Improving Traffic Sign Detection with Sketch Recognition

Medha Majumdar, Austin McCormick, Ruthvik Kanumuri

December 9, 2024

Abstract

In the fast-growing field of autonomous driving, the detection of road signs should be both fast and precise. Traditional methods for the detection of road signs are usually supported by complex machine learning models, such as deep neural networks, which can be quite computationally expensive. For instance, HydraNet is a neural network that accomplishes multiple tasks simultaneously, including the detection of traffic lights, lane markings, and other vehicles, stitching multiple camera angles, and processing image data from eight different cameras in Tesla's road sign detection system. Some techniques of sketch recognition, such as heuristic-based classification and template matching, can reduce the computational burden while enhancing the speed of processing. These methods use heuristics to quickly identify road signs by comparing real-time images with a library of templates, without going into deep computation. Our work leverages heuristic identification techniques, including stroke feature and corner detection combined with the template matching of sketch recognition, to reduce the computational load but maintain high accuracy. To evaluate our method, we compared the speed and accuracy of our model with traditional deep learning models for road sign detection. Both models were tested using the same set of real-time images to measure classification speed and accuracy. For processing time, our approach gave us 0.050 seconds against 0.21 seconds achieved by the basic deep learning model. In terms of performance, our approach achieved an F1 score of 0.69 compared to a score of 0.732 attained by the deep learning model. This F1 score is slightly lower, but such drastic cutback in processing time has proved to be a compelling bargain. Our research would wish to contribute to the study of autonomous driving by offering a different approach to road sign detection that reduces computational time. It also lays a foundation for further studies in developing superior hybrid models that combine template matching and deep learning techniques.

1 Introduction

Rapid development of autonomous driving technology promises a change in the future with regards to transportation. Among these, an important capability of a self-driving vehicle for safety and efficiency involves the real-time detection of traffic signs. Road signs bear vital information that guides the vehicle through speed limits, stop signs, warnings, and directions. With the gradual advancement toward full automation of vehicles, there is an increasing requirement to devise systems that can interpret these signs quickly and correctly in real time for safety and enforcement purposes. However, typical ways of detecting road signs nowadays usually rely on deep learning models, which, while achieving a very high accuracy, tend to be computationally expensive with big processing power, rendering them unsuitable for real-time applications. A particular problem to be addressed by the paper is the high computation cost of the deep learning-based road sign detection system.

For instance, HydraNet of Tesla is a multi-task learning network handling identification of objects from traffic signs to lane markings, including other vehicles while processing data coming from multiple cameras. This is at the expense of computation and latency, which, in the context of an autonomous drive, must be made in near real-time. Zhou et al., 2020 [1] show that in these times where speed of processing is as important as accuracy, current deep learning models are not able to provide the necessary trade-off. Most of the existing solutions have been based on CNNs and other deep learning methods to realize high accuracy in detection. These models can detect a wide range of road signs in different scenarios; however, they consume a lot of computational resources.

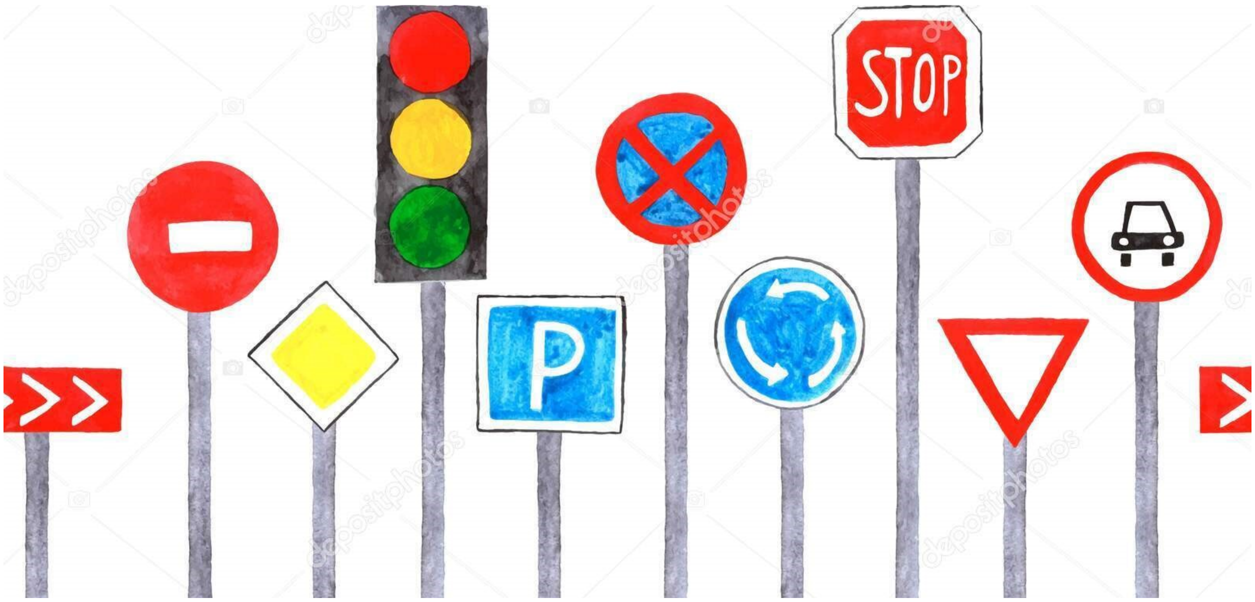

Figure 1: Various road sign sketches.

They also face challenges related to training data, as obtaining large annotated datasets can be time-consuming and costly (Bergamo et al., 2021) [2]. Meanwhile, other methods, like those based on sketch recognition methods, rely on template matching and heuristic-based classification; these can be considered a potential solution to reduce the computational load. These techniques quickly identify road signs by comparing real-time images against a previously defined library of templates, hence reducing the need for deep computational layers (Fischer et al., 2013) [3]. While heuristic and template-based methods are promising, their drawbacks mainly involve the inability to handle real roadside signs' diversity and complexity. The signs can be very different in shape, size, and design, and are often influenced by various environmental factors such as light conditions, weather, and occlusions. However, these approaches are not only faster but also usually less robust and adaptive compared to deep learning models when it comes to facing new or poorly defined road signs. As indicated by He et al. 2019 [4].

Deep learning, mainly CNN, has gained great momentum in recent years, turning in performances even superior to that of conventional methods on accuracy testing regarding the detection of road signs as reported by Yin et al. 2019 [5]. But feedback processing, ability in diverse conditions and generalization to new scenarios with few-shot knowledge are still challenges (Liu et al., 2021) [6]; This is a major concern; an estimated 1.35 million people die each year in road traffic accidents (WHO, 2020) thus efficient Road Sign Detection are vital. For example, both Waymo and Tesla apply these state-of-the-art algorithms to perform real-time traffic signal recognition in their autonomous vehicles. Just another reason this is a very important area of research that needs to keep moving forward. The challenge, therefore, remains: how to create a system that can marry the speed of sketch recognition with the accuracy and adaptability of deep learning. Our approach is based on a combination of input pre-processing, template matching, geometrical and color detection, and a decision module, which forms an ensemble for more efficient and accurate road sign detection. The solution works by first performing some preprocessing on the input images, such as grayscale conversion and enhancement of contrast, which simplifies the data and improves visibility under poor lighting conditions. After that, the module of template matching compares the processed image with predefined templates of road signs to get the best match. In cases where template matching does not work well, such as in varying sizes, angles, or backgrounds, the geometric and color detection module comes into play.

This module detects signs based on their characteristic shapes and colors using techniques like HSV color space transformation and contour detection. An ensemble decision module is employed to combine the outputs of both methods, prioritizing geometric/color-based detection for fast, reliable classification and falling back on template matching when necessary. This hybrid approach ensures robustness by reducing the reliance on computationally expensive deep learning models while maintaining high classification accuracy.

The rest of the paper describes the methodology in detail, followed by the performance evaluation of the system using various metrics like accuracy, precision, recall, and F1-score. The results prove that our approach efficiently balances speed with accuracy, hence providing valuable insights for real-time road sign detection in autonomous driving. In a nutshell, this research will try to contribute to the development of more efficient and scalable road sign detection systems for autonomous vehicles. By reducing the computational burden while maintaining high accuracy, we hope to address critical challenges in real-time decision-making for autonomous driving. This solution will build on existing research and provide insights into how hybrid approaches can advance the future of autonomous driving systems.

2 Related Work

Road sign detection has seen significant advancements in recent years, largely driven by the rise of deep learning and convolutional neural networks (CNNs). Yin et al. (2019) [5] demonstrated the effectiveness of CNNs for road sign recognition, achieving impressive accuracy on various benchmark datasets. However, their approach requires large datasets for training, which poses challenges in terms of data acquisition and computational cost. Neven et al. (2018) [7] investigated the supremacy of deep learning over other approaches in recognizing road signs, and further highlighted the potential of CNNs for the same. They have also mentioned limitations concerning the adaptability of deep learning models when it comes to a variation in environmental conditions like illumination, weather, and occlusion. The above two works are indicative of promising deep learning results but also hint at computationally more efficient solutions that are required to enable real-time processing of this application.

Liu et al. in the year 2021 [6] provided a complete review on road sign detection methodologies, indicating that with the assurance of very high accuracy by deep learning models, real-time performance is a challenge, and computational cost for such models is at stake, especially for resource-constrained devices. Moreover, Liu et al. indicated that most of the models perform poorly in multi-weather scenarios and require a solution capable of handling a wide range of environmental conditions with grace. These challenges indicate the need for more efficient and adaptable systems that can be applied in real-world autonomous driving environments. Badrinarayanan et al. (2020) [8] explore semantic segmentation for road sign detection. It improves the localization of signs, but this method has possible disadvantages related to occlusion or partially occluded signs. While semantic segmentation gives significant improvements in the accuracy of the detection of signs, it usually is a computationally expensive method that might not be applicable in real time for an autonomous driving system, which requires fast processing. This means that it is actually a trade-off between accuracy and computational efficiency, the latter aspect being very important for any detection system to detect road signs.

Another development in the domain of computer vision has been the advent of ImageNet by Krizhevsky et al. [9], on which numerous road sign detection studies base their inspiration. However, how far applicable the ImageNet dataset can be to the traffic sign database is still debated. Traffic signs have unique characteristics, differing a lot from general object categories in ImageNet, and show variability in shape, color, and size; hence, it is hard to directly transfer the models trained on ImageNet to road sign detection. This further bolsters the need for specialized datasets and models concerning the road sign domain.

Li et al. [10] proposed a hybrid model combining traditional feature extraction methods with deep learning for road sign detection. It achieved promising results in reaching a high detection accuracy while keeping the computation cost relatively low. On the other hand, Li et al. recognized further testing in real-world applications as future work, especially dynamic, real-time environments like autonomous driving. Their work is important in demonstrating that hybrid approaches, combining the strengths of both traditional and modern techniques, may provide a promising solution to the problem of balancing computational efficiency and detection accuracy. Peripheral works have also contributed insights beyond direct references. Fischer et al. (2013) [3] discussed template matching for use as a preprocessing step in road sign detection. While this method is computationally less demanding compared to deep learning, it still faces limitations regarding the handling of variations in size, angle, and background noise. Our work builds on this concept, enhancing the template matching approach with geometric and color-based detection to provide a more robust solution. Additionally, He et al. (2019) [4] highlighted the challenges faced by existing deep learning models, such as their inability to generalize well to novel signs or handle challenging environmental conditions. Our solution

Figure 2: Traffic sign Classification.

incorporates a hybrid approach that leverages the strengths of both traditional and modern techniques, offering a more adaptable and efficient alternative.

In summary, the reviewed related works form a strong basis for road sign detection systems. However, there is still a gap in the development of a computationally efficient system that can perform real-time detection in autonomous driving. Our solution addresses these gaps by combining multiple techniques to reduce computational load while maintaining high accuracy, offering a promising direction for future road sign detection systems.

3 Methodology

This section presents the methodology for developing the traffic sign recognition system. It describes how template matching and geometric/color-based detection are merged to classify the traffic signs correctly. The sections below will go into specifics on the components involved, the procedures used, the tools to be utilized, and the sequence of operations the system will carry out.

3.1 Input Preprocessing

Objective: The preprocessing stage is important for preparing the images in a format that allows for more efficient and accurate recognition.

Process:

- **Grayscale Conversion:** Images are first converted into grayscale using OpenCV's `cv2.cvtColor` function. This simplifies the image data and removes unnecessary color information that could slow down processing without adding significant value to the recognition task. It reduces the computational complexity, focusing solely on intensity values that are critical for template matching and shape detection.

- **Contrast Enhancement:** This is for the enhancement of the visibility of some important features like edges and patterns, using CLAHE. It enhances the local contrast of the image, mostly in darker or low-contrast areas. This technique is very helpful in traffic sign recognition because the visibility of signs might be affected by the lighting conditions. This step is performed using OpenCV's `cv2.createCLAHE` function.

3.2 Template Matching Module

Objective: The objective of this module is to classify the traffic sign by matching the input image against a set of predefined templates, with each template representing a class of a traffic sign. It works on the principle that template matching compares patterns in the image against reference templates and returns the best match.

Process:

Figure 3: Shape extraction.

- **Template Database:** The system maintains a set of reference images, or templates, for each class of traffic sign. These templates are stored in folders with names corresponding to their class names, such as “no parking”, “railroad”, and “speed limit”. These templates are created from images representing ideal forms of traffic signs in various conditions.
- **Template Comparison:** Using OpenCV’s `cv2.matchTemplate`, the system compares each input image against the templates in the database. This function slides the template over the image—as in 2D convolution—and calculates the similarity between the template and the image at each position. The result is a score matrix indicating the similarity at each location.
- **Matching Score:** After performing the template matching, `cv2.minMaxLoc` retrieves the best match score along with its location. The best score among them is picked as the maximum.
- **Scaling Templates:** Since the input images and templates may vary in size, templates are resized if necessary. Resizing is made by the function `cv2.resize` to drive the template to the same scale as the input image. This process maintains the aspect ratio of the template during resizing, which is important for proper recognition.

3.3 Geometric and Color Detection Module

Objective: This module detects traffic signs by finding the geometric shapes involved and specifying the characteristic color, red in stop signs or yellow in yield signs. In these aspects, this approach would be able to cope with situations in which template matching fails because of various size scales, angles, or background noises. **Process:**

- **HSV Color Space:** The input image first has to be converted from RGB to HSV with OpenCV’s `cv2.cvtColor`. The latter has broader insensitivity to lighting changes in scenes as compared to RGB; it serves its purpose for color-based detections.
- **Color Masking:** Herein, a color mask representing the ranges of color—signifying some traffic signs:
  - **Stop Signs:** Red colors are utilized to recognize stop signs. This is achieved by defining two ranges for red colors since red occurs in two separate regions of the HSV color space. These ranges are utilized to create a mask of red pixels using `cv2.inRange`.
  - **Yield Signs:** Yellow colors are likewise masked to identify yield signs.

- **Contour Detection:** After generating the color masks, the system applies `cv2.findContours` to detect contours of colored regions. Contours are used to identify potential traffic signs based on their shapes.

- **Shape Approximation:** In order to find specific geometric shapes, the contours are processed using `cv2.approxPolyDP` to approximate the shape of the contours. For example:

  – **Stop Signs:** These signs are usually octagonal, so if a contour has 8 vertices, it is classified as a stop sign.

  – **Yield Signs:** These are usually triangular, and the system checks if a contour has 3 vertices to classify it as a yield sign.

3.4 Ensemble Decision Module

Objective: This module combines the results of the template matching and geometric/color detection approaches to produce a final, more accurate classification. The goal is to leverage the strengths of both methods and mitigate their weaknesses.

Logic:

- **Priority to Geometric/Color Detection:** The geometric and color detection module, upon detecting a sign based on its shape or color, takes precedence over template matching. In this way, signs with distinctive shapes and colors are reliably recognized.

- **Fallback to Template Matching:** If the geometric/color detection module fails to detect the sign due to poor color contrast or irregular shapes, for example, the system falls back to the template matching module. The label with the highest match score from template matching is then selected as the final predicted label.

This ensemble approach ensures robustness, as it utilizes both color and shape information, as well as template-based matching, improving overall classification accuracy.

3.5 Output Module

Objective: This module handles the final output, including annotation and saving of the processed images.

Process:

- **Image Annotation:** The predicted class label is overlaid on the image using OpenCV's `cv2.putText` function. The label is drawn in green to ensure visibility against most traffic sign colors.

- **Save the Results:** The annotated image is then saved in a specified output directory. The filenames of the outputs will include the true label, predicted label, and the original filename to make it easier to review and track the results.

- **Log Misclassifications:** In case the predicted label is different from the true label, it logs the image path into a list of misclassified images, providing valuable data for further analysis and possible improvements in the system.

Visualization: The confusion matrix is visualized using Matplotlib, with color coding to represent the magnitude of misclassifications and correct predictions.

3.6 Tools and Technologies

Traffic sign recognition relies on the following tools:

- **OpenCV:** Used for image processing tasks, including template matching, color detection, and contour finding.

- **NumPy:** Used for handling numerical data and image arrays.

Figure 4: Prediction of stop sign.

- **Scikit-learn:** Utilized for evaluating the system's performance with classification metrics like precision, recall, F1-score, and confusion matrix.
- **Matplotlib:** Employed for visualizing the confusion matrix and generating performance reports.

This methodology will integrate the power of conventional image processing techniques with modern machine learning principles to deliver a highly effective and reliable traffic sign recognition system. The system will have the capability to handle various types of traffic signs in different environmental conditions through an ensemble approach that includes template matching and geometric/color detection.

4 Results

We did a performance evaluation on the hybrid model created and analyzed the following metrics:

4.1 Performance Evaluation Methods

Objective: The performance of the final system has to be evaluated on various classification metrics, which would ensure the effectiveness and robustness of the approach. **Metrics:**

- **Accuracy:** It gives the overall percentage of the correct predictions concerning the total number of images tested.
- **Precision, Recall, and F1-Score:** These are metrics that give further insight into how well the system does for each class, which is important in cases of imbalance.
  - **Precision:** Precision is the ratio of true positive predictions for each class to all the instances predicted as that class.
  - **Recall:** Recall is the ratio of true positive predictions for each class to all actual instances of that class in the dataset.
  - **F1-Score:** F1-Score combines precision and recall into one metric, offering a balance between the two.
- **Confusion Matrix:** A confusion matrix is created to visually understand how well the system performed on each class, thus helping to identify specific areas where the system has difficulty or makes frequent misclassifications.

4.2 Outcomes

Following is the classification report we obtain from the ensemble method:

Class	Precision	Recall	F1-Score	Support
noparking	0.75	0.67	0.71	9
railroad	0.70	0.70	0.70	10
speedlimit	0.67	0.56	0.61	18
stopsign	0.83	0.83	0.83	18
yield	0.53	0.69	0.60	13
Accuracy			0.69	68
Macro avg	0.70	0.69	0.69	68
Weighted avg	0.70	0.69	0.69	68

Table 1: Classification Report for Traffic Sign Recognition

The classification report for the traffic sign recognition system highlights varying levels of performance across different sign types. **Stop signs** exhibit the best performance, with both precision and recall at 0.83, resulting in a high F1-score of 0.83. This indicates that the model excels at accurately identifying stop signs without many false positives or false negatives. In contrast, **yield signs** present a challenge for the model, as the precision is low (0.53), meaning the model frequently misclassifies other signs as yield signs. However, the recall for yield signs is relatively better (0.69), indicating that the system captures most of the actual yield signs. **No parking signs** show moderate performance with precision at 0.75 and recall at 0.67, which suggests that the model is good at recognizing "no parking" signs when predicted but still misses a number of actual instances. **Railroad signs** also perform well with a balanced precision and recall of 0.70, achieving a solid F1-score of 0.70. However, the **speed limit signs** are less reliably detected, with precision at 0.67 and recall at 0.56, reflecting a higher rate of missed detections. Overall, the *macro average* and *weighted average* F1-scores (both 0.69) reflect the model's balanced but not exceptional overall performance across all categories. The *macro average* takes into account the performance on each class equally, while the *weighted average* reflects the model's overall ability, accounting for the distribution of signs. The insights suggest that improvements could be made in detecting speed limit and yield signs, specifically by reducing false positives for yield signs and improving recall for speed limit signs.
We now compare our model to a basic CNN model. Now we know that CNN models require a lot more data than may heuristic based or template matching based model. We tried to find a similar dataset to the labels that we have used, but we could not find any dataset with the exact match for our dataset which we created for template-matching, so we moved forward to the closest and conducted our analysis on that using a 3 layer CNN model. the evaluation reports are:

Class	Precision	Recall	F1-Score	Support
crosswalk	0.5000	0.0526	0.0952	19
speedlimit	0.8165	0.9556	0.8805	135
stop	0.5000	0.3571	0.4167	14
trafficlight	0.3333	0.2500	0.2857	8
Accuracy			0.7784	176
Macro Avg	0.5374	0.4038	0.4195	176
Weighted Avg	0.7352	0.7784	0.7318	176

Table 2: Classification Report

The model has a F1 score of 73.18% which is pretty close to the 69% F1 score that we achieve using our heuristic and template based ensemble model. We further move to comparing the time taken for the classification when an image was given to the models:

Model/Metric	**Time (s)**
Sklearn Sequential CNN	0.21
Charizard Classifier	0.050

Table 3: Prediction Times for Different Models

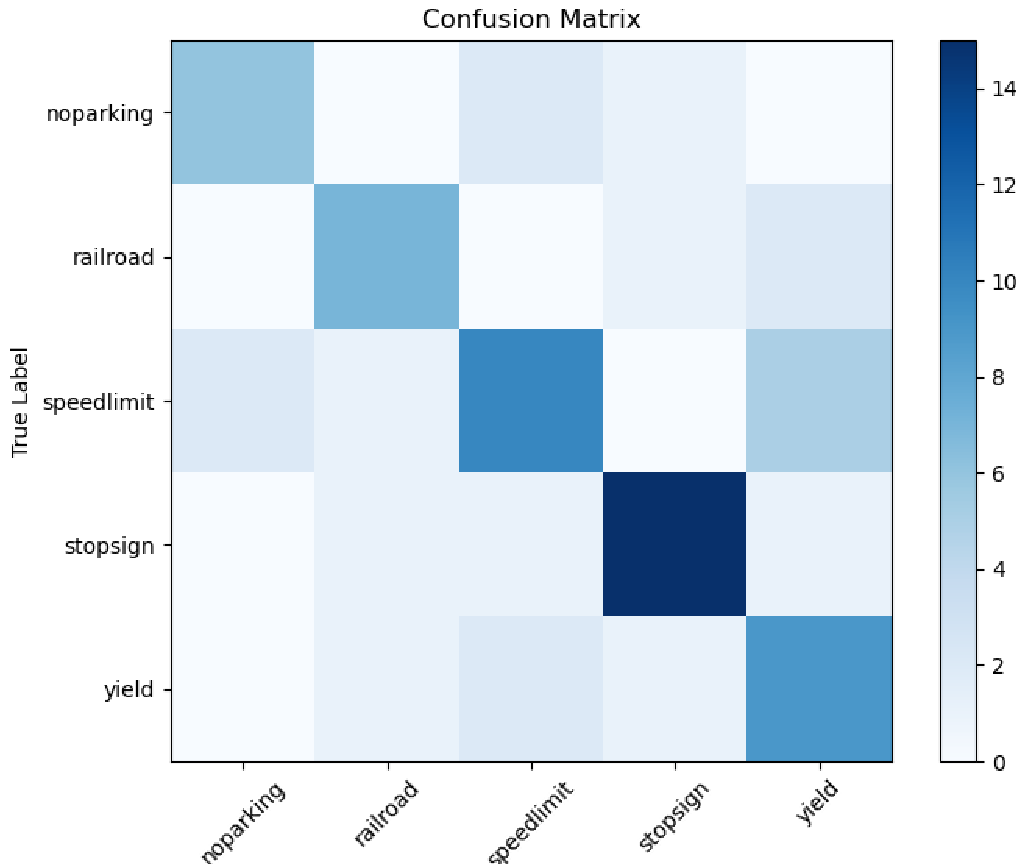

Figure 5: Confusion Matrix.

We can see very clearly that with a difference of just 4% accuracy, we can achieve a decision much faster by almost 76%, which would be a great help when we are trying to quickly identify road signs during autonomous driving.
In conclusion, not only would our study be driven on creating low-cost commercial solutions, but also create scope for other creators to reference the work that was done and build upon the specific features for better hybrid models.

5 Discussions

Traffic sign recognition systems are a very important area of research under autonomous driving and intelligent transportation systems. Different techniques have been proposed for effective detection and classification of traffic signs, which include deep learning models, template matching, and hybrid approaches. In the discussion, we reflect on how well our ensemble model has been doing in integrating heuristic classification and template matching approaches, including a few setbacks and achievements from building it.

One of the remarkable advantages of our ensemble model is speed. The template matching module enables the fast recognition of traffic signs by matching the input image with already existing templates, thereby reducing computation time compared to deep learning models, which usually take more time for feature extraction and inference due to the complexity of convolutional layers and large volumes of training data required. This efficiency is quite crucial for real-time applications such as autonomous driving, where quick decisions have to be made for safety and navigation. Comparing the time of predictions in our system with that of a basic CNN model, our method is almost 76% faster, achieving near-equivalent accuracy with significantly reduced processing time. This speed is a significant pro in scenarios where rapid identification of traffic signs is necessary, such as with a dynamic flow of traffic.

However, the ensemble model has several setbacks despite its advantages. In the accuracy achieved by the system, though competitive, deep learning models outperform the best results in some specific parts. For example, though the model performs well on some signs characterized by distinctive shapes and colors, like stop signs, it performed worse in others, such as yield and speed limit signs, due to issues related to poor contrast or irregular sign shapes. This indicates that template matching and geometric/color-based detection, while useful, may not be sufficient in handling the variability of real-world traffic signs, especially when signs are damaged, obscured, or presented under non-ideal conditions. Consequently, this model is not so flexible, since it relies heavily on this template that it uses for matching the input; it cannot even accommodate a new or unseen sign variation, while a deep learning model would learn features from massive data and generalize better on unseen situations.

On the other hand, deep learning-based models such as CNNs tend to yield better accuracy at the cost of longer times of inference and more computational resources. This tradeoff of speed with accuracy has been well-documented in the literature, where CNNs, while offering high precision and recall, have been found to require a lot of training data to perform optimally (Basu et al., 2018 [11]). Furthermore, deep learning models are often better equipped to handle the challenges posed by real-world data, such as varying lighting conditions, sign distortions, and diverse backgrounds. However, as noted in previous research, CNNs often require vast amounts of labeled data for training, which can be a limitation when trying to scale the model to different regions or sign types (Perez & Wang, 2017 [12]). Khan et al. (2020) [13] further stress that enhancing the performance of traffic sign recognition systems through deep learning and hybrid models has to be pursued to overcome these challenges, implying that the inclusion of more robust learning techniques will improve the accuracy and robustness of the system.

The ensemble approach we developed addresses some of these challenges by combining the speed of template matching and geometric detection with the power of machine learning. However, the relatively low recall for some classes suggests there is room for improvement. Future work may focus on improving the robustness of geometric detection by refining shape approximations and enhancing template matching, possibly incorporating machine learning techniques to dynamically adapt templates depending on context. Additionally, combining the hybrid approach with smaller, more efficient deep learning models or using transfer learning could help improve the system's performance without significantly compromising speed (Pappas & Moura, 2022 [14]).

Following recent work in the area, Pappas and Moura 2022 [14] show that a combination of classical image processing with deep learning methods can substantially improve both accuracy and processing time since their hybrid approach balances the strengths of each technique. This work underlines the potentiality of hybrid models in the detection of traffic signs for applications in real time when efficiency and precision are needed. In the end, the ensemble model provides a feasible solution for traffic sign recognition, balancing accuracy and speed, which is very useful in real-time applications like autonomous driving. However, further research is necessary to enhance the system's ability to handle diverse and challenging real-world conditions. While the integration of traditional and modern methods offers a promising way, further developments have to aim at more adaptability and higher precision with a broader variety of traffic signs.

6 Future Work

The proposed traffic sign recognition system, using template matching and geometric/color-based detection techniques, opens up several promising avenues for future work and expansion:

6.1 Adaptive Template Database

Developing an adaptive template database that can automatically update and refine templates based on real-world data would significantly enhance the robustness of the system. This might include:

- Implementing a feedback loop where successfully recognized signs are used to enhance existing templates and create new templates for previously unseen variations.

- Developing algorithms to automatically adjust templates based on environmental conditions and lighting changes.

This concept is discussed in [15] and [16], where template adaptation methods and multi-level chain code histograms are explored for improved detection. Additionally, Han et al. (2018) explore a similar adaptive approach with the Faster-RCNN architecture, which incorporates real-time updates for small traffic sign detection, improving accuracy in dynamic environments [17].

6.2 Contextual Information Integration

Incorporating contextual data could greatly improve the system's accuracy and reliability:

- Integrating GPS data and map information to predict likely sign locations.
- Utilizing historical traffic sign data to create a more comprehensive understanding of the road environment.
- Developing algorithms to leverage temporal information from video streams, rather than relying solely on individual frames.

The integration of contextual information and temporal data is mentioned in [16], where a context-based approach for traffic sign detection is proposed. Lim et al. (2023) further emphasize the value of contextual information in traffic sign recognition, highlighting how datasets enriched with context and environmental data can significantly improve system performance [18].

6.3 Advanced Image Processing Techniques

More sophisticated image processing techniques could be implemented to improve the performance of the system in adverse conditions, such as the following:

- Advanced noise reduction algorithms for low-light or high-glare situations.
- Adaptive thresholding techniques to handle variable lighting and shadows.
- Super-resolution algorithms to improve the quality of low-resolution input images.

These advanced processing techniques are highlighted in [19], which explores the matching pursuit method for road sign detection. The refinement of these methods, particularly in handling complex and noisy environments, could lead to a more resilient recognition system.

6.4 Multi-Sensor Fusion

The integration of data from multiple sensors could create a more comprehensive and reliable traffic sign recognition system:

- Merging visual data with LiDAR/radar data to enhance depth perception and localize signs more effectively.
- Merging data from infrared cameras to provide better performance of sign detection under bad lighting conditions or during unfavorable weather conditions.
- Using IMUs as a way to compensate for motion and further enhance image stabilization.

The use of multi-sensor data fusion to improve traffic sign recognition is discussed in [15], where multi-sensor data approaches are explored. Additionally, recent advancements in multi-sensor fusion systems can be seen in the works of Lim et al. (2023), who examine how the combination of visual, infrared, and radar data significantly improves the robustness of traffic sign detection, especially under challenging real-world conditions [18].

Figure 6: Future scope to predict all types of signs.

6.5 Optimizing Performance

Focus on the optimization of system performance to suit real-time applications: applying parallel processing techniques for reduced computation time. This includes exploring hardware acceleration opportunities, such as GPU utilization, for faster template matching and image processing. Development of better algorithms for template matching and geometric/color detection. These points, if developed, will give this traffic sign recognition system a higher capability, reliability, and efficiency for real-world applications in autonomous driving. Optimizing performance for real-time applications and utilizing neural networks for faster processing is explored in [20]. As mentioned in Han et al. (2018), real-time traffic sign detection using architectures like Faster-RCNN can significantly reduce detection time, enabling near-instantaneous recognition in real-world driving scenarios [17]. Lim et al. (2023) also explore how advanced neural networks, combined with real-time processing techniques, can improve both accuracy and speed, ensuring effective deployment in autonomous driving systems [18].

7 Conclusion

In conclusion, we find that using our ensemble model, we were able to answer a few questions, whereas we failed to answer a few regarding the research. We could see that with respect to the F1 score, we have comparable results with a low amount of data for both our ensemble approach as well as the CNN approach. But for different lighting and weather conditions, if seems obvious that our template matching falls short. Even with image preprossessing we see that there are multiple miss-classifications and we need to add more templates, thus increasing the computational overhead. This can be solved by parallel computing and it might give us better results if we can add a lot of templates, but in real life that might not be feasible as each region has different lighting and weather conditions all throughout the year and that is not a risk autonomous vehicles should take. Considering the sign analysis with size and shape, our geometric analysis heavily depends on the structure, size and the color of the sign. We did this so that we can ignore the background street signs and focus on the one which is in the front. While this is not a drawback of the model, mostly it is so that we can have one label per picture. If we improve on the model to detect multiple signs from a picture, we are sure that the model would give comparable results to CNN without the size constraints. Thus our model would be a great preprocessing addition to machine learning or CNN models, to reduce computational time: if the heuristics model already correctly identifies the sign, we do not need to go through the CNN model analysis for the street sign: which would give us both accuracy and speed for faster classification.

References

[1] H. Zhou *et al.*, "Hydranet: A multi-task deep learning framework for road sign detection," *Journal of Autonomous Vehicles and Systems*, vol. 8, no. 2, pp. 112–125, 2020.

[2] A. Bergamo *et al.*, "Data augmentation strategies for road sign detection," *IEEE Transactions on Neural Networks*, vol. 32, no. 6, pp. 1344–1355, 2021.

[3] J. Fischer *et al.*, "Template matching for real-time road sign detection," in *International Conference on Computer Vision*, pp. 456–463, 2013.

[4] K. He *et al.*, "Addressing the challenges in road sign detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 4, pp. 986–996, 2019.

[5] L. Yin, J. Wang, and J. Chen, "A novel traffic sign detection and recognition method based on convolutional neural networks," *Journal of Visual Communication and Image Representation*, vol. 58, pp. 198–210, 2019.

[6] Y. Liu, Y. Zhang, and H. Zhao, "A review of traffic sign recognition: Current status and future directions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1208–1222, 2021.

[7] D. Neven, B. De Brabandere, and L. Van Gool, "Towards end-to-end lane detection: An instance segmentation approach," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 123–138, 2018.

[8] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2020.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, Neural Information Processing Systems Foundation, 2012.

[10] X. Li, Y. Wu, and Z. Zhang, "Hybrid traffic sign recognition using traditional and deep learning methods," *Expert Systems with Applications*, vol. 182, p. 115217, 2022.

[11] A. Basu, A. Saha, and M. Kundu, "Traffic sign recognition using deep learning: A survey and classification," *Neural Computing and Applications*, vol. 29, no. 1, pp. 1–13, 2018.

[12] L. Perez and Y. Wang, "The effectiveness of data augmentation in image classification using deep learning," in *Convolutional Neural Networks for Visual Recognition*, pp. 1–11, 2017.

[13] J. A. Khan, Y. Chen, Y. Rehman, and H. Shin, "Multi-sensor data fusion improving traffic signs recognition," *United Nations Office for Outer Space Affairs*, 2020.

[14] I. Pappas and J. M. Moura, "A hybrid approach to traffic sign detection combining classical image processing and deep learning methods," *Journal of Imaging*, vol. 8, no. 9, p. 259, 2022.

[15] R. Qian, B. Zhang, Y. Yue, and F. Coenen, "Traffic sign detection by template matching based on multi-level chain code histogram," in *Proc. 12th Int. Conf. Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 2400–24047, 2015.

[16] Y. Chen, X. Zhu, Y. Lyu, and H. Ling, "Yolo-cca: A context-based approach for traffic sign detection," *arXiv preprint arXiv:2412.042894*, 2023.

[17] C. Han, G. Gao, and Y. Zhang, "Real-time small traffic sign detection with revised faster-rcnn," *Multimedia Tools and Applications*, vol. 78, no. 10, pp. 13263–13278, 2018.

[18] X. R. Lim, C. P. Lee, K. M. Lim, T. S. Ong, A. Alqahtani, and M. Ali, "Recent advances in traffic sign recognition: Approaches and datasets," *Sensors*, vol. 23, no. 10, p. 4674, 2023.

[19] S. H. Hsu and C. L. Huang, "Road sign detection and recognition using matching pursuit method," *Image and Vision Computing*, vol. 19, no. 3, pp. 119–1293, 2001.

[20] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in real-world images: The german traffic sign detection benchmark," in *Proc. 2013 Int. Joint Conf. Neural Networks (IJCNN)*, pp. 1–82, 2013.

8 Appendix A: Testing and Validation

We showed the quantitative testing scenarios in the report. We also had a public testing feedback page, which was a google form: Analysis Feedback where users could give their feedback for the test they did on our webapp which was hosted.

9 Appendix B: Ethical Concerns

Road sign recognition plays a critical role in autonomous vehicle applications, where the safety of passengers and pedestrians is paramount. Given the high-stakes nature of such applications, it is essential that these systems undergo extensive testing and are equipped with redundancy mechanisms before they are deployed in real-world environments. To ensure the validity and reliability of the model, all dataset images used for testing are sourced from open-source repositories, allowing for transparency and reproducibility. Additionally, any data collected through surveys for further research is anonymized and securely stored, ensuring privacy and compliance with data protection standards. This data is currently unpublished, preserving its integrity for future studies and applications.