© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12-1719091

CNN-Based Traffic Signal Detection for Low-Visibility Scenarios

ROHAN THAN AGE[1], ONKAR SHETE[2], UTKARSH TOPE[3], DR. D. V. GORE[4]
[1, 2, 3, 4] *Department of Computer Engineering* PES *Modern College of Engineering* Pune, India

Abstract- ***Traffic signal detection is a crucial component of intelligent transportation systems (ITS), enabling safer and more efficient road usage. However, accurately identifying traffic signals under adverse environmental conditions such as fog, rain, and low-light environments remains a significant challenge. Traditional image processing techniques often fail to provide reliable results due to poor visibility and environmental noise. This project presents a Convolutional Neural Network (CNN)-based approach for robust traffic signal detection in low-visibility scenarios. The proposed system utilizes a deep learning model trained on a diverse dataset of traffic signal images collected under various weather conditions. Preprocessing techniques such as image resizing, normalization, and contrast enhancement are applied to improve feature extraction and model performance. The model is developed using Python with TensorFlow and scikit-learn, and evaluated based on performance metrics including accuracy, precision, and recall. The system is designed to classify traffic signals into different categories such as red, yellow, and green with high reliability. A simple graphical user interface (GUI) is also implemented using Tkinter to facilitate user interaction. The expected outcome of the project is an efficient and accurate traffic signal detection system capable of operating under challenging environmental conditions. This system can be further integrated into autonomous vehicles, driver assistance systems, and smart traffic management solutions to enhance road safety and automation.***

I. INTRODUCTION

In recent years, the rapid advancement of Intelligent Transportation Systems (ITS) has significantly improved road safety, traffic efficiency, and urban mobility. A key component of these systems is traffic signal detection, which enables vehicles and monitoring systems to recognize and respond appropriately to traffic lights. This technology plays a vital role in applications such as autonomous vehicles, driver assistance systems (ADAS), and smart traffic control.

Accurate detection of traffic signals is essential for ensuring safe driving decisions. However, real-world conditions often present challenges that affect detection performance. Environmental factors such as fog, rain, shadows, glare, and low-light conditions can significantly reduce visibility, making it difficult for traditional image processing techniques to identify traffic signals reliably. These limitations can lead to incorrect predictions and increase the risk of road accidents.

To address these challenges, deep learning techniques, particularly Convolutional Neural Networks (CNNs), have emerged as powerful tools for image recognition and object detection tasks. CNNs are capable of automatically learning complex features from images, making them highly effective in handling variations in lighting, weather conditions, and backgrounds.

This project proposes a CNN-based traffic signal detection system designed specifically for low-visibility scenarios. The system utilizes a dataset consisting of traffic signal images captured under various environmental conditions, including foggy, rainy, and nighttime settings. Image preprocessing techniques such as resizing, normalization, and contrast enhancement are applied to improve visibility and feature extraction.

II. RELATED WORK

Several researchers have worked on traffic signal and traffic sign detection systems using deep learning and computer vision techniques. Earlier approaches mainly relied on traditional image processing methods such as color thresholding, edge detection, and shape analysis. However, these methods were highly sensitive to environmental changes such as fog, rain, shadows, and poor lighting conditions.

Computer Vision and deep learning techniques, especially Convolutional Neural Networks (CNNs), have significantly improved object detection accuracy in transportation systems. Krizhevsky et al.

introduced deep CNN architectures for image classification, which later became the foundation for modern traffic detection systems. Similarly, Simonyan and Zisserman proposed the VGGNet architecture, which improved feature extraction using deeper neural networks.

Recent studies have focused on improving detection performance under adverse weather conditions. Gao et al. proposed a CNN-based traffic sign detection framework that combines adverse weather classification, image enhancement, and YOLO-based detection modules. Their system achieved high accuracy under rain, snow, fog, and blurred conditions.

Ahmed et al. introduced the DFR-TSD framework for robust traffic sign detection in challenging weather conditions. Their approach used CNN-based enhancement networks to improve image quality before detection, resulting in significant improvements in precision and recall compared to traditional methods.

Liu et al. proposed Image-Adaptive YOLO (IA-YOLO), which adaptively enhances low-quality images before object detection. The model demonstrated improved detection performance in foggy and low-light environments.

III. SYSTEM ARCHITECTURE

The proposed CNN-Based Traffic Signal Detection System is designed to identify and classify traffic signals accurately under adverse weather conditions such as fog, rain, and low-light environments. The system architecture consists of multiple stages, including image acquisition, preprocessing, feature extraction, classification, and result display.

Input Image
↓
Image Preprocessing
↓
CNN Feature Extraction
↓
Traffic Signal Classification
↓
Prediction Output
↓
GUI Display

The working of the system begins with the image acquisition stage, where traffic signal images are collected from Kaggle datasets and other image sources. The dataset contains images captured under normal as well as unfavorable weather conditions to ensure that the model learns diverse traffic scenarios.

Since environmental factors such as fog, rain, and poor lighting reduce image clarity, preprocessing becomes an important part of the architecture. In this stage, all images are resized to a fixed resolution so that they can be processed uniformly by the CNN model. Normalization is applied to scale pixel values between 0 and 1, which helps in faster and more stable model training.

Additional preprocessing techniques such as contrast enhancement and noise reduction are used to improve visibility in low-light and foggy images. Data augmentation techniques like rotation, flipping, zooming, and brightness adjustment are also performed to increase dataset diversity and reduce overfitting.

IV. COMMITMENT AND VERIFICATION PROTOCOLS

Commitment and Verification Protocols (Mathematical Format) Let,

- I= {i_1, i_2, i_3, i_n} be the set of input traffic signal images.
- P(I) represent the preprocessing function applied to input images.
- F(I) represent feature extraction using CNN.
- C(F) represent classification of extracted features.

- O represent the final output class.

Commitment Protocol

1. Dataset Commitment

The system ensures that all dataset images are validated and categorized properly.

$$D=\{D_n,D_f,D_r,D_l\}$$

Where:

D_n= Normal images
D_f= Foggy images
D_r= Rainy images
D_l= Low-light images

2.Preprocessing Commitment

Each input image undergoes preprocessing before training and testing.

$$P(I)=N(R(I))$$

Where:

- R(I)= Resized image
- N(I)= Normalized image

Additional enhancement:

$$E(I)=C(I)+D_A$$

Where:

- C(I)= Contrast enhancement
- D_A= Data augmentation

3. CNN Feature Extraction Commitment

The CNN extracts important features from the processed image.

$$F(I)=I*K$$

Where:

- I= Input image
- K= Convolution kernel/filter
- *= Convolution operation

Activation Function:

$$f(x)=\max(0, x)$$

(ReLU activation function)

4. Classification Commitment

The system classifies traffic signals into:

- Red
- Yellow
- Green

Softmax Function:

$$P(y_i) =e^{(z_i)}/(\sum_{j=1}^{n} e^{(z_j)})$$

Where:

- P(y_i)= Probability of class i
- z_i= Output score

Final output:

$$O = \arg\max(P(y_i))$$

Verification Protocol

1. Accuracy Verification

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

Where:

- TP= True Positive
- TN= True Negative
- FP= False Positive
- FN= False Negative

2. Precision Verification

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

3. Recall Verification

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

4. Balanced Accuracy Verification

$$Balanced\ Accuracy = \frac{Sensitivity + Specificity}{2}$$

5. System Verification Condition

$$V = \begin{cases} 1, & \text{if prediction = actual label} \\ 0, & \text{otherwise} \end{cases}$$

Where:

- V=1→ Correct detection
- V=0→ Incorrect detection

V. CNN LAYERS

© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12-1719091

1. Input Layer

The input layer receives the preprocessed traffic signal image. All images are resized into a fixed dimension before being passed into the CNN model. The input image contains pixel values representing traffic signals under different environmental conditions such as fog, rain, and low-light environments.

Example:

$$Input\ Image = 224 \times 224 \times 3$$

Where:

- 224 × 224 = Image dimensions
- 3 = RGB color channels

2. Convolution Layer

The convolution layer is the core layer of CNN responsible for feature extraction. It applies filters (kernels) to the input image to detect important visual features such as edges, textures, shapes, and colors of traffic signals.

Mathematical representation:

$$\left(I * K)(x,y) = \sum_m \sum_n I\,(m,n)K(x-m, y-n\right)$$

Where:

- I= Input image
- K= Kernel/filter
- *= Convolution operation

This layer helps the model identify traffic signal patterns.

3. Activation Layer (ReLU)

After convolution, the ReLU (Rectified Linear Unit) activation function is applied to introduce non-linearity into the network. It removes negative values and improves learning efficiency.

Mathematical function:

$$f(x) = \max\,(0, x)$$

ReLU helps the CNN learn complex features effectively.

4. Pooling Layer

The pooling layer reduces the dimensions of feature maps while preserving important information. This decreases computational complexity and prevents overfitting.

Commonly used pooling:

- Max Pooling
- Average Pooling

Max pooling operation:

$$P(x,y) = \max\,(R)$$

Where:

- R= Region of feature map

Pooling improves model efficiency and speed.

5. Flatten Layer

The flatten layer converts the 2D feature maps obtained from convolution and pooling into a 1D feature vector. This vector is then passed to the fully connected layer for classification.

Example:

$$Feature\ Map \rightarrow 1D\ Vector$$

6. Fully Connected Layer

The fully connected layer combines all extracted features and performs classification. It learns the relationship between features and output classes.

7. Output Layer (Softmax Layer)

The output layer uses the Softmax activation function to calculate the probability of each traffic signal class.

Mathematical representation:

$$P(y_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$$

Where:

- P(y_i) = Probability of class i
- z_i= Output score

The class with the highest probability is selected as the final output.

VI. IMPLEMENTATION

Dataset Collection

- Traffic signal images are collected from Kaggle datasets
- Dataset includes:
  - Normal traffic signal images
  - Foggy condition images
  - Rainy condition images
  - Low-light/night images

Image Preprocessing

- Resize images into fixed dimensions
- Normalize pixel values between 0 and 1
- Apply contrast enhancement for better visibility
- Remove noise from foggy and rainy images
- Perform data augmentation:
  - Rotation
  - Flipping
  - Zooming
  - Brightness adjustment

CNN Model Development

- CNN model is implemented using TensorFlow and Keras
- The architecture includes:
  - Convolution layers
  - ReLU activation layers
  - Pooling layers
  - Flatten layer
  - Fully connected layers
  - Softmax output layer

Model Training

- Training dataset is provided to the CNN model
- Multiple epochs are used for learning
- Adam optimizer is used for optimization
- Categorical cross-entropy is used as the loss function

Model Evaluation

- Model performance is evaluated using:
  - Accuracy
  - Precision
  - Recall
  - Balanced Accuracy

GUI Implementation

- Tkinter is used to develop the graphical user interface
- Users can:
  - Upload traffic signal images
  - View prediction results
  - Display detected traffic signal output

A. Summary

The proposed project, "CNN-Based Traffic Signal Detection for Low-Visibility Scenarios," focuses on developing an intelligent and reliable traffic signal detection system using deep learning techniques. The system is designed to accurately identify traffic signals under adverse environmental conditions such as fog, rain, and low-light environments, where traditional image processing methods often fail.

The project uses a Convolutional Neural Network (CNN) model for automatic feature extraction and classification of traffic signals into red, yellow, and green categories. Traffic signal images are collected from Kaggle datasets and undergo preprocessing operations such as resizing, normalization, contrast enhancement, and data augmentation to improve image quality and model performance.

B. Limitations and Trust Assumptions

We explicitly enumerate the system's limitations:
The system performance depends heavily on the quality and diversity of the dataset used for training.

Detection accuracy may decrease in extremely poor visibility conditions such as heavy fog, storms, or excessive glare.

The model is trained only for traffic signal detection and classification into red, yellow, and green categories.

Small or partially occluded traffic signals may not be detected accurately.

Real-time performance may require high computational resources and GPU support.

The system may produce incorrect predictions if images contain complex backgrounds or distorted signals.

The current implementation is limited to image-based detection and does not fully support live video stream processing.

Input images contain visible traffic signals within the camera frame.

C. Future Work

1) The proposed system can be extended for real-time traffic signal detection using live video streams and surveillance cameras.
2) Advanced deep learning models such as YOLO, ResNet, or EfficientNet can be integrated to improve detection speed and accuracy.
3) The system can be enhanced to detect additional road objects such as traffic signs, pedestrians, vehicles, and lane markings.
4) More diverse datasets containing extreme weather conditions such as snow, storms, and heavy fog can be used to improve robustness.
5) The project can be deployed on embedded systems such as Raspberry Pi or NVIDIA Jetson Nano for real-time implementation in vehicles.
6) Integration with autonomous vehicles and Advanced Driver Assistance Systems (ADAS) can improve road safety and intelligent driving.

REFERENCES

[1] Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017.

[2] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv preprint arXiv:1409.1556, 2014.

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

[4] P. Sharma, A. Gupta, and R. Singh, "Traffic Light Detection under Adverse Weather Conditions Using CNN," IEEE Access, vol. 9, pp. 112430–112441, 2021.

[5] M. Hossain and A. Rahman, "Improved Traffic Light Recognition Using Convolutional Neural Networks under Low Visibility Conditions," Scientific Reports, vol. 13, no. 5, pp. 8891–8899, 2023.

[6] D. Chen and F. Luo, "Enhancing Road Safety Using CNN-Based Traffic Light Detection System," Scientific Reports, vol. 14, no. 2, pp. 2567–2576, 2024.

[7] R. Jain and A. Sinha, "Detection of Traffic Lights Using Transfer Learning in Adverse Conditions," IEEE Region 10 Conference (TENCON), 2022.

[8] X. Zhang, P. Sun, and J. Wang, "Multi-Weather Condition Robust Traffic Signal Detection Using CNNs," IEEE Access, vol. 10, pp. 55341–55352, 2022.

[9] R. Patel and K. Shah, "Smart Traffic Management Using CNN-Based Signal Detection and SQLite Integration," International Journal of Emerging Trends in Engineering Research, vol. 12, no. 8, pp. 145–152, 2024.

[10] S. Mehta, D. Patel, and A. Bhatt, "Foggy and Rainy Weather Image Classification Using CNN and Data Augmentation Techniques," International Journal of Computer Applications, vol. 185, no. 2, pp. 10–18, 2023.