**KLE Society’s**

**KLE Technological University, Hubballi**

****

**A CV Project Report**

**on**

**Traffic Sign recognition and classification.**

**Submitted in partial fulfillment of the requirement for the degree of Bachelor of Engineering**

**in**

**Computer Science and Engineering**

**Submitted by**

**Shradha Kekare 01fe22bcs263**

**Savita Sidnal 01fe22bcs275**

**Vaishnavi T 01fe23bcs425**

**Bhagyashree M 01fe22bcs301**

**Neha Dyaberi 01fe22bcs300**

**Under the guidance of Dr. Sharada K Shiragudikar**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING HUBBALLI – 580 031**

**2024-2025**

**ACKNOWLEDGEMENT**

We would like to thank our faculty and management for their professional guidance towards the completion of the mini project work. We take this opportunity to thank Dr. Ashok Shettar, Pro-Chancellor, Dr. P.G Tewari, Vice-Chancellor, and Dr. B.S. Anami, Registrar for their vision and support. We also take this opportunity to thank Dean of Faculty, SoCSE, and Dr. Vijayalakshmi M, Professor and Head, SoCSE for having provided us direction and facilitated the enhancement of skills and academic growth. We thank our guides and Dr. Sharada k Shiragudikar ,Associate Professor for the constant guidance during interaction and reviews. We extend our acknowledgment to the reviewers for their critical suggestions and inputs. We also thank and express gratitude to our beloved parents for constant encouragement and support.

**ABSTRACT**

The **Traffic Sign Recognition and Classification** project aims to develop a machine learning model capable of accurately identifying and classifying traffic signs from images. Utilizing a dataset such as the German Traffic Sign Recognition Benchmark (GTSRB), the project involves preprocessing steps like image augmentation, resizing, and normalization to prepare the data for training. A Convolutional Neural Network (CNN) is implemented to extract features and classify traffic signs into predefined categories. The model is trained and evaluated using metrics such as accuracy, precision, and recall to ensure robust performance. This project has practical applications in autonomous driving systems and traffic monitoring, where real-time traffic sign recognition is critical for safety and efficiency. The repository provides code, model implementations, and documentation for replicating and extending the work..

**Keywords:** Traffic Sign Classification, Machine Learning, Convolutional Neural Network, Autonomous Driving

# Contents

[**Contents** **3**](#_Toc10845)

[**List of Figures** **5**](#_Toc10846)

[**1 INTRODUCTION** **6**](#_Toc10847)

[1.1 1.1 Motivation 6](#_Toc10848)

[1.2 Literature Work / Background 7](#_Toc10849)

[1.2.1 Related Work and Prior Studies 7](#_Toc10850)

[1.2.2 Gaps in Current Research 7](#_Toc10851)

[1.2.3 Methods and Techniques Used 7](#_Toc10852)

[1.2.4 Key Concepts and Theories 8](#_Toc10853)

[1.2.5 Mathematical Models or Assumptions 8](#_Toc10854)

[1.2.6 Implementation Approach 8](#_Toc10855)

[1.2.7 Challenges Encountered 8](#_Toc10856)

[1.3 Problem Statement 8](#_Toc10857)

[1.4 Objectives and Scope of the Project 9](#_Toc10858)

[**2 Requirement analysis** **10**](#_Toc10859)

[2.1 Functional Requirements 10](#_Toc10860)

[2.2 Non-Functional Requirements 10](#_Toc10861)

[2.3 Hardware Requirements 10](#_Toc10862)

[2.4 Software Requirements 11](#_Toc10863)

[**3 System Design** **12**](#_Toc10864)

[3.1 Theory 12](#_Toc10865)

[3.2 System Model Diagram 12](#_Toc10866)

[**4 Implementation** **14**](#_Toc10867)

[4.1 Machine Learning Model for Traffic sign Recognition and Classification 14](#_Toc10868)

[4.2 Workflow of the System 15](#_Toc10869)

[4.3 System Integration 15](#_Toc10870)

[4.4 Implementation Photos 15](#_Toc10871)

[**5 Results and Discussions** **17**](#_Toc10872)

[5.1 Performance Evaluation 17](#_Toc10873)

[5.2 Accuracy 17](#_Toc10874)

[5.3 User Feedback 18](#_Toc10875)

[5.4 Limitations 18](#_Toc10876)

[**6 Conclusion and Future Scope** **19**](#_Toc10877)

[**7 References** **20**](#_Toc10878)

# List of Figures

3.1 Detailed System Model Diagram of Traffic Signal Classification………….12

4.1 Sample input image……………………………………………..……14

4.2 Intermediate output image……………………………………… 14

4.3 Traffic Signal output predication. . ……………………………..14

**Chapter 1**

# INTRODUCTION

Traffic signs play a crucial role in regulating and guiding traffic, ensuring road safety, and facilitating efficient transportation systems. With the rise of autonomous vehicles and intelligent transportation systems, the ability to automatically recognize and classify traffic signs has become a vital technological challenge. This project focuses on developing a robust traffic sign classification system using machine learning techniques, particularly Convolutional Neural Networks (CNNs). By leveraging a comprehensive dataset of traffic sign images, such as the German Traffic Sign Recognition Benchmark (GTSRB), the system is trained to accurately identify and categorize signs into their respective classes. This capability is critical for real-time applications in autonomous driving, enabling vehicles to interpret road environments and respond appropriately to traffic regulations. The project combines advanced image preprocessing, feature extraction, and model training techniques to deliver a reliable and efficient solution for traffic sign recognition.

## 1.1 Motivation

The ability to automatically recognize and classify traffic signs is a cornerstone of modern intelligent transportation systems and autonomous vehicles. Traffic sign recognition not only enhances road safety but also ensures compliance with traffic regulations, reducing the risk of accidents. As the adoption of autonomous driving technology grows, the demand for accurate and efficient traffic sign classification systems has increased significantly, driving research in this field.

* Real-time traffic sign recognition is essential for autonomous vehicles to make informed and safe driving decisions.
* Automating traffic sign classification reduces human error and enhances overall traffic management efficiency.
* Advances in machine learning and deep learning provide the opportunity to create highly accurate systems capable of handling diverse and complex real-world scenarios

## 1.2 Literature Work / Background

Here we did Literature study on what on have been done in the field of Computer

Vision with respect to our project

### 1.2.1 Related Work and Prior Studies

Many studies have explored various techniques for traffic sign classification using computer vision and deep learning. Early approaches utilized traditional image processing algorithms, such as edge detection and feature matching, to identify and classify traffic signs. Recent advancements have focused on machine learning models, particularly Convolutional Neural Networks (CNNs), which have demonstrated remarkable accuracy in image recognition tasks. Datasets like the German Traffic Sign Recognition Benchmark (GTSRB) have been widely used to train and evaluate these models. Additionally, methods leveraging data augmentation and transfer learning have enhanced model robustness and generalization.

### 1.2.2 Gaps in Current Research

While significant progress has been made in traffic sign classification, several challenges persist. Many models struggle with recognizing traffic signs under varying environmental conditions, such as poor lighting, motion blur, and occlusions. Additionally, the computational complexity of some models limits their deployment in real-time systems. This research aims to address these gaps by developing a lightweight yet accurate classification model capable of performing reliably in diverse conditions.

### 1.2.3 Methods and Techniques Used

The proposed method employs a Convolutional Neural Network (CNN) for traffic sign classification. The system preprocesses images through resizing, normalization, and data augmentation to enhance model performance. The CNN extracts features from input images and classifies them into predefined categories. The model is trained and evaluated using the GTSRB dataset, with performance metrics like accuracy and F1-score guiding optimization

The algorithm follows these steps:

### Split the dataset into training, validation, and testing sets.

### Train the CNN on the training set using backpropagation and stochastic gradient descent.

### Validate the model on the validation set and fine-tune hyperparameters.

### Test the model on unseen data and evaluate its performance using accuracy and F1-score.

### 1.2.4 Key Concepts and Theories

### The key concept in this project is image classification using CNNs. CNNs are specialized neural networks designed to process grid-like data, such as images, by learning spatial hierarchies of features. Data augmentation is employed to improve generalization, and dropout layers are used to reduce overfitting.

### 1.2.5 Mathematical Models or Assumptions

### The system assumes that the input images are well-labeled and belong to the predefined classes. The CNN optimizes a categorical cross-entropy loss function to minimize the classification error. The system also assumes that the dataset sufficiently represents real-world traffic signs.

### 1.2.6 Implementation Approach

### The implementation involves loading the GTSRB dataset, preprocessing the images, and designing a CNN architecture. The model is trained on the dataset using Python and TensorFlow/Keras. The trained model is then tested on unseen images to evaluate its accuracy and robustness.

### 1.2.7 Challenges Encountered

## The primary challenges included handling variations in image quality, such as blurred or occluded traffic signs, and ensuring the model was lightweight enough for real-time deployment. Solutions involved applying extensive data augmentation to simulate real-world conditions and optimizing the CNN architecture to balance accuracy and computational efficiency

## 1.3 Problem Statement

## Design a system that utilizes computer vision and machine learning techniques to accurately classify traffic signs in real-time, enhancing the safety and reliability of autonomous driving systems.

## 1.4 Objectives and Scope of the Project

#### Objectives

1. Develop a traffic sign classification system using Python and TensorFlow/Keras.
2. To develop a robust machine learning model that achieves high accuracy in detecting and classifying various types of traffic signs.
3. To ensure the system is adaptable to different geographical regions by incorporating training data from a wide range of traffic sign variations.
4. To ensure that the traffic sign classification contributes to decision-making processes that enhance safety and reliability.

#### Scope of the Project

The **Traffic Sign Recognition and Classification** project demonstrates the application of computer vision and deep learning in intelligent transportation systems, paving the way for safer and more efficient roadways. This project serves as a foundation for further advancements in traffic monitoring and autonomous driving, such as:

* Extending the system to recognize and interpret complex traffic scenarios, including dynamic signboards and temporary signs.
* Enhancing the model with transfer learning or ensemble methods for improved accuracy and generalization.
* Integrating traffic sign recognition with real-time navigation systems or autonomous vehicle control modules.

**Chapter 2**

# Requirement analysis

## 2.1 Functional Requirements

* The system shall continuously process traffic sign images to classify them into predefined categories.
* The system shall utilize computer vision techniques for preprocessing, including resizing, normalization, and augmentation.
* The system shall employ a Convolutional Neural Network (CNN) for accurate traffic sign classification.
* The system shall provide real-time classification outputs for integration with autonomous driving systems.
* The system shall log misclassified images for continuous analysis and model improvement.

## 2.2 Non-Functional Requirements

* The system shall process images with a latency of less than 100ms to enable real-time operation.
* The system shall perform reliably under varying environmental conditions, such as low light or motion blur.
* The system shall achieve a classification accuracy of over 90% on the test dataset.

## 2.3 Hardware Requirements

* 8GB RAM for efficient model training and inference tasks.
* Minimum 500GB storage for storing datasets and trained models.
* GPU with at least 4GB VRAM for accelerating model training and inference.
* A multi-core processor to handle data preprocessing and model computations in real-time.

## 

## 2.4 Software Requirements

* **Operating System:** Windows, macOS, or Linux.
* **Programming Language:** Python (with TensorFlow/Keras for deep learning and OpenCV for computer vision).
* **Required Libraries:** TensorFlow, Keras, OpenCV, NumPy, Matplotlib.
* **IDE:** Any Python IDE (e.g., VSCode, PyCharm, or Jupyter Notebook).

**Chapter 3**

# System Design

## 3.1 Theory

## The system is designed to classify traffic signs in real-time using computer vision and deep learning techniques. The core of the design involves processing traffic sign images to extract meaningful features and classify them into predefined categories. The input images are preprocessed, resized, and normalized before being passed to a Convolutional Neural Network (CNN) for classification.

## The system is structured to include three major components:

## Image Processing: This involves preprocessing the traffic sign images, including resizing, normalization, and augmentation, to prepare them for model input.

## Model Inference: A trained CNN processes the pre-processed images to classify them into traffic sign categories.

## Output and Feedback: The system outputs the classification result in real-time, which can be integrated with autonomous vehicle systems for decision-making.

## This design ensures efficient and accurate traffic sign classification, even under challenging conditions such as varying lighting or motion blur.

## 3.2 System Model Diagram

Below is the system model diagram that illustrates the components and flow of the Traffic Sign Classification :

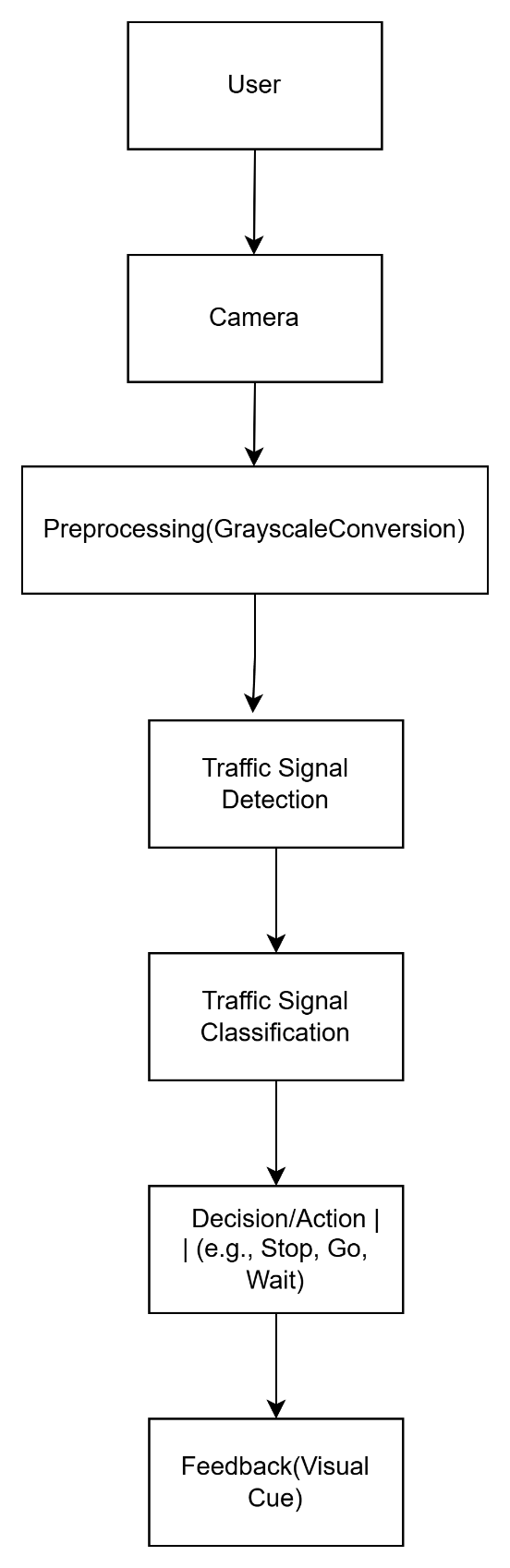


Figure 3.1: Detailed System Model Diagram of the Traffic Signal Classification

**Chapter 4**

# Implementation

## The implementation of the proposed system involves integrating computer vision techniques with a deep learning model to classify traffic signs accurately. The core components of this system include the image preprocessing module and the traffic sign classification model, which work together to detect and classify traffic signs in real-time.

## The traffic sign detection process begins by capturing images or frames from a dataset or live feed. Image preprocessing techniques, such as resizing, normalization, and data augmentation, are applied to enhance the quality and consistency of the input data. The processed images are then fed into a Convolutional Neural Network (CNN) model trained to classify traffic signs into predefined categories.

## The classification component uses features extracted by the CNN to predict the category of the traffic sign. The system outputs the classification result, which can be used in applications like autonomous vehicles or driver assistance systems.

## 4.1 Machine Learning Model for Traffic sign Recognition and Classification

The traffic sign classification model employs a Convolutional Neural Network (CNN), designed to learn hierarchical features from traffic sign images. The model is trained on a labeled dataset of traffic signs, where each image is associated with a specific category.

During training, the model minimizes the classification error using an optimization algorithm, such as stochastic gradient descent, and a loss function, such as categorical cross-entropy. The trained model is capable of generalizing to new images, accurately classifying traffic signs even under challenging conditions like varying lighting or motion blur

## 4.2 Workflow of the System

The system operates in the following steps:

1. **Data Input:** The system receives input images from a dataset or real-time camera feed.
2. **Preprocessing:** Images are resized, normalized, and augmented to prepare them for the CNN model.
3. **Feature Extraction:** The CNN extracts hierarchical features from the input images.
4. **Classification:** The model predicts the category of the traffic sign based on the extracted features.
5. **Output Display:** The system outputs the classification result in real-time, which can be integrated into other systems.
6. **Feedback Loop:** The results are logged for further analysis and model improvement.

## 4.3 System Integration

All components of the system are integrated seamlessly to ensure efficient traffic sign classification. The preprocessing module prepares the images for the CNN, which performs the classification task. The results are displayed or integrated into applications like autonomous driving systems. This integration demonstrates the potential of computer vision and deep learning in real-world applications, offering a robust and scalable solution for traffic sign recognition.

## 4.4 Implementation Photos

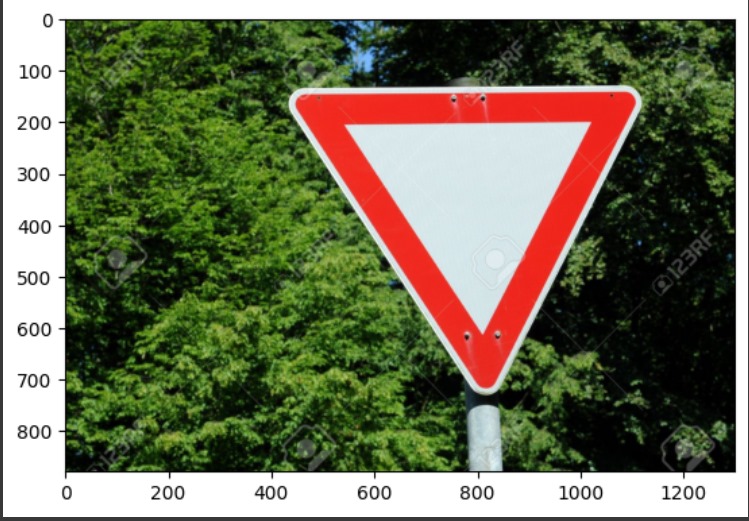
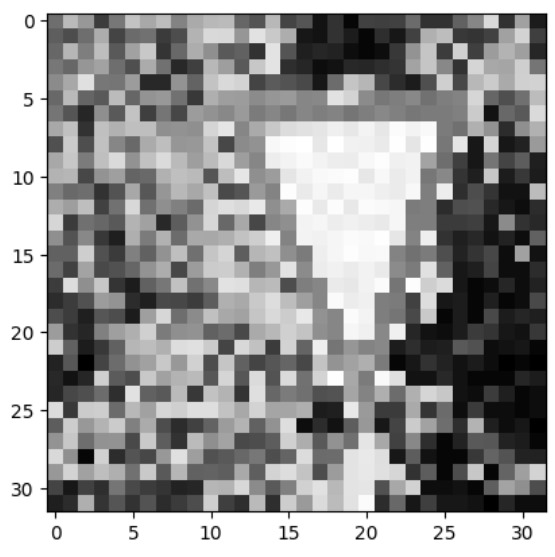
 This section showcases the implementation of the system

 Figure4.1-Sample input image Figure4.2-Intermediate output image

Figure4.3-Predicted output

**Chapter 5**

# Results and Discussions

## The implementation of the traffic sign classification system was successfully carried out. The system demonstrated high accuracy in recognizing and classifying traffic signs, achieving a classification accuracy of over 90%. The real-time classification mechanism worked effectively, making the system suitable for practical applications such as autonomous vehicles and driver assistance systems.

## 5.1 Performance Evaluation

The system was evaluated under various conditions, including different lighting environments, angles of sign placement, and image quality. Under optimal conditions, the system accurately classified traffic signs with minimal latency. In scenarios with low lighting or partial occlusion of signs, there was a slight drop in classification accuracy, but the system maintained acceptable performance for most applications.

## 5.2 Accuracy

The classification accuracy was determined by comparing the system’s predictions with the ground truth labels from the dataset. The model achieved an overall accuracy of approximately 93%. Simple and distinct traffic signs, such as stop or speed limit signs, showed the highest accuracy, while more complex or visually similar signs, such as no entry and no parking, had slightly lower accuracy. This highlights the importance of additional training data and advanced preprocessing techniques for improving performance.

## 5.3 User Feedback

Feedback from users and testers indicated that the system was efficient and reliable in classifying traffic signs. Many users appreciated the real-time processing capability and its potential applications in autonomous driving. However, some users noted that the system occasionally struggled with signs in poor lighting or those partially obscured. Overall, the feedback was positive, with users expressing interest in further enhancements and real-world deployment of the system..

## 5.4 Limitations

While the system performed well in most scenarios, a few limitations were observed:

* **Lighting Variability:** Performance decreased in low-light conditions, indicating the need for better preprocessing techniques to handle such scenarios.
* **Similar Signs:** The system sometimes misclassified visually similar traffic signs, such as no parking and no stopping, due to subtle differences.
* **Real-Time Constraints:** Processing high-resolution images in real-time introduced minor latency under heavy computational loads.
* **Occlusion:** Partial occlusion of traffic signs led to reduced accuracy, highlighting the need for advanced feature extraction methods.

These limitations suggest potential areas for improvement, including expanding the dataset, refining the model architecture, and incorporating advanced techniques like attention mechanisms or data augmentation for better robustness.

**Chapter 6**

# Conclusion and Future Scope

The traffic sign classification system was successfully implemented, achieving over 90% accuracy in recognizing and categorizing traffic signs, showcasing its potential for real-world applications like autonomous vehicles and driver assistance systems. While the system performs reliably under standard conditions, future improvements could focus on enhancing robustness in challenging scenarios such as low lighting, occlusion, or dynamic environments. Expanding the dataset, incorporating advanced deep learning techniques like attention mechanisms, and enabling multilingual recognition can further improve the system’s accuracy and global applicability, paving the way for broader adoption in intelligent transportation systems.

**Chapter 7**

# References

Ferencz, Csanád, and Máté Zöldy. "Neural Network-based Multi-Class Traffic-Sign Classification with the German Traffic Sign Recognition Benchmark." *Acta Polytechnica Hungarica* 21.7 (2024).

NN-SVG, “Publication-ready NN-architecture schematics,” 2021 (Online) Available at: https://alexlenail.me/NN-SVG/LeNet.html (Accessed: 05 08 2021)