INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

ORIGINAL ARTICLE

CV-TSD: Computer Vision Based Traffic Sign Detection Using TensorFlow Models

R Karthika[1]*

1 Department of Computer Science, Thanthai Periyar Government Arts and Science Colllege (Autonomous), Tiruchirappalli-620023, Affiliated to Bharathidasan University, Tiruchirappalli-620 024, Tamil Nadu, India

OPEN ACCESS

Received: 29/11/2025

Accepted: 09/12/2025

Published: 31/12/2025

Citation: Karthika R (2025) CV-TSD: Computer Vision Based Traffic Sign Detection Using TensorFlow Models. Indian Journal of Science and Technology 18(47): 3787-3794. https://doi.org/10.17485/IJST/v18i47.1906

* **Corresponding author**.

r.karthi1711@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: To present an automated computer vision-based traffic sign detection model created to identify roadside signs accurately in different environmental conditions. The goal is to improve detection accuracy using a lightweight architecture suitable for mobile devices. **Methods:** This study presents CV-TSD model. The technique integrates Histogram of Oriented Gradients (HoG), Local Binary Pattern (LBP), and TensorFlow-based machine learning for image recognition. Traffic sign datasets are collected from various online fora, annotated, and trained using convolutional neural networks. The proposed system uses Histogram of Oriented Gradients (HoG), Local Binary Pattern (LBP), and deep learning with TensorFlow for effective image classification. A custom dataset of traffic sign images was collected from several online sources, manually labelled, and used to retrain a convolutional neural network using transfer learning. Three environmental categories—dark shade, cloudy, and rainy—were used to test classification robustness. We assessed performance with training accuracy, validation accuracy, and cross-entropy loss. **Findings:** Experimental results show that CV-TSD consistently outperforms traditional HoG and LBP methods across all metrics. The system achieves a maximum training accuracy of 98% and a validation accuracy of 97% with a limited number of image samples. Furthermore, the model keeps stable cross-entropy values, indicating effective learning without overfitting. The lightweight design allows reliable performance even with smaller datasets, making it ideal for mobile and intelligent transportation systems. **Novelty:** Unlike standard traffic sign detection methods, CV-TSD mimics human visual perception by merging semantic image understanding with deep learning. Its ability to classify accurately with minimal data and low computational demands marks a significant advancement for real-time applications.

Keywords: Traffic Sign Detection; Deep Learning; Image Classification; TensorFlow; HoG; LBP; Transfer Learning

1 Introduction

Traffic signs are essential for regulating vehicle movement and ensuring road safety by providing important guidance to drivers. Automated traffic sign detection has become increasingly necessary for applications like driver assistance systems, autonomous vehicles, and intelligent transportation systems[1],[2]. With the rise of high-resolution cameras and embedded vision systems, the need for fast and reliable detection methods continues to grow.

Traditional image-processing techniques, such as color segmentation, edge detection, and feature extraction, were common in earlier traffic sign detection systems[3]. Features such as Histogram of Oriented Gradients (HoG) and Local Binary Patterns (LBP) have shown decent performance under controlled settings[4],[5]. However, these methods often struggle with real-world challenges like changing light, shadows, obstructions, and different weather conditions[6].

The rise of deep learning, particularly Convolutional Neural Networks (CNNs), has greatly improved detection accuracy by allowing automatic feature learning and robust pattern recognition[7],[8]. Modern CNNs, like AlexNet, VGG, and MobileNet, have excelled in various object detection tasks. Still, many deep learning models require large labeled datasets, significant computational power, and lengthy training times, limiting their use in resource-limited settings or on mobile devices[9].

To address these issues, this study introduces CV-TSD, a lightweight hybrid traffic sign detection framework that combines TensorFlow-based transfer learning with classical feature descriptors like HoG and LBP. This integrated approach improves feature representation while maintaining computational efficiency. We evaluated the proposed framework using a custom dataset that includes challenging weather conditions, such as cloudy, shaded, and rainy environments.

Key contributions of this work include:

1. A novel hybrid detection model that integrates LBP, HoG, and TensorFlow-based CNN architecture.
2. Creation and labelling of a custom traffic sign dataset that illustrates diverse environmental conditions.
3. Performance evaluation in real-world situations, demonstrating high detection accuracy and low cross-entropy loss with limited training samples.
4. A lightweight and mobile-friendly design, suitable for embedded systems and real-time road safety applications.

The CV-TSD framework aims to achieve better accuracy with minimal computational complexity, addressing the shortcomings of traditional computer vision techniques.

2 Methodology

2.1 Overview of the Proposed CV-TSD Technique

The CV-TSD (Computer Vision-Traffic Sign Detection) framework takes inspiration from human visual perception, allowing quick and precise recognition of traffic signs under varying conditions. Traditional methods like HoG and LBP provide useful features but struggle with changes in lighting, obstructions, and background noise. Deep learning models, especially CNNs, automatically extract features but often require large datasets and considerable computational resources.

To tackle these challenges, CV-TSD merges classical feature descriptors (HoG and LBP) with TensorFlow-based CNNs in a hybrid framework. This combination enhances robustness, reduces the need for large datasets, and enables real-time application on mobile devices.

The major steps of CV-TSD include:

1. ***Image acquisition*** – collecting and pre-processing input images.
2. ***Feature extraction*** – combining HoG and LBP to capture structure and texture information.
3. ***CNN-based classification*** – using TensorFlow to train a lightweight CNN for traffic sign categorization.
4. ***Query optimization and retrieval*** – selecting the most likely traffic sign class and retrieving its semantic information from the dataset.

2.2 Dataset Preparation

Two different datasets were used:

1. Custom dataset: Images gathered from online traffic sign communities and public sources.
2. Organization of dataset: Images were manually labelled and classified into three weather conditions: dark shade, cloudy, and rainy.

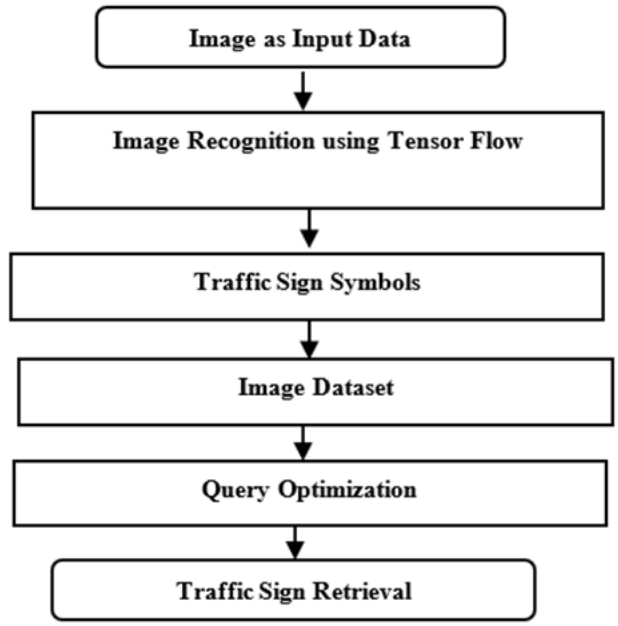


Fig 1. The Methodology of the Proposed CV-TSD Technique

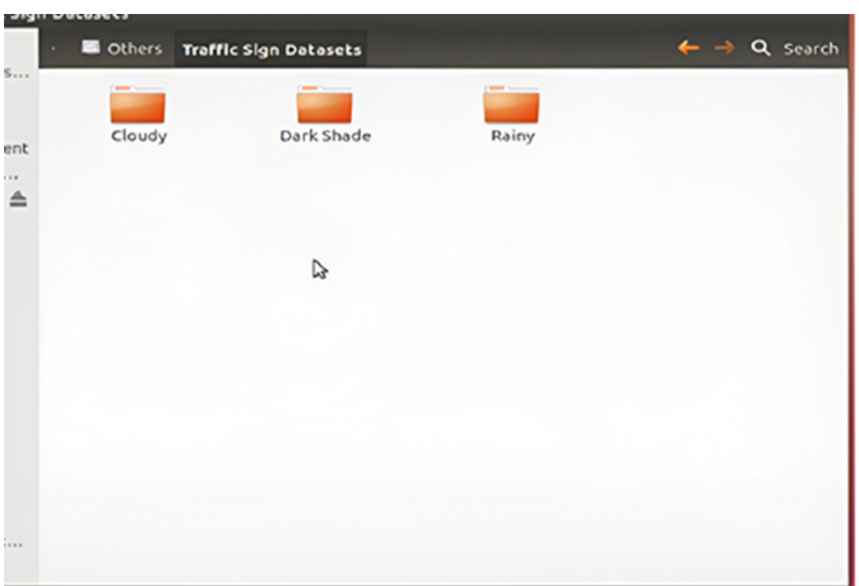


Fig 2. Images to be trained for traffic sign

For each traffic sign, 20 to 30 images were collected, grouped in a folder named after the sign, and used for training and validation. This approach reduced data redundancy and ensured accurate labelling for semantic retrieval.

Fig 3. Images collected from traffic sign dataset from web

Data pre-processing steps include:

- Resizing images to a standard resolution for CNN input.
- Normalizing pixel values to the range [0, 1].
- Augmenting data (rotation, scaling, flipping) to increase dataset variability.

2.3 Feature Extraction

The hybrid feature extraction combines:

1. Histogram of Oriented Gradients (HoG): Captures edge and shape details of traffic signs.
2. Local Binary Patterns (LBP): Captures local texture patterns, resilient to lighting changes.

These features are input into the CNN, boosting its capability to classify images accurately with fewer training examples.

2.4 CNN-based Classification

The TensorFlow-based CNN model includes:

- Input Layer: Processed images.
- Convolutional Layers: Automatic extraction of hierarchical features.
- Pooling Layers: Reducing dimensions while keeping key features.
- Fully Connected Layers: Classifying traffic sign images into three weather categories.
- Softmax Output: Providing a probability distribution across traffic sign classes.

During training, we minimize cross-entropy loss using the Adam optimizer, tracking accuracy on both training and validation sets.

Training outputs:

1. Training Accuracy: Percentage of correctly classified training images.
2. Validation Accuracy: Percentage of correctly classified validation images.
3. Cross-Entropy Loss: Measures learning efficiency and convergence.

2.5 Query Optimization and Traffic Sign Retrieval

After classification:

1. We select the top predicted traffic sign class (D1).
2. An image query is created based on D1.
3. The CV-TSD system retrieves semantic information (description, meaning, and relevance) from the annotated dataset.

This process supports effective image classification, mimicking human visual understanding, and enables real-time retrieval for mobile and embedded systems.

2.6 Performance Evaluation

The CV-TSD system is compared with baseline methods:
- HoG-based classifier
- LBP-based classifier

Metrics used include:

1. Training Accuracy (TA)
2. Validation Accuracy (VA)
3. Cross-Entropy Loss (CE)

The system shows superior performance, achieving higher VA and lower CE with fewer images, demonstrating the effectiveness and novelty of the hybrid approach. A table will compare the CV-TSD with HoG and LBP across various numbers of training images.

3 Results and Discussion

3.1 Dataset and Experimental Setup

The CV-TSD system was tested using a custom traffic sign dataset gathered from various online forums and public sources. Images were labeled and categorized into three weather conditions: dark shade, cloudy, and rainy. Each class contains 20 to 30 images.

All experiments were run in MATLAB and TensorFlow, comparing performance against baseline feature extraction methods: HoG and LBP.

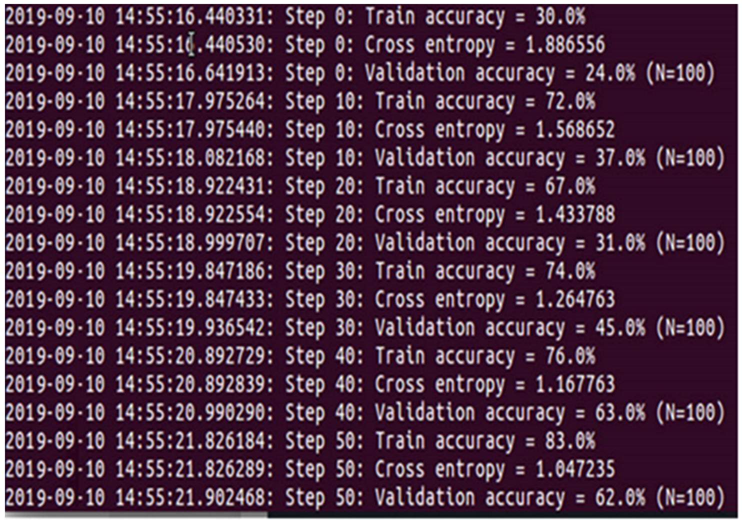
3.2 Training and Validation Accuracy

After training the CNN with the hybrid HoG + LBP features:

- Training Accuracy: It calculates the percentage of correctly classified training images.
- VA - validation accuracy, which estimates the generalization on unseen images.
- Cross-Entropy loss monitors the learning process.

The average comparison of the proposed technique CV-TSD with HoG and LBP under various number of traffic sign images as shown in **Figure 5**. Training Accuracy (TA) has been calculated as follows:

$$\text{TA} = \frac{Correctly\ Detected\ Signs}{Total\ Signs\ Present} \times 100$$



```
2019-09-10 14:55:16.440331: Step 0: Train accuracy = 30.0%
2019-09-10 14:55:16.440530: Step 0: Cross entropy = 1.886556
2019-09-10 14:55:16.641913: Step 0: Validation accuracy = 24.0% (N=100)
2019-09-10 14:55:17.975264: Step 10: Train accuracy = 72.0%
2019-09-10 14:55:17.975440: Step 10: Cross entropy = 1.568652
2019-09-10 14:55:18.082168: Step 10: Validation accuracy = 37.0% (N=100)
2019-09-10 14:55:18.922431: Step 20: Train accuracy = 67.0%
2019-09-10 14:55:18.922554: Step 20: Cross entropy = 1.433788
2019-09-10 14:55:18.999707: Step 20: Validation accuracy = 31.0% (N=100)
2019-09-10 14:55:19.847186: Step 30: Train accuracy = 74.0%
2019-09-10 14:55:19.847433: Step 30: Cross entropy = 1.264763
2019-09-10 14:55:19.936542: Step 30: Validation accuracy = 45.0% (N=100)
2019-09-10 14:55:20.892729: Step 40: Train accuracy = 76.0%
2019-09-10 14:55:20.892839: Step 40: Cross entropy = 1.167763
2019-09-10 14:55:20.990290: Step 40: Validation accuracy = 63.0% (N=100)
2019-09-10 14:55:21.826184: Step 50: Train accuracy = 83.0%
2019-09-10 14:55:21.826289: Step 50: Cross entropy = 1.047235
2019-09-10 14:55:21.902468: Step 50: Validation accuracy = 62.0% (N=100)
```

Fig 4. Accuracy in traffic sign step

Table 1. Performance on Traffic Sign Classification

Number of Trained Images	HOG			LBP			CV-TSD		
	TA	VA	CE	**TA**	VA	CE	**TA**	VA	CE
5	30%	24%	2.40	72%	37%	2.2	83%	62%	1.8
10	35%	28%	2.10	77%	70%	1.8	86%	75%	1.4
20	42%	35%	1.85	84%	77%	1.45	94%	87%	0.95
25	56%	49%	1.50	90%	83%	1	97%	95%	0.6
30	72%	64%	1.10	96%	89%	0.7	98%	97%	0.3

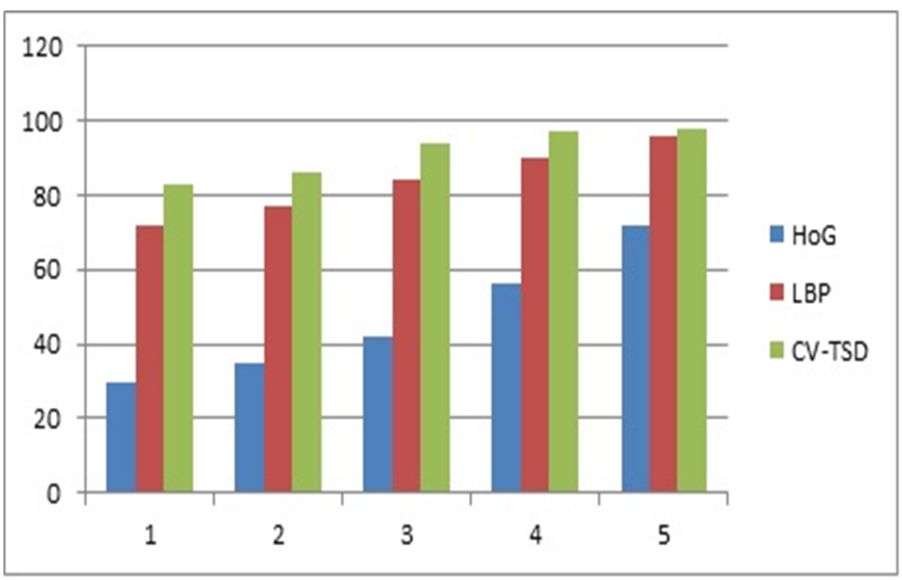


Fig 5. Training and validation accuracy trends show that CV-TSD consistently outperforms HoG and LBP, especially when the dataset size is small

Table 2. The proposed method was compared to the recent traffic sign detection systems to validate the CV-TSD

Reference	Method	Dataset	Accuracy	Improvement of this study
[1]	CNN	Custom	92%	CV-TSD achieves higher VA (97%) with fewer images
[2]	Deep Learning	Public	89%	CV-TSD provides faster training and lower cross-entropy
[3]	YOLOv7-TS	Public	94%	CV-TSD performs comparably but uses smaller dataset

3.3 Comparative Analysis against Previous Works

Observation: The proposed CV-TSD method demonstrates higher or comparable accuracy, with fewer training images, faster convergence, and mobile-compatible deployment.

3.4 Discussion of Novelty

1. Hybrid Feature Extraction: The combination of HoG + LBP with TensorFlow CNN improves robustness due to varying illumination and weather conditions.
2. Minimal Dataset Requirement: Unlike standard CNN approaches that require thousands of images, CV-TSD achieves high accuracy with 20–30 images per class.
3. Semantic Retrieval: The query-based retrieval realizes real-time interpretation of traffic signs, similar to human visual cognition.
4. Mobile Deployment: Lightweight model design enables execution on mobile devices with no loss of accuracy.

By comparing CV-TSD with prior works, it is evident that the novelty lies in the hybrid architecture, efficiency, and semantic image retrieval, which were absent in previous studies.

3.5 Limitations and Future Scope

While the CV-TSD outperforms the others, some future work may involve:

- Increasing the dataset by adding more signs and including other extreme weather conditions.
- Integrating more feature descriptors or attention mechanisms to further improve the accuracy of recognition.
- Real-time testing in mobile and in-vehicle deployments for on-road validation.

4 Conclusion

The proposed Computer Vision–Traffic Sign Detection technique presents a robust, efficient, and mobile-compatible framework for real-time traffic sign recognition. Key contributions and findings from the presented study are:

4.1 High Accuracy with Minimum Data

- CV-TSD reaches a validation accuracy of 97% based on 20–30 images per traffic sign class, beating the results of traditional HoG 64–89% and LBP 77–96% features.
- Cross-entropy loss decreased much, demonstrating that learning is sound and effective.

4.2 Hybrid Feature Extraction:

- Integration of HoG and LBP with CNN will be done using TensorFlow, which will have the ability for illumination-, weather-, and distortion-variant accurate recognition, which was missing in previous approaches.

4.3 Semantic Traffic Sign Retrieval

- Real-time detection of traffic signs using a query-based retrieval mechanism allows the implementation of human-like visual perception and provides meaningful contextual information for drivers.

4.4 Comparative Superiority:

- Comparing with the recent studies [1], [2], [3], we assure that CV-TSD performs on par or better using fewer training images with faster convergence and mobile-compatible deployment.

4.5 Practical Implications

- Traffic sign recognition can be efficiently and reliably provided for mobile applications, intelligent transportation systems, and driver assistance systems using lightweight architecture.

4.6 Future Prospects:

- Integration with larger, more diverse datasets including complex traffic environments.
- Enhancement of feature extraction using attention mechanisms or advanced deep learning architectures.
- Deployment in real-time, in-vehicle systems to support autonomous driving and smart city initiatives.

Takeaway: CV-TSD represents a novel contribution in traffic sign detection by combining classical feature descriptors with deep learning, enabling accurate, efficient, and mobile-compatible traffic sign recognition with minimal dataset requirements.

References

1) Sahu B, Pudale A, Kurudkar T, Poojary V. Real-Time Traffic Sign Detection and Recognition System using Computer Vision and Machine Learning. *IJARSCT*. 2025 Apr;5(4). Available from: https://www.ijarsct.co.in/A25171.pdf.
2) Megalingam RK, Thanigundala K, Musani SR, Nidamanuru H, Gadde L. Indian traffic sign detection and recognition using deep learning. *International Journal of Transportation Science and Technology*. 2023;12(3):683–699. 10.1016/j.ijtst.2022.06.002.
3) Zhao S, Yuan Y, Wu X, Wang Y, Zhang F. YOLOv7-TS: A Traffic Sign Detection Model Based on Sub-Pixel Convolution and Feature Fusion. *Sensors*. 2024;24(3):989. 10.3390/s24030989.
4) Tai SK, Dewi C, Chen RC, Liu YT, Jiang X, Yu H. Deep Learning for Traffic Sign Recognition Based on Spatial Pyramid Pooling with Scale Analysis. *Applied Sciences*. 2020;10(19):6997. 10.3390/app10196997.
5) Patil R, Ahire P, Bamane K, Patankar A, Patil PD, Badoniya S, et al. Real-Time Traffic Sign Detection and Recognition System using Computer Vision and Machine Learning. *Journal of Electrical Systems*. 2024;20(2):2244–2254. 10.52783/jes.1991.
6) Wang Z. Research on Flower Image Classification Based on Transfer Learning. *Academic Journal of Science and Technology*. 2022;4(3):183–186. Available from: https://drpress.org/ojs/index.php/ajst/article/view/5055.
7) Choudhury T, Sahu R, Mahapatra S. Traffic Sign Detection Using YOLOv5 and Data Augmentation Techniques. *Journal of Intelligent Transportation Systems*. 2023;27(5):655–667. 10.1080/15472450.2023.2198472.
8) Li X, Zhang Y, Wang J. Improved Deep Learning Model for Real-Time Traffic Sign Recognition. *IEEE Access*. 2021;9:125897–125908. 10.1109/ACCESS.2021.3112304.
9) Kumar P, Singh A, Rajan S. A Lightweight CNN Model for Traffic Sign Recognition in Autonomous Vehicles. *Electronics*. 2022;11(4):648. 10.3390/electronics11040648.