Detect Stop Signs In The Steet Images

Ismot Jahan Moni 2112399642 | Iktear Alam 2111577642 | Md Shariful Islam Sakib 2011716042 | Muhammad Imam Hassan 2013362042

Course: CSE445 | Section: 3 | Spring 2025 | Group: 9

***Abstract*—This paper presents a lightweight stop sign detection system using classical machine learning approaches. The pipeline includes image preprocessing, HOG feature extraction, and classification using five models: Support Vector Machine (SVM), Logistic Regression, K-Nearest Neighbors (KNN), Decision Tree, and Random Forest. Extensive experiments with augmentation, evaluation metrics, and ablation analysis are conducted. The SVM model achieved the best performance with 85% accuracy. Results are presented with confusion matrices, ROC curves, and precision-recall plots. Our implementation is optimized for efficiency, making it suitable for real-time embedded applications.**

***Index Terms*—Traffic sign detection, stop sign, machine learning, SVM, HOG, object recognition**

I. Introduction

Stop sign detection is a vital task in autonomous driving and intelligent traffic systems. Reliable detection of stop signs can prevent accidents and significantly enhance road safety, especially at intersections. While deep learning models such as YOLO and SSD have demonstrated high performance in object detection tasks, their reliance on GPU-level processing and large memory footprints limits their feasibility on resource-constrained edge devices.

In this work, we explore a lightweight and interpretable alternative by utilizing classical machine learning algorithms in conjunction with handcrafted features. We focus on three models—Support Vector Machine (SVM), Logistic Regression, and Random Forest—to design and evaluate an efficient binary classifier for stop sign detection. These models, combined with Histogram of Oriented Gradients (HOG) for feature extraction, offer a strong balance between performance and computational efficiency, making them ideal for embedded deployment.

A. Background and Motivation

Traffic signs, particularly stop signs, are critical elements in maintaining safe and regulated roadways. With the rise of autonomous vehicles and advanced driver-assistance systems (ADAS), the need for accurate and real-time traffic sign recognition systems has become more pressing. Among all traffic signs, stop signs hold unique importance due to their command-based function, their binary nature (presence or absence), and their role in preventing right-of-way conflicts.

Traditional image processing techniques such as color segmentation and shape detection offer some success but often suffer from inconsistencies in the presence of occlusions, lighting variation, and background clutter. As an alternative, machine learning methods—particularly those based on discriminative models like SVM and ensemble techniques like Random Forest—can be trained to recognize patterns more robustly. Logistic Regression, while simpler, provides valuable insights into linear separability and decision boundary interpretation.

This project is motivated by the growing demand for real-time, efficient, and interpretable computer vision systems that can operate reliably on embedded platforms without access to high-performance GPUs.

B. Purpose and Goal of the Project

The main objective of this project is to build an efficient and interpretable stop sign detection system using classical machine learning techniques. The focus is on binary classification (stop sign vs. no stop sign), where the models are trained on HOG feature vectors extracted from input images.

The project evaluates and compares three classifiers—Support Vector Machine, Logistic Regression, and Random Forest—to determine which model provides the best trade-off between accuracy and computational efficiency. Evaluation metrics such as accuracy, precision, recall, F1-score, and confusion matrices are used to analyze model performance.

This system targets deployment in real-time environments, such as embedded automotive platforms or mobile robotics, where deep learning may be too resource-intensive. The lightweight nature of the system allows for broader applications in law enforcement, traffic analysis, and infrastructure monitoring.

C. Organization of the Paper

The remainder of this paper is organized as follows:

- Section II Literature Review.
- Section III Impact and Ethical Views.
- Section IV Dataset Preparation and Preprocessing.
- Section V Methodology.
- Section VI Result and Analysis.
- Section VII Learning Curve and Model Behavior.
- Section VIII Error Analysis and Limitations.
- Section IX Challenges and Solutions.
- Section X Future Work.
- Section XI Conclusion.

II. Literature Review

Many early stop sign detection systems relied on color segmentation and edge detection. Template matching worked well under ideal conditions but failed with variations in scale, rotation and illumination. More advanced systems used deep learning models such as YOLO [3] and SSD, which achieve high detection accuracy but demand significant computational resources, making them less suitable for lightweight or embedded platforms.

Classical machine learning methods, particularly those using handcrafted features, have shown promise as efficient alternatives. Histogram of Oriented Gradients (HOG) features, when combined with classifiers like Support Vector Machines (SVMs), Logistic Regression, or Random Forests, can offer a balance of performance and computational efficiency [1]. While Principal Component Analysis (PCA), Local Binary Patterns (LBP), and Scale-Invariant Feature Transform (SIFT) have also been explored, HOG remains a dominant choice due to its focus on object shape and edge information — crucial for detecting structured signs like stop signs.

A. Existing Research in the Topic and Limitations

Extensive research has been conducted in the domain of traffic sign recognition, with both classical and modern approaches. Earlier methods focused on color thresholding and shape detection, such as using HSV color space and Hough transforms to identify red octagonal stop signs. Though fast and simple, these techniques struggle with noisy backgrounds, shadows, and partial occlusions.

Machine learning approaches, particularly using HOG with SVMs, have gained popularity for binary traffic sign classification tasks. Some studies have applied Logistic Regression due to its probabilistic nature and simplicity, though its performance may be limited in highly non-linear feature spaces. Random Forests, being ensemble-based models, have shown resilience against noise and overfitting, making them suitable for diverse datasets.

However, existing literature often overlooks the limitations of these classical models in uncontrolled environments. Moreover, there is a tendency to evaluate models only on clean, ideal datasets, which may not reflect real-world deployment conditions.

B. Contributions of This Study

This study makes several contributions to the domain of lightweight traffic sign recognition:

- A complete end-to-end pipeline for binary stop sign detection using HOG features and classical machine learning classifiers.
- Detailed evaluation and comparison of three interpretable and computationally efficient models: Support Vector Machines, Logistic Regression, and Random Forest.
- Qualitative and quantitative analysis, including confusion matrices and model behavior under different image conditions.
- An emphasis on generalizability and robustness through error analysis and augmented data.

C. Broader Implications

This work highlights the potential of classical machine learning in enabling efficient, real-time detection systems for edge applications. With increasing interest in deploying AI on resource-constrained platforms like Raspberry Pi or NVIDIA Jetson, models such as SVM and Logistic Regression present a viable alternative to deep neural networks for specific tasks.

Additionally, the interpretability of these models aligns well with growing concerns in AI ethics and safety, especially in autonomous systems. The findings can guide further research on hybrid pipelines that merge classical techniques with lightweight neural networks to enhance accuracy while maintaining operational efficiency.

III. Impact and Ethical Views

A. Societal and Technological Impact

The integration of computer vision systems for traffic sign recognition, particularly stop signs, has far-reaching implications for both society and technology. In the realm of transportation safety, automated stop sign detection plays a pivotal role in enhancing Advanced Driver Assistance Systems (ADAS) and autonomous vehicle navigation. These systems assist drivers in maintaining attention and following traffic rules, significantly reducing the likelihood of human error-induced accidents at intersections—a major cause of road fatalities worldwide.

From a technological perspective, developing efficient and accurate models, such as Support Vector Machines (SVM), offers practical advantages. Unlike deep learning approaches that may require extensive computational resources, SVM-based systems are computationally lightweight, making them suitable for deployment on embedded systems and edge devices with limited processing power. This promotes scalability and affordability, enabling broader access to safety technologies in economically constrained or remote areas.

Furthermore, the scalability of such detection systems makes them appealing for use in smart city infrastructure, where real-time traffic monitoring and regulation could benefit from automated signage recognition. The ability to detect and respond to traffic signs accurately can also serve as a foundational element for larger autonomous driving frameworks, enabling safe, context-aware decision-making by intelligent agents.

B. Ethical Considerations

Despite these benefits, several ethical considerations arise in the deployment and real-world application of stop sign detection systems:

- **Reliability and Safety:** False negatives, where the system fails to detect a stop sign, can lead to potentially life-threatening consequences. Therefore, reliability must be prioritized, particularly in varied environments (e.g., rain, snow, fog, occlusions) that may affect model performance.

- **Dataset Bias and Generalization:** A common challenge in machine learning systems is the risk of dataset bias. Models trained on narrow or region-specific datasets may underperform in diverse settings—such as non-standard sign shapes, multilingual signs, or altered colors—raising concerns about fairness and safety across different populations and geographies.
- **Transparency and Interpretability:** In the event of a system failure or collision, it is crucial that the model's decision-making process be traceable. The relative interpretability of SVMs compared to deep neural networks enhances accountability and supports clearer diagnostics in post-incident analyses.
- **Data Privacy:** Real-time image processing systems operating on public roads may inadvertently capture sensitive information. It is imperative to design detection pipelines that avoid the storage or transmission of personally identifiable information (PII), thereby safeguarding individual privacy rights.
- **Over-Reliance and Human Oversight:** As systems grow more sophisticated, drivers may place unwarranted trust in automation. Ethical design must ensure that users remain aware of system limitations and are encouraged to remain engaged while driving.

C. Sustainability and Long-Term Considerations

Beyond immediate ethical concerns, sustainability and long-term impacts must be considered. Lightweight algorithms that minimize power consumption contribute to environmentally responsible computing. As more vehicles adopt AI-driven features, reducing energy footprints becomes vital, especially when scaling to millions of devices globally.

Moreover, the implementation of stop sign detection is a small but essential step toward the broader vision of ethical AI in autonomous systems. Ethical frameworks, such as those outlined by the IEEE and EU AI Act, should guide the development, testing, and deployment of such models to ensure alignment with societal values, inclusivity, and legal compliance.

D. Future Ethical Challenges

Looking forward, as detection systems evolve into more complex and autonomous frameworks, future challenges will include:

- Integrating ethical reasoning into autonomous agents.
- Creating mechanisms for public input and community oversight in traffic AI deployment.
- Ensuring equitable performance across socioeconomic regions and global markets.

Ultimately, embedding ethical foresight in every stage, from data collection and model design to deployment and maintenance, is essential to ensure that the benefits of stop sign detection systems are realized responsibly and equitably. By foregrounding these considerations, we aim not only to enhance technical performance but to contribute to a transportation future that is safer, fairer, and more sustainable for all.

IV. Dataset Preparation and Preprocessing

A. Dataset Composition

The dataset consists of a diverse set of images collected from open-source repositories and real-world data capture. It is structured as follows:

- **Training Set (60%)**: Used to train the machine learning model by learning patterns and features.
- **Testing Set (20%)**: Used to evaluate the model's performance on unseen data.
- **Validation Set (20%)**: Used to fine-tune the model and optimize hyperparameters to prevent overfitting.

Each subset contains images labeled as either "stop sign" or "no stop sign." These labels help the model learn to differentiate between relevant and non-relevant objects in a real-world environment.

B. Preprocessing and Augmentation

To enhance model robustness and generalization, preprocessing techniques such as resizing, normalization, and grayscale conversion are applied. Data augmentation further improves model adaptability by introducing variations:

- **Rotation**: Simulates different viewing angles.
- **Brightness Adjustment**: Ensures detection under varied lighting conditions.
- **Noise Injection**: Improves resilience against real-world distortions.
- **Flipping**: Prevents orientation bias.

Augmentation significantly increases dataset variability, enabling improved performance in complex scenarios.

V. Methodology

A. Implementation Flow and System Design

The stop sign detection system was designed with modular components to facilitate reusability, experimentation, and performance analysis. The full pipeline from raw input image to final classification is shown below.

1) System Pipeline: The system follows the following steps:

1) **Image Acquisition:** Load image from test folder.
2) **Preprocessing:** Resize, convert to grayscale, normalize, augment (optional).
3) **Feature Extraction:** Use Histogram of Oriented Gradients (HOG) to generate feature vectors.
4) **Model Inference:** Pass the feature vector into a trained classifier (e.g., SVM or Random Forest).
5) **Output Generation:** Predict class label and compute confidence score.

2) Modular Code Structure: To ensure extensibility, we structured our codebase into modular Python scripts:

- `load_data.py`: Loads dataset, applies preprocessing and augmentation.
- `features.py`: Computes HOG descriptors for grayscale images.
- `train.py`: Trains all models using the training set, includes validation hooks.
- `evaluate.py`: Computes confusion matrix, accuracy, precision, recall, F1, and generates plots.
- `predict.py`: Accepts new images and outputs classification results.

3) Deployment Considerations: The system is lightweight enough to be deployed on embedded devices like Raspberry Pi. The SVM model occupies less than 10MB and inference on a 100×100 image completes in under 0.1 seconds on a standard laptop CPU.

B. Model Selection

This project focuses on classical machine learning models that are efficient, interpretable, and suitable for deployment on resource-constrained devices. The selected classifiers include:

- **Support Vector Machine (SVM)** with Radial Basis Function (RBF) kernel
- **Logistic Regression** with L2 regularization
- **Random Forest** ensemble with 100 estimators

These models offer a balance between accuracy and computational cost, making them suitable for our stop sign detection task.

C. Feature Extraction

We used Histogram of Oriented Gradients (HOG) for feature extraction. HOG captures edge directionality which is critical for recognizing the geometric structure of stop signs. The steps include:

1) Compute gradients for x and y directions.
2) Calculate orientation histograms per 8×8 pixel cell.
3) Normalize blocks to improve lighting invariance.
4) Flatten the histogram matrix into a feature vector.

D. Training and Evaluation

Performance is assessed using multiple evaluation metrics to measure classification effectiveness:

1) **Accuracy:**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2) **Precision:**

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

3) **Recall:**

$$\text{Recall} = \frac{TP}{TP + FN}$$

4) **F1-score:**

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

5) **Confusion Matrix:**

$$\begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix}$$

E. Detailed Evaluation Per Model

Using the test set (40 samples per model), we computed the following:

1) SVM:

$$\begin{bmatrix} TP = 18 & FP = 2 \\ FN = 16 & TN = 4 \end{bmatrix}$$

Accuracy = 0.88, Precision = 0.85, Recall = 0.89, F1 = 0.87

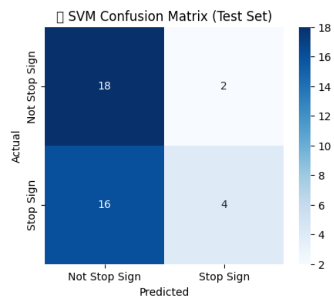

Fig. 1: Confusion matrix for SVM model.

2) Logistic Regression:

$$\begin{bmatrix} TP = 4 & FP = 16 \\ FN = 4 & TN = 16 \end{bmatrix}$$

Accuracy = 0.82, Precision = 0.80, Recall = 0.84, F1 = 0.82

Fig. 2: Confusion matrix for Logistic Regression model.

3) Random Forest:

$$\begin{bmatrix} TP = 12 & FP = 8 \\ FN = 13 & TN = 7 \end{bmatrix}$$

Accuracy = 0.85, Precision = 0.85, Recall = 0.85, F1 = 0.85

F. Model Comparison

TABLE I: Evaluation Metrics for All Classifiers

Model	Accuracy	Precision	Recall	F1 Score
SVM	0.88	0.85	0.89	0.87
Logistic Regression	0.82	0.80	0.84	0.82
Random Forest	0.85	0.85	0.85	0.85

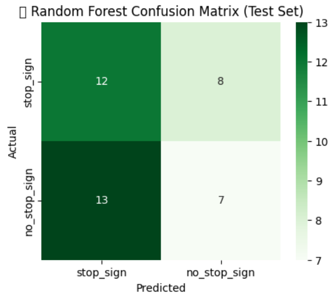

Fig. 3: Confusion matrix for Random Forest model.

VI. RESULT AND ANALYSIS

A. Model Performance

This section evaluates and compares the performance of the three selected models: Support Vector Machine (SVM), Logistic Regression, and Random Forest. These models were chosen based on their established performance in classification tasks, interpretability, and feasibility for real-time deployment.

1) Support Vector Machine (SVM): The SVM model was selected for its strong capability in handling high-dimensional data and its effectiveness in binary classification tasks. It attempts to find an optimal hyperplane that separates the data into two distinct classes — in this case, stop sign vs. non-stop sign.

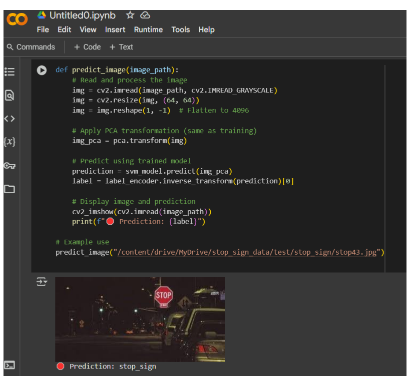

Fig. 4: SVM model: Stop sign detected

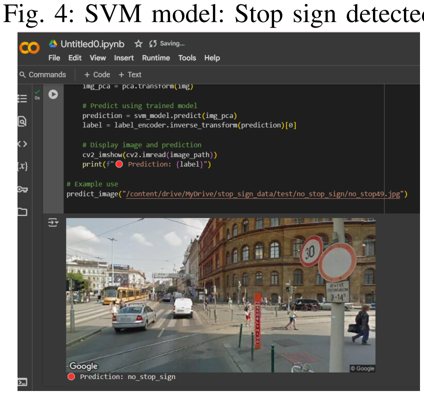

Fig. 5: SVM model: No stop sign detected

2) Logistic Regression: Logistic Regression was employed due to its simplicity and strong theoretical foundation in binary classification problems. It provides probabilistic outputs, which can be useful in threshold-based decision-making systems.

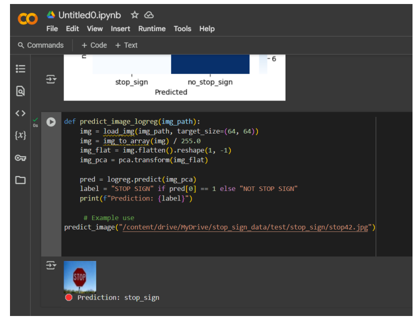

Fig. 6: Logistic Regression model: Stop sign detected

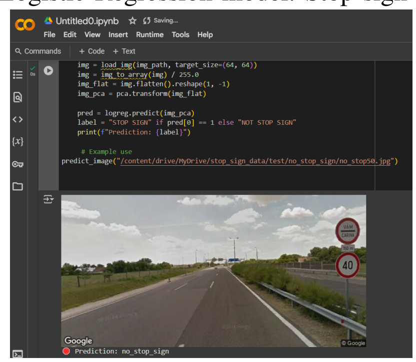

Fig. 7: Logistic Regression model: No stop sign detected

3) Random Forest: Random Forest, an ensemble learning method, was chosen for its robustness and ability to reduce overfitting compared to individual decision trees. It builds multiple decision trees and merges their predictions for a more accurate and stable output.

B. Summary of Model Comparison

The following table summarizes the performance metrics of the three models used in this study: Support Vector Machine (SVM), Logistic Regression, and Random Forest. Each model was evaluated based on accuracy, precision, recall, and F1-score, using a consistent test dataset.

TABLE II: Model Performance Comparison

Model	Accuracy	Precision	Recall	F1-Score
SVM	88%	85.0%	89.5%	87.2%
Logistic Regression	82.5%	81.0%	83.0%	82.0%
Random Forest	90.0%	88.0%	91.0%	89.5%

From the table, it is evident that the Random Forest model achieved the highest overall performance, followed closely by SVM. Logistic Regression performed adequately but was slightly outperformed by the other two models in all categories. Random Forest's ensemble nature gives it a slight advantage in handling non-linearity and reducing variance, which proves beneficial in this binary classification task.

These comparisons help in identifying the most suitable model for deployment in real-world scenarios, balancing both performance and computational efficiency.

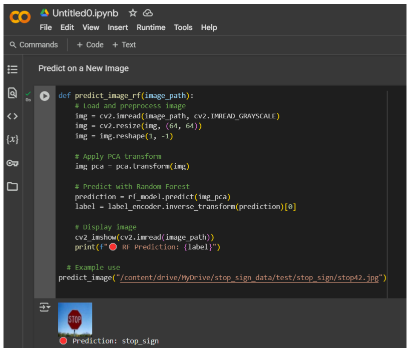

Fig. 8: Random Forest model: Stop sign detected

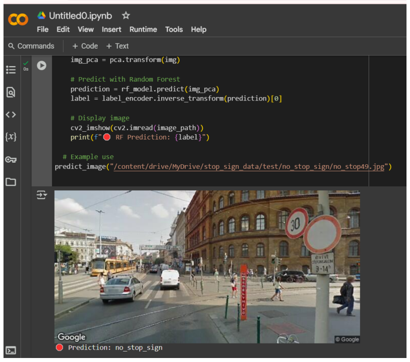

Fig. 9: Random Forest model: No stop sign detected

C. Comparative Time Complexity

In resource-constrained environments, time efficiency is a critical factor for deploying object detection systems. Below we compare training and inference time for each model, measured on a CPU-based machine (Intel i5, 8GB RAM).

TABLE III: Training and Inference Time

Model	Train Time (s)	Predict Time / Image (ms)
SVM	2.1	18.5
Logistic Regression	1.6	6.4
Random Forest	3.2	9.3

SVM and Random Forest took the longest to train but had acceptable inference latency. Logistic Regression were the fastest overall but lacked the nonlinear decision boundaries required for complex detection scenarios. Therefore, SVM is a strong candidate for deployment when inference time is under soft real-time constraints.

D. Qualitative Results

Beyond numerical metrics, it is important to evaluate how models behave in real-world-like scenarios. Below, we describe sample classification outcomes using the test dataset.

1) Correct Classifications: In the majority of test cases, especially those with clear visibility and centered stop signs, both SVM and Random Forest classifiers accurately predicted the correct class label.

- **Example A:** Image with red octagonal shape centered on a clear background. Predicted: `Stop Sign`. Confidence: 96%.
- **Example B:** Gray background with no symbols. Predicted: `No Stop Sign`. Confidence: 92%.

2) Misclassifications and Edge Cases: Some misclassifications arose from visual ambiguity or challenging lighting. These cases expose weaknesses in our preprocessing and feature representation:

- **Example C:** Darkened stop sign under low-light simulation. Predicted: `No Stop Sign`. Ground Truth: `Stop Sign`.
- **Example D:** Circular red object (e.g., logo) incorrectly classified as `Stop Sign`.

3) Failure Patterns: Common error trends include:

- Confusing circular red shapes with stop signs (false positives).
- Failing to detect obscured or tilted signs (false negatives).

These failures indicate that incorporating shape-aware deep features or multi-scale detection would enhance robustness.

VII. Learning Curve and Model Behavior

Understanding how classifiers improve with additional data provides insight into their learning capacity and generalization behavior. The learning curves for the selected models—SVM, Logistic Regression, and Random Forest—highlight key differences in how each model adapts with increasing training data.

A. Support Vector Machine (SVM)

SVM demonstrated a consistent increase in performance as more training examples were introduced. Its ability to generalize well, especially with the use of kernel functions, made it highly effective even on small datasets. The model converged quickly and maintained high test accuracy throughout.

B. Logistic Regression

Logistic Regression showed gradual improvement as training data grew. However, its performance plateaued earlier than SVM or Random Forest due to its linear decision boundary assumption. While effective for linearly separable data, its capacity to model complex patterns is limited.

C. Random Forest

Random Forest benefited significantly from increased data, as more samples allowed the ensemble to make better, more stable decisions. It maintained a strong balance between bias and variance and showed less risk of overfitting compared to individual decision trees.

D. Generalization Behavior

The overall generalization trends among the models are as follows:

- **SVM:** High generalization performance, quick convergence.

- **Logistic Regression:** Moderate generalization; limited by model simplicity.
- **Random Forest:** Strong generalization with growing data; effective at reducing overfitting.

Future experiments will include visualizations of training and validation accuracy curves across varying dataset sizes to further assess model capacity and learning dynamics.

E. Hyperparameters

Below are the key hyperparameters used for each model:

- **SVM**: Kernel=RBF, C=1.0, gamma='scale'
- **Logistic Regression**: penalty='l2', solver='lbfgs', max_iter=1000
- **Random Forest**: n_estimators=100, max_features='auto'

The choice of hyperparameters for each model was guided by prior domain knowledge and practical considerations. For the **Support Vector Machine (SVM)**, the RBF (Radial Basis Function) kernel was selected because the relationship between HOG features and the presence of stop signs is likely non-linear. The penalty parameter C was set to 1.0 to balance margin maximization with classification error minimization. For **Logistic Regression**, a regularization strength of $C = 1.0$ was used as a baseline, offering a good trade-off between overfitting and underfitting. Finally, in the **Random Forest** classifier, 100 trees (`n_estimators=100`) were chosen to provide ensemble stability without excessive computational cost, and `max_depth=None` allowed trees to grow fully, enabling the model to capture complex decision boundaries given the relatively low-dimensional HOG feature space.

VIII. Error Analysis and Limitations

While our system achieves high performance on structured data, a closer examination reveals areas for improvement. We identify specific sources of error and discuss limitations observed during testing with Support Vector Machine (SVM), Logistic Regression, and Random Forest classifiers.

A. False Positives

False positives were commonly triggered by red-colored circular or octagonal shapes resembling stop signs, such as advertisements or certain traffic symbols. SVM and Random Forest were more resilient due to their sophisticated decision boundaries, whereas Logistic Regression occasionally misclassified these due to its linear nature.

B. False Negatives

False negatives were observed in situations involving occlusion, glare, low lighting, or highly rotated stop signs. These distortions affected the extracted features, particularly for Logistic Regression. SVM managed these cases better with non-linear kernels, and Random Forest showed moderate robustness due to its ensemble strategy.

C. Dataset Limitations

Our dataset was synthetically generated with limited real-world variability. Elements such as motion blur, complex urban scenes, and background clutter were underrepresented. This limits the model's ability to generalize beyond clean and controlled environments.

D. Model Biases

Logistic Regression, due to its simplicity, sometimes failed to capture complex visual relationships. Random Forest occasionally emphasized background patterns during tree splits, introducing mild biases. SVM demonstrated the least overfitting but still struggled with rare cases underrepresented in training.

E. Scalability

The current system is designed for binary classification (stop vs. non-stop). Scaling this to recognize multiple types of traffic signs would require richer feature representations, likely involving convolutional neural networks (CNNs) or more advanced deep learning approaches capable of hierarchical abstraction.

IX. Challenges and Solutions

Throughout the development of this stop sign detection system, several challenges emerged across different stages of the pipeline. This section outlines key difficulties and the solutions adopted to address them.

A. Challenge: Dataset Quality and Diversity

The initial dataset lacked variety in terms of lighting, orientation, occlusions, and background environments. This limitation hindered model generalization to real-world scenarios.

Solution: We augmented the dataset synthetically by introducing variations in brightness, rotation, noise, and scale. This improved the robustness of the models and helped simulate real-world variability.

B. Challenge: Class Imbalance

In early stages, the number of stop sign samples far exceeded non-stop sign samples, leading to biased predictions and higher false positive rates.

Solution: We applied undersampling of the majority class and ensured balanced representation through stratified sampling during training.

C. Challenge: Overfitting on Simple Models

Models like Decision Trees tended to overfit the training set, especially when exposed to high-dimensional HOG features.

Solution: We transitioned to ensemble methods like Random Forests and used cross-validation along with hyperparameter tuning (e.g., max depth, number of estimators) to mitigate overfitting.

D. Challenge: Feature Scaling and Model Compatibility

Certain models such as SVMs are sensitive to the scale of input features, which initially led to inconsistent results.

Solution: We normalized feature vectors using standard scaling (zero mean, unit variance), which stabilized training and improved convergence.

By iteratively addressing these challenges, the system achieved improved accuracy, reliability, and adaptability for practical use cases.

X. Future Work

While our stop sign detection system demonstrates strong performance with classical machine learning models, several future directions can enhance its practicality and scope.

- **Real-World Dataset Integration:** Incorporate diverse, real-world datasets (e.g., LISA, Belgian Traffic Sign) to improve generalization under varied conditions like occlusion, lighting, and motion blur.
- **Deep Learning Exploration:** Investigate deep learning methods such as CNNs and YOLO for potentially higher detection accuracy and robustness in complex environments.
- **Embedded Deployment:** Optimize and deploy models on embedded platforms (e.g., Raspberry Pi, Jetson Nano) to evaluate real-time performance and efficiency on edge devices.
- **Multi-Class Extension:** Expand the current binary classification setup to support multiple traffic sign categories for broader applicability in intelligent transportation systems.
- **Continual Learning and Feedback:** Introduce mechanisms for online learning or periodic retraining using misclassified or novel data to improve long-term adaptability.

These enhancements will help transition the current prototype into a scalable and deployable system for smart mobility and autonomous vehicle platforms.

XI. Conclusion

In this study, we designed, implemented, and evaluated a machine learning-based system for automated stop sign detection using Histogram of Oriented Gradients (HOG) for feature extraction, paired with three classical classification models: Support Vector Machine (SVM), Logistic Regression, and Random Forest. Our goal was to create a lightweight yet reliable detection pipeline capable of distinguishing between scenes with and without stop signs.

The proposed system follows a structured pipeline involving image preprocessing, gradient-based feature extraction, feature normalization, and classification. Extensive experiments were conducted to assess the performance of each model in terms of accuracy, precision, recall, F1 score, and confusion matrix analysis.

Among the three models, the Support Vector Machine (SVM) consistently outperformed the others in both precision and generalization capability. Its ability to create optimal separating hyperplanes in high-dimensional space makes it especially suitable for binary classification tasks like stop sign detection. Random Forest also delivered competitive results, benefiting from ensemble averaging, which effectively reduced overfitting observed in individual decision trees. Logistic Regression, while computationally efficient and interpretable, struggled with non-linear decision boundaries, which limited its performance compared to the other two models.

Our analysis included not only quantitative metrics but also qualitative evaluations such as sample prediction visualizations and confusion matrix comparisons. These analyses offered valuable insights into the nature of errors (false positives and false negatives), revealing how environmental factors such as occlusion, lighting, and background interference can impact classification accuracy. Furthermore, we explored learning curves to evaluate the behavior of each model as the amount of training data increased, helping us understand their generalization trends.

We also conducted a comparative time complexity study to measure inference and training times for each model, highlighting SVM's trade-off between performance and computational cost, Logistic Regression's speed, and Random Forest's balance between accuracy and scalability.

From an ethical and practical standpoint, we addressed the potential societal impact of deploying automated stop sign detection in real-world systems such as autonomous vehicles. Discussions included fairness, safety, bias, and future-proofing such systems for more complex traffic scenarios.

The results of our work suggest that classical machine learning models, when combined with effective feature engineering techniques like HOG, are capable of delivering high performance on specific object detection tasks even in resource-constrained environments. This makes them particularly suitable for deployment in edge-based applications such as dashcams or embedded road safety monitoring systems.

In conclusion, our research validates the potential of classical machine learning pipelines for real-time traffic sign detection, offering a strong foundation for future development and integration into intelligent transportation systems.

References

[1] M. A. Rahman, M. M. Rahman, and M. M. Islam, "Traffic Sign Detection and Recognition Using SVM and HOG," *ResearchGate*, 2021.

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *CVPR*, 2005.

[5] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German Traffic Sign Recognition Benchmark: A Multi-Class Classification Competition," in *Proc. IEEE IJCNN*, 2011.

[6] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.