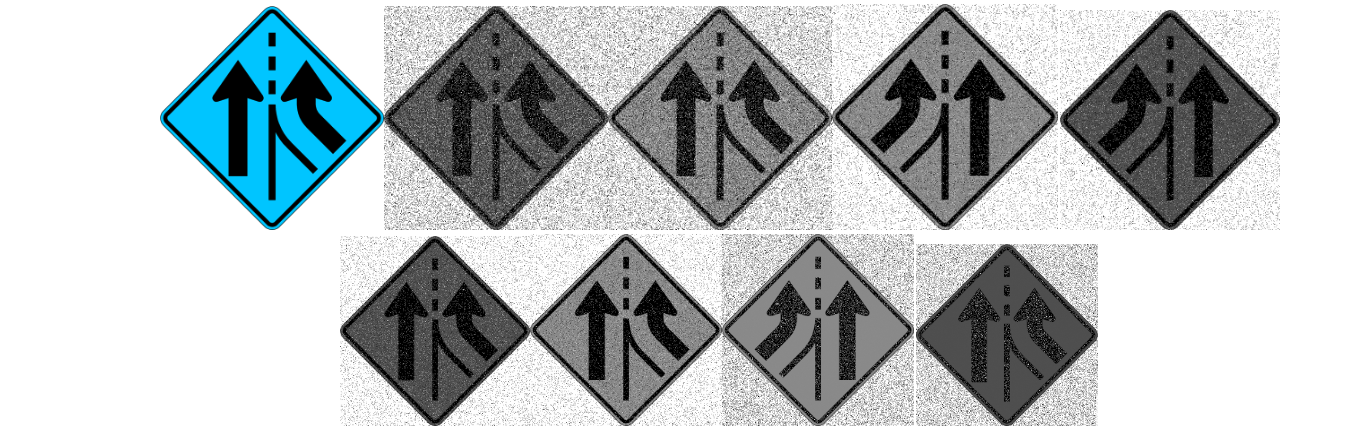


Fig. Images generated using CV

* 1. **YOLO**

YOLO, or "You Only Look Once," is a real-time object detection algorithm that has gained significant popularity due to its speed and accuracy. Developed by Joseph Redmon and Ali Farhadi, YOLO is a deep learning-based algorithm that takes a single pass through the input image and predicts the presence of objects and their corresponding bounding boxes simultaneously.

The YOLO algorithm works by dividing the input image into a grid (e.g., 13x13, 19x19, or 52x52 cells), and each grid cell is responsible for predicting a fixed number of bounding boxes. These bounding boxes have predefined aspect ratios (also called anchors) to accommodate various object shapes. Each bounding box prediction consists of:

1. The position and dimensions of the bounding box.
2. A confidence score, indicating the likelihood of an object being present in the bounding box.
3. Class probabilities for each object class that the model is trained to recognize.

To obtain the final object detection results, the YOLO algorithm applies a threshold to the confidence scores and class probabilities, keeping only the bounding boxes with high scores. Non-maximum suppression is then used to remove duplicate detections, resulting in the final set of detected objects and their respective bounding boxes.

The YOLO algorithm has undergone several iterations, with each version improving upon the previous one. YOLOv2, YOLOv3, YOLOv4, and YOLOv5 are some of the well-known versions. These newer versions introduce architectural improvements, better feature extraction, and other optimizations that result in better accuracy and speed.

YOLO's single-pass approach and end-to-end architecture enable it to perform object detection much faster than other object detection methods, such as R-CNN and its variants, making YOLO more suitable for real-time applications.

**Architecture:**

1. **Backbone:** The backbone in YOLOv5 uses CSPNet (Cross Stage Partial Network) architecture, which enables efficient feature extraction and better gradient flow during training. It leverages CSPDarknet53, a modified version of Darknet53, as its backbone.
2. **Neck:** The neck in YOLOv5 uses PANet (Path Aggregation Network) and BiFPN (Bidirectional Feature Pyramid Network) architectures for aggregating and processing features from different layers in the backbone. This helps the model in detecting objects of varying scales and sizes effectively. The neck also utilizes SPP (Spatial Pyramid Pooling) layers to aggregate context information from different spatial scales.
3. **Head:** The head of YOLOv5 consists of multiple output layers corresponding to different anchor box scales. Each output layer predicts bounding box coordinates, objectness scores, and class probabilities for a fixed number of anchor boxes. The head employs BCE (Binary Cross Entropy) loss for class probabilities and CIoU (Complete Intersection over Union) loss for bounding box regression.

**Code:**

**Model definition:** YOLOv5 uses a modular design with a focus on maintainability and scalability. The model architecture is defined using a YAML configuration file, which describes the structure of the backbone, neck, and head, as well as the number of classes and anchors. This approach allows for easy customization of the model architecture.

**Dataset and data augmentation:** YOLOv5 supports various dataset formats and includes a data augmentation pipeline. The data augmentation techniques used in YOLOv5 include random resizing, flipping, translation, rotation, and color jittering. These augmentations help the model generalize better to new data.

**Training loop:** The training loop in YOLOv5 code iterates over the training dataset for a predefined number of epochs. In each iteration, the model processes a batch of images and calculates the losses for the objectness, class probabilities, and bounding box coordinates. The gradients are then backpropagated through the network, and the model weights are updated using an optimizer, such as SGD or Adam.

**Learning rate scheduling:** YOLOv5 employs a custom learning rate scheduler called "OneCycle" policy, which starts with a low learning rate, increases it to a maximum value, and then decreases it again. This approach helps in achieving faster convergence and better performance.

**Model evaluation:** During training, YOLOv5 is periodically evaluated on a validation dataset. The evaluation metrics include mean average precision (mAP) and other class-wise metrics. TensorBoard is used to visualize the training progress and metrics.

**Model export:** YOLOv5 supports exporting the trained model in different formats, such as TorchScript and ONNX, for deployment on various platforms.

**Inference:** For inference, YOLOv5 processes an input image, generates output predictions, and applies non-maximum suppression (NMS) to filter out overlapping detections. The remaining high-confidence detections are then returned as the final output.

**Model Optimization and Evaluation:** To ensure the effectiveness of the road sign detection and recognition system, the models were optimized and evaluated using various techniques. Hyperparameter tuning was conducted to optimize the model's performance, adjusting parameters such as learning rate, batch size, and weight decay. Different model architectures, including YOLOv3, YOLOv5, and YOLOv7, were experimented with to identify the most suitable architecture for the task. Model evaluation was carried out using metrics like precision, recall, F1-score, and mean Average Precision (mAP), along with the analysis of confusion matrices.

1. **Implementation**

**4.1 Google Colab and Google Drive Integration:**

The real-time road sign detection, recognition, and driver guidance system was implemented using the powerful cloud-based Jupyter Notebook environment provided by Google Colab Pro. The integration of Google Drive allowed for seamless access and management of the project's datasets while taking advantage of the computational resources offered by Google Colab Pro, such as GPUs for accelerated model training.

**4.2 Individual Model Development:**

Both the detection and classification models were built and tested individually to ensure that they performed well in their respective tasks. The detection model, based on YOLOv5, was trained on the LISA dataset using transfer learning, data augmentation, and appropriate preprocessing techniques. The classification model, implemented using the ResNet architecture, was trained on the custom dataset of 47 road sign types, which had been augmented and preprocessed to meet the model's input requirements.

**4.3 Model Training Pipeline and Ensemble Solution:**

A training pipeline was designed for the two-layer ensemble solution to streamline the process. Once the detection model identified a road sign, the cropped image was passed to the classification model through the pipeline. The classification model then accurately classified the detected sign. This approach combined the strengths of both models and enabled them to work in tandem to achieve the desired system performance.

**4.4 Hyperparameter Tuning and Model Selection:**

The models were fine-tuned using various hyperparameters, such as learning rate, batch size, and weight decay, as well as different model architectures, including YOLOv3, YOLOv5, and YOLOv7. The YOLOv5 model was ultimately chosen for the detection layer and the ResNet model for the classification layer, based on their superior performance and accuracy. Several iterations of training and testing were performed to ensure the optimal hyperparameters were selected.

**4.5 Model Evaluation and Performance Analysis:**

The trained models were thoroughly evaluated using a combination of performance metrics, including precision, recall, F1-score, and mean Average Precision (mAP). Confusion matrix analysis was used to identify areas for improvement and refine the models accordingly. The individual models were evaluated separately, and their performance was further assessed when working in tandem within the ensemble solution. This comprehensive evaluation ensured that the models were capable of handling real-world scenarios effectively.

**4.6 Modifying Loss Function:**

To improve the performance of the detection model and better address the specific requirements of the road sign recognition task, the loss function was modified to cater to object detection, bounding box regression, and class prediction. The SignboardLoss class was implemented as a custom loss function, which incorporated modifications.

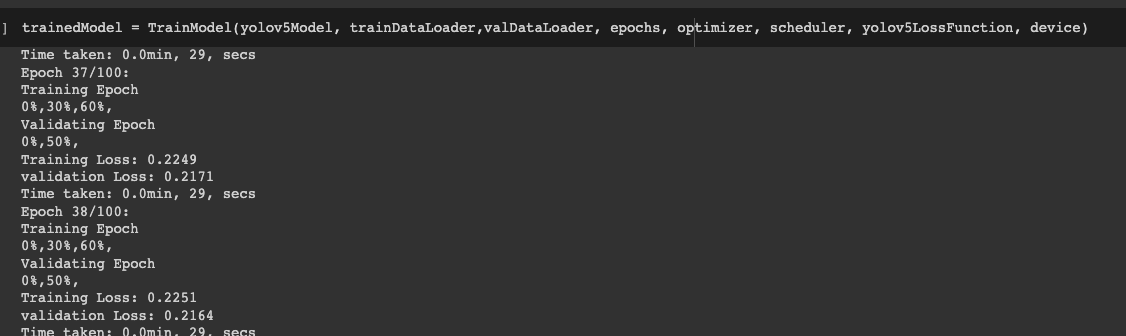
****

Fig. Training model

**5. Results**

The results obtained from our real-time road sign detection, recognition, and driver guidance system project are summarized below:

Layer 1 - Sign Detection: After experimenting with YOLOv3, YOLOv5, and YOLOv7, we chose YOLOv5 for the first layer due to its superior performance. During the training process, the learning rates and loss values demonstrated that the model was converging and improving.

Layer 2 - Sign Classification: In the second layer, we started with basic neural networks using a sigmoid activation function. However, after testing various algorithms, we determined that the ResNet architecture provided better classification results for road sign images. As with the first layer, the learning rates and loss values showed that our model was effectively learning.

The combination of the YOLOv5 model for road sign detection and the ResNet model for road sign classification resulted in a system that effectively recognized and identified road signs in real-time, ultimately improving driver guidance and road safety.

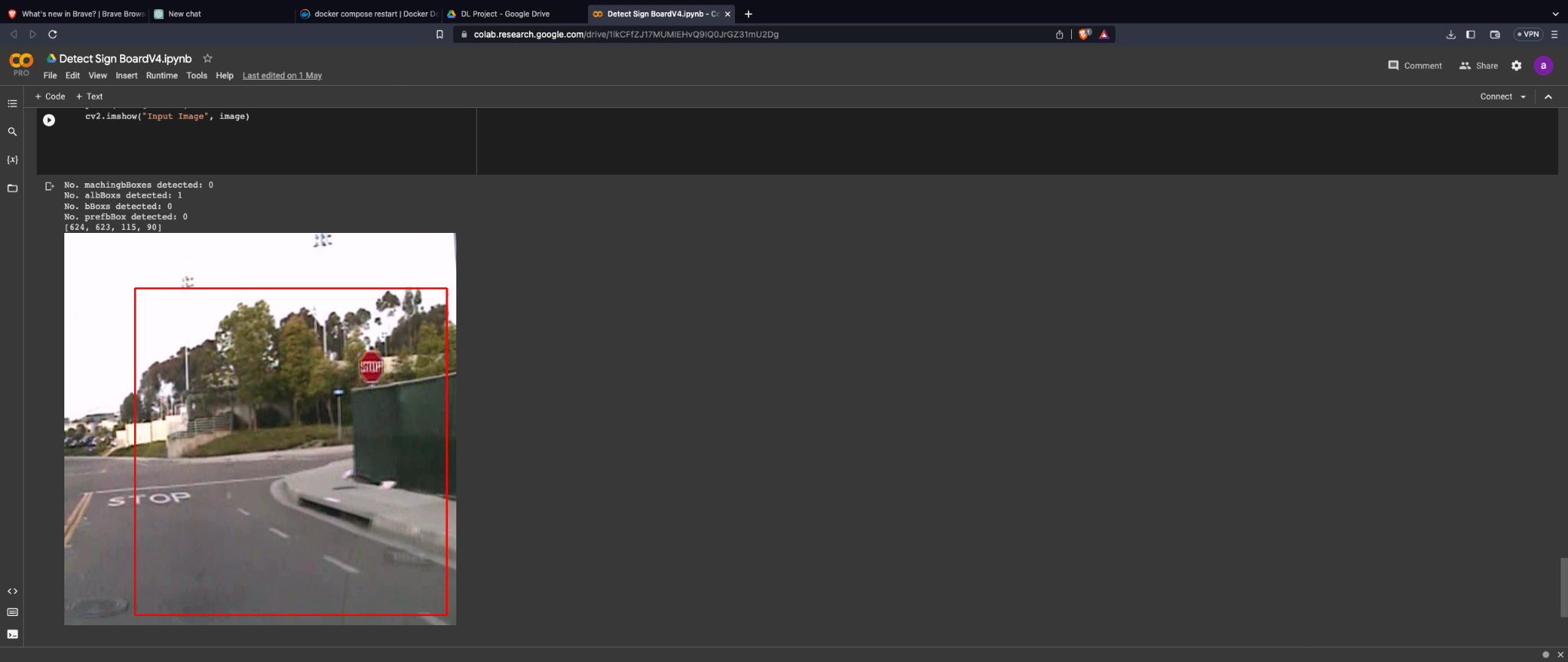
****

Fig. Initial trained version

Despite being unable to measure the exact accuracy of our models, the learning rates and loss values suggest that they are effectively learning the road sign detection and classification tasks. There is significant potential for future work in refining these models, improving their accuracy, and ultimately developing a driver guidance system based on the road sign detection and recognition capabilities.

In conclusion, our project has laid the groundwork for a comprehensive real-time road sign detection, recognition, and driver guidance system. Although we faced time constraints that limited our ability to measure the accuracy of our models, the preliminary results show promise for future development. With further research and improvements, our project could contribute significantly to enhancing driver safety and assistance on the road.

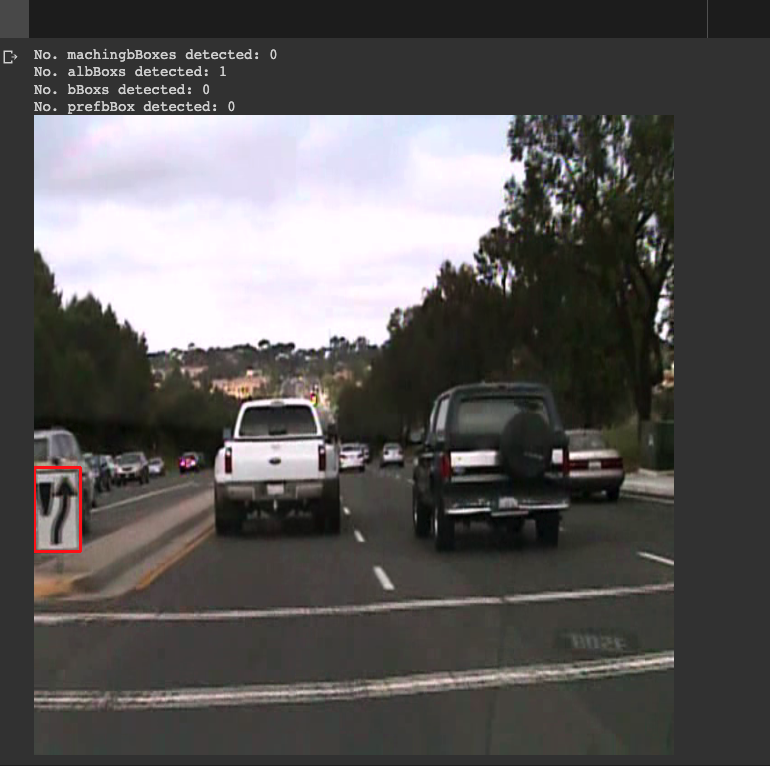
****

Fig. Final Version Results of the Trained Model

**6. Future Work**

**1.** **Real-time driver guidance system**: As the current project focuses primarily on road sign detection and classification, a natural extension of the project would be to implement a real-time driver guidance system. This system would alert drivers of upcoming road signs, providing context-aware safety warnings and speed limit notifications based on the detected road signs.

**2.** **Enhanced model performance**: Continuous improvements can be made to the model's performance by exploring new architectures, incorporating additional pre-processing techniques, or employing advanced optimization methods. Techniques such as model ensembling or transfer learning from other object detection models could potentially enhance the model's detection and classification abilities.

**3.** **Expand the range of road signs:** The project currently focuses on a limited set of 47 road signs. Future work could involve expanding the range of road signs recognized by the system, as well as incorporating variations in road sign designs from different countries or regions.

**4.** **Integration with other transportation systems:** The road sign detection and recognition system could be integrated with navigation applications, vehicle telematics, or intelligent transportation systems. This would enable a more comprehensive and context-aware driving experience, improving overall road safety and efficiency.

**7. Conclusion**

The Real-time Road Sign Detection, Recognition, and Driver Guidance System project successfully developed a two-layered model for detecting and classifying road signs using an ensemble approach. The first layer was responsible for road sign detection, while the second layer focused on classification among 47 different sign types. A combination of YOLOv5 and ResNet architectures was employed to achieve accurate and efficient detection and classification.

Data collection and preparation involved using the LISA dataset and additional images obtained through web scraping and image augmentation techniques. A custom loss function was developed to optimize object detection, bounding box regression, and class prediction, resulting in a more specialized model tailored to the unique challenges posed by road sign recognition.

The project demonstrated promising results and provides a strong foundation for future work in real-time driver guidance systems, enhanced model performance, and integration with other transportation systems. By continually refining the model and expanding its capabilities, the project has the potential to contribute significantly to road safety and improve the driving experience.

**References**:

1. <https://github.com/ultralytics/yolov5>
2. LISA Traffic Sign Dataset. (n.d.). Retrieved from<http://cvrr.ucsd.edu/LISA/lisa-traffic-sign-dataset.html>