Traffic Sign Localization and Classification for Autonomous Vehicles

Jennifer Dong, Sharvil Limaye, Vian Miranda, Dhruv Rai, John Tawfik
{jennifer.dong, sharvil.limaye, vian.miranda, dhruv.rai, john.tawfik}@rutgers.edu

Abstract

Traffic sign detection is essential for autonomous vehicles and driver-assistance systems to ensure road safety and reduce human error. This project introduces a two-stage model: a localization model using Faster R-CNN with a Vision Transformer (ViT) backbone and a custom Convolutional Neural Network (CNN) for classification. The models were trained on Mapillary and Kaggle datasets, achieving a validation Intersection-over-Union (IoU) of 0.27 for localization and an accuracy of 81.3% for classification. The system demonstrates a scalable approach to traffic sign detection under diverse conditions.

1 Introduction

Traffic signs guide drivers and regulate traffic flow, but factors like poor visibility and human error contribute to missed signs, causing accidents. In 2022, 4.2% of drivers involved in fatal crashes failed to obey traffic signs, highlighting the need for automated systems.

Similar to the capabilities of YOLO [1], this project focuses on developing a robust model designed to perform both localization and classification. The model specifically targets traffic signs, processing dashcam footage captured under a wide range of environmental and lighting conditions. The system comprises a two-stage architecture: a localization model and a classification model. The localization model detects the presence of traffic signs within each frame of the video, applies a bounding box around the detected signs, and crops the bounded regions for further analysis. These cropped regions are then passed to the classification model, which identifies the type of traffic sign, such as stop signs, traffic lights, or speed limits.

The classification result is then annotated onto the original video frame. This process is applied sequentially to every frame in the input video. Finally, the labeled frames are stitched together to reconstruct the video, offering a fully annotated output that highlights traffic signs in real time. By addressing challenges such as low visibility and varying dataset quality, the system demonstrates a scalable approach for enhancing traffic awareness and safety in autonomous driving applications.

2 Methodology

2.1 Datasets

Two datasets were utilized:

- **Mapillary Traffic Sign Dataset** [2]: Contains global traffic scenes under diverse lighting and weather conditions. It includes bounding box annotations for sign localization.
- **Kaggle Traffic Sign Detection Dataset** [3]: Provides cropped traffic sign images across 15 classes, used for classification.
  - **Classes:** 'Green Light', 'Red Light', 'Speed Limit 10', 'Speed Limit 100', 'Speed Limit 110', 'Speed Limit 120', 'Speed Limit 20', 'Speed Limit 30', 'Speed Limit 40', 'Speed Limit 50', 'Speed Limit 60', 'Speed Limit 70', 'Speed Limit 80', 'Speed Limit 90', 'Stop'

2.2 Localization Model

The localization model uses Faster R-CNN with a ViT (DINO) backbone [4]. The ViT divides images into patches, extracts features using multi-head self-attention, and integrates with Faster R-CNN components like Region Proposal Network (RPN) and ROI pooling for precise detection.

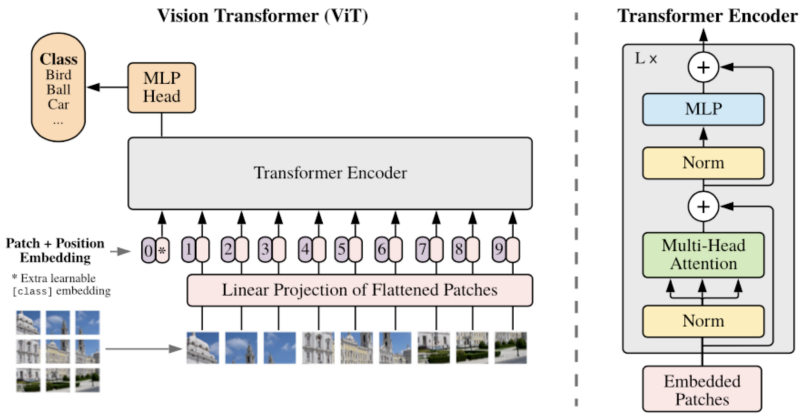

Figure 1: ViT Architecture [5]

The Vision Transformer (ViT) is the localization model's backbone. It divides the input image into patches, flattens them, and combines them with positional embeddings. These patches are then passed through a Transformer Encoder, consisting of multi-head self-attention layers and feed-forward networks. The Transformer learns the relationships between patches which results in the final output being processed by a Multi-Layer Perceptron (MLP) head to classify or, in our case, detect traffic signs. ViT's ability to capture global dependencies makes it highly effective for visual tasks like object detection.

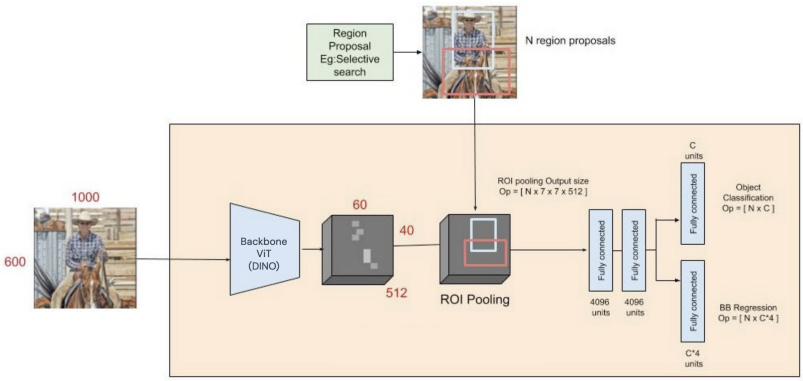

Figure 2: Localization Model Architecture (Faster R-CNN with ViT) [6]

The figure above shows the integration of the Vision Transformer within the Faster R-CNN architecture. The image first passes through the ViT backbone to extract feature maps. These feature maps are fed into a Region Proposal Network (RPN), which identifies potential object regions using anchors. The ROI Pooling layer processes these proposals, resizing them into fixed-sized feature maps. Finally, these are passed to fully connected layers for classification and bounding box regression. The combination of ViT for feature extraction and Faster R-CNN for localization enables precise detection, even in complex visual scenarios.

2.3 Classification Model

The classification model is a custom Convolutional Neural Network (CNN) designed to classify cropped RGB images of traffic signs into 15 distinct categories. As illustrated in Figure 3, the architecture begins with an input layer processing 224x224 images with three color channels (RGB).

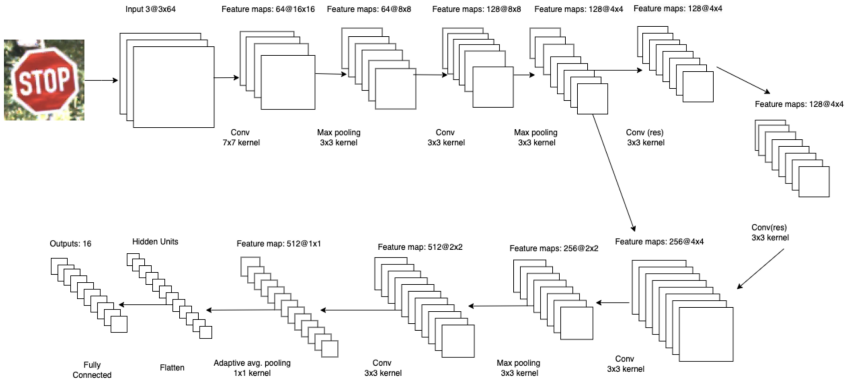

Figure 3: Classification Model Architecture

The architecture features four convolutional layers with increasing depth (64 to 512 filters), interleaved with max-pooling layers to reduce spatial dimensions, and a residual block that improves feature extraction by maintaining gradient flow and learning identity mappings. These components enable the model to capture both low-level features like edges and high-level semantic patterns critical for traffic sign recognition. The final convolutional block feeds into an adaptive average pooling layer, followed by a fully connected layer that outputs class probabilities using softmax regression.

To enhance robustness and prevent overfitting, the model employs batch normalization to stabilize training and data augmentation techniques, such as slight random rotations and color adjustments, to simulate real-world variations in traffic signs. Training was conducted using the Adam optimizer with a learning rate of 10^{-3}, cross-entropy loss, and early stopping to maximize performance on the validation set.

3 Results

3.1 Localization

Validation IoU is a measure of how well the predicted bounding boxes align with the given ground truth bounding boxes of each image.

The localization model achieved a training loss of 0.26 and a validation IoU of 0.27. It demonstrated effective bounding box prediction, as shown in Figure 4. The validation IoU results were low despite performing well, which may be due to a difficult dataset.

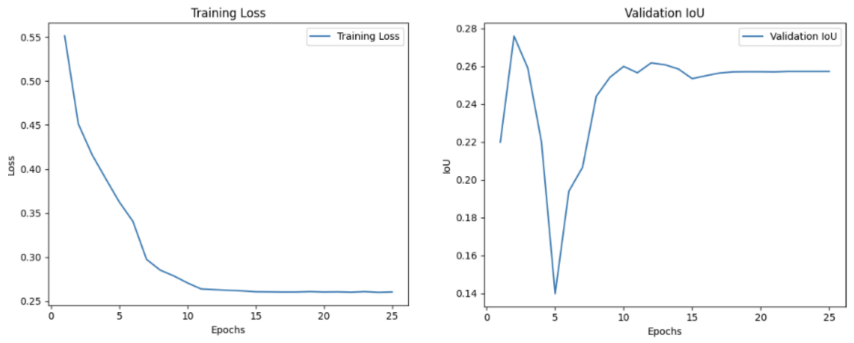

Figure 4: Localization Results: Predicted vs Ground Truth

3.2 Classification

The classification model achieved the following metrics:

- **Accuracy** 81.30%
- **Precision** 82.75%
- **Recall** 78.02%
- **F1 Score:** 78.14%

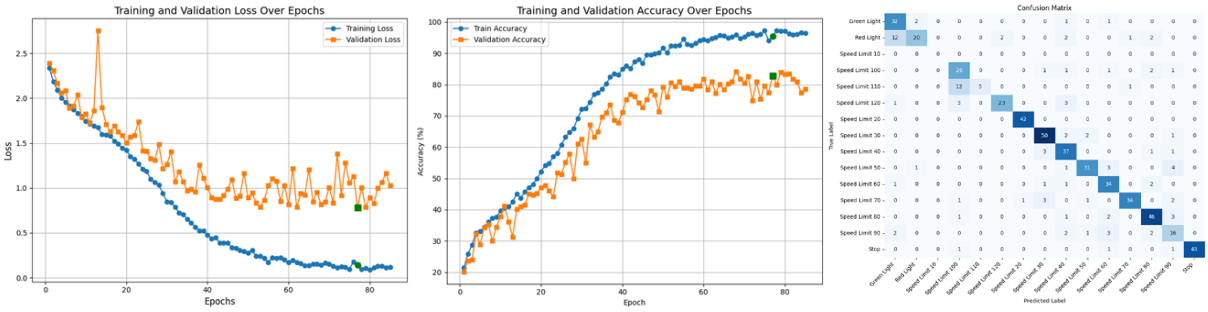

Figure 5: Classification Results

3.3 Final Results

Figure 6: Screenshot from Final Result Demo [7]

4 Challenges

- **Dataset Quality:** Many datasets contained poorly annotated or unlabelled data, which required extensive cleaning and manual verification to ensure correct labels. This not only increased the preprocessing time but also impacted the model's ability to generalize to new data.
- **Training Complexity:** The resource constraints, especially with limited computational power, made training large models difficult and time-consuming. This required us to optimize hyperparameters and utilize techniques like early stopping to prevent overfitting while ensuring model convergence.

5 Conclusion

This project demonstrates a scalable traffic sign detection system, addressing real-world challenges like diverse lighting and occlusions. Future work will focus on improving dataset quality and incorporating temporal data from video sequences for more robust detection across dynamic environments.

References

1. Ultralytics, YOLOv5 documentation: https://github.com/ultralytics/yolov5.
2. Mapillary Traffic Sign Dataset: https://www.mapillary.com/dataset/trafficsign.
3. Kaggle Traffic Sign Detection Dataset: https://www.kaggle.com/datasets/pkdarabi/cardetection/data.
4. ShirAmir. "Shiramir/Dino-VIT-Features: Official Implementation for the Paper 'Deep Vit Features as Dense Visual Descriptors'." GitHub, github.com/ShirAmir/dino-vit-features. Accessed 18 Nov. 2024.
5. Google-Research. "Google-Research/Vision_transformer." GitHub, https://github.com/google-research/vision_transformer. Accessed 10 Dec. 2024.
6. Suardinata, I Wayan, and Vivien Arief Wardhany. "Object detection in online proctoring through two camera using faster-RCNN." Jurnal Jartel Jurnal Jaringan Telekomunikasi, vol. 13, no. 2, 11 Apr. 2023, pp. 120–127, https://doi.org/10.33795/jartel.v13i2.738.
7. Aaron Ryan, YouTube, www.youtube.com/watch?v=TCtIK2KsTWQ. Accessed 18 Nov. 2024.