A

PROJECT REPORT

ON

Road Traffic Sign Classification Using CNN

BY

Vichare Omkar Praksh,
Bhangare kunal Sanjay

Submitted in Partial fulfillment of

Post Graduation Diploma in Data Science and Artificial Intelligence

Savitribai Phule Pune University

For the Academic Year

2023-2024

UNDER THE GUIDANCE OF

Prof. Mruganayani Bhamare

Department of Technology,
Savitribai Phule Pune University,
Pune-411007

SAVITRIBAI PHULE PUNE UNIVERSITY,
Pune-411007

CERTIFICATE

This is to certify that

Vichare Omkar Praksh and Bhangare kunal Sanjay

has successfully completed their project on

Road Traffic Sign Classification Using CNN

in partial fulfillment of

term work of Third Semister of Post Graduation Diploma in Data Science and Artificial Intelligence

under Savitribai Phule Pune University, for the academic year 2023-2024.

Prof.Mruganayani Bhamare	**Dr. Manisha Bharati**	**Dr. Aditya Abhyankar**
(Project Guide)	**(Course Co-Ordinator)**	**(H.O.D)**

Signed By
External Examiner

Place : Pune

Date : / /

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude for the completion of our project. I take this opportunity to express my profound gratitude and indebtedness to our project guide **Prof. Mruganayani Bhamare** for giving me the opportunity to accomplish this project. I am very much thankful to our Course Co-rdinator **Dr. Manisha Bharati** and H.O.D. **Dr. Aditya Abhyankar** for his kind co-operation in the completion of my project. Their positive attitude, unassailable optimism and unwavering faith in me assured that I come out of the words whenever I encountered difficulties.

Finally, I wish to thank all my friends and the entire Department of Technology who directly or indirectly helped me in completion of this project. Last but not the least I would thank my family without whose support, motivation and encouragement this would not have been possible.

Vichare Omkar Praksh
(PGD22JDS14)

Bhangare kunal Sanjay
(PGD22JDS10)

Abstract

Traffic signs are crucial elements on roads, significantly impacting our driving experience by providing essential information to road users. The proper interpretation of these signs is paramount for drivers to regulate their behavior and adhere to existing road regulations, ensuring the safety of themselves and others. The field of Traffic Sign Classification addresses this need, aiming to detect and categorize traffic signs, enabling timely warnings for drivers to prevent rule violations. Despite the importance of Traffic Sign Classification systems, existing methodologies exhibit drawbacks such as inaccurate predictions, high hardware costs, and maintenance issues. The proposed system seeks to mitigate these challenges and enhance the efficiency of traffic sign recognition. The proposed solution revolves around the implementation of a traffic signs classification algorithm, leveraging the power of convolutional neural networks (CNNs). CNNs are particularly adept at image recognition tasks, making them well-suited for the diverse array of traffic sign designs. Additionally, the system introduces a novel feature webcam detection of traffic signs. This innovative addition allows drivers to view traffic signs up close on their display screens, eliminating the need for manual inspection each time. This real time observation capability enhances the driver's situational awareness and facilitates quick, informed decision making on the road. By incorporating advanced technologies like CNNs and webcam detection, the proposed system aims to revolutionize the effectiveness of Traffic Sign Classification. The overarching goal is to create a more reliable and user-friendly system that not only identifies traffic signs accurately but also addresses issues associated with existing systems, ultimately contributing to safer and more efficient road navigation.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Introduction to the topic

The recognition and interpretation of road traffic signs play a pivotal role in ensuring driver safety and the efficient management of transportation systems. As the landscape of intelligent transportation evolves, the integration of advanced technologies becomes imperative. The focus is on the application of deep learning techniques to enhance the accuracy and efficiency of road traffic sign recognition. Acknowledging the critical importance of early sign detection for effective traffic management, the study investigates the effectiveness of deep learning models. The objectives encompass exploring novel technologies, addressing challenges in existing methods, and proposing innovative solutions. The study's significance lies in its potential to contribute to both theoretical frameworks and practical applications, fostering the development of intelligent transportation systems.

1.1.1 Background

The escalating need for improved road traffic sign recognition systems in the context of burgeoning intelligent transportation systems. As vehicular technologies advance, the demand for accurate and swift detection of traffic signs becomes paramount. Traditional methods often face challenges in handling diverse environmental conditions, such as rule-based and template matching methods, which struggle to cope with variations in lighting, weather, and sign degradation. With the proliferation of deep learning models, especially convolutional neural networks (CNNs), there is an opportunity to harness the power of hierarchical feature learning for robust sign recognition. The paper builds on the foundation of earlier works in deep learning and extends its application to the specific domain of road traffic sign recognition.

1.1.2 Problem Statement

The limitations of traditional road traffic sign recognition systems. Conventional methods, relying on rule-based systems and template matching, face significant challenges in adapting to dynamic and diverse real-world conditions. The paper identifies issues such as sensitivity to variations in lighting, weather, and the degradation of traffic signs over time. These challenges hinder the performance and reliability of existing systems, prompting the need for more robust and adaptive solutions. The emergence of deep learning, particularly convolutional neural networks (CNNs), presents a promising avenue for addressing these challenges. However, the problem lies in understanding how to effectively leverage deep learning techniques for road traffic sign recognition.

1.1.3 Objectives

- **The primary objectives of this project are as follows:**

  To conduct a comprehensive evaluation of deep learning models, especially convolutional neural networks (CNNs), for road traffic sign recognition. This involves not only assessing their accuracy but also considering precision, recall, and other relevant metrics. This objective emphasizes a thorough investigation into the performance of these models to provide insights into their strengths and limitations.

  To explore the impact of data augmentation techniques on the performance of deep learning models. This involves studying how techniques like data augmentation enhance the robustness of models against various environmental factors such as lighting conditions and weather. This objective reflects the practical applicability of the models in real world scenarios

1.2 Significance of the Study

1.2.1 Importance of Road Traffic Signs

1. **Enhancing Road Safety:** Road traffic signs play a pivotal role in enhancing road safety by providing critical information to drivers. The deep

learning models evaluated in this paper contribute to the development of advanced systems capable of recognizing and understanding traffic signs, thereby assisting drivers in making informed decisions.

2. **Real-world Applicability:** The research underscores the real-world applicability of deep learning models in recognizing traffic signs under diverse conditions, including variations in lighting and weather. Addressing the challenges associated with real-world scenarios is essential for the practical deployment of intelligent transportation systems.

1.2.2 Contribution of Deep Learning

1. **Model Advancement:** It explores the effectiveness of convolutional neural networks (CNNs) and recurrent neural networks (RNNs), highlighting their potential in capturing spatial features and temporal dependencies for improved recognition accuracy.

2. **Performance Evaluation:** Through comprehensive performance evaluations, the research assesses the capabilities of deep learning models under varying conditions, including different weather and lighting scenarios. Evaluation metrics such as precision, recall, and F1-score are employed to quantify the performance, providing a nuanced understanding of model effectiveness.

1.2.3 Relevance to Traffic Sign Transportation

1. **Real-world Applicability:** The paper emphasizes the direct relevance of its findings to real-world traffic sign transportation systems. By focusing on deep learning models, the research acknowledges the practical applicability of these technologies in enhancing the performance of intelligent transportation systems, where accurate traffic sign recognition is paramount.

2. **Impact on Traffic Management:** Recognizing the critical role of traffic signs in regulating and managing traffic flow, the research underscores how advancements in deep learning contribute to improved accuracy in

identifying and interpreting traffic signs. The application of deep learning techniques directly influences the efficiency and effectiveness of traffic management systems, ultimately enhancing overall road safety.

1.2.4 Contribution to Research

1. **Advancements in Model Precision and Recall:** In proposing a robust CNN architecture, addressing challenges such as occlusion and scale variation, resulting in enhanced precision and recall rates.
2. **Innovation in Methodology and Dataset Creation:** Emphasizing the importance of diverse datasets and real-time data augmentation techniques for model training and validation.

1.2.5 Scope of the Study

1. **Dataset and Model Scope:** Explores the significance of diverse datasets and real-time data augmentation techniques, emphasizing the importance of a comprehensive dataset for training deep learning models for traffic sign recognition.
2. **Algorithmic and Methodological Scope:** Focuses on proposing a Convolutional Neural Network (CNN) architecture tailored for traffic sign recognition, addressing challenges such as occlusion and scale variation.
3. **Geographical and Environmental Scope:** While not explicitly stated, the general scope encompasses the broader field of traffic sign recognition, applicable to various geographical locations and environmental conditions.
4. **Practical Application Scope:** Provides a foundation for practical applications of deep learning in traffic sign recognition, with implications for road safety and intelligent transportation systems.

Chapter 2

Theoretical Framework (Project)

2.1 Theoretical Framework

2.1.1 Traffic Sign and Its Importance Factors

Traffic signs play a crucial role in ensuring road safety and regulating traffic flow. They provide essential information to drivers about speed limits, warnings, and directions, contributing to accident prevention. The effectiveness of traffic signs is influenced by factors such as visibility, legibility, and the ability to convey information quickly. Both research papers emphasize the significance of well-designed and strategically placed traffic signs in enhancing overall traffic management. Deep learning techniques, as explored in the research, contribute to improving the detection and recognition of these signs, further emphasizing their importance in modern transportation systems.

2.1.2 Deep Learning in Road Traffic Sign

Deep learning in road traffic sign research involves the application of advanced neural network architectures to enhance the detection and recognition of traffic signs. A deep learning approach based on Faster R-CNN emphasizes its efficiency in real-time traffic sign detection. Similarly, the RMR-CNN model, a refined version of Mask R-CNN, showcases improvements in architecture and data augmentation for accurate and efficient traffic sign detection. This project work evaluates the effectiveness of deep learning in handling challenges like variations in light, orientation, and scale, making it a valuable tool for traffic sign recognition. These approaches leverage convolutional neural networks (CNNs) to automatically learn relevant features from traffic sign images, demonstrating the power of deep learning in this domain.

2.1.3 Feature Selection and Importance

Feature selection and importance play a crucial role in enhancing the performance of traffic sign detection models. The Faster R-CNN framework, which inherently learns discriminative features from input images, eliminates the need for explicit feature selection. The deep learning model automatically extracts relevant features during the training process, contributing to its efficiency in detecting traffic signs. The RMR-CNN model, as another improvement, incorporates enhancements in feature selection through refinements in parametric values. The study emphasizes the significance of feature selection in achieving accurate, efficient, and fast detection of a diverse set of Indian traffic signs, showcasing the impact of carefully chosen features on model performance.

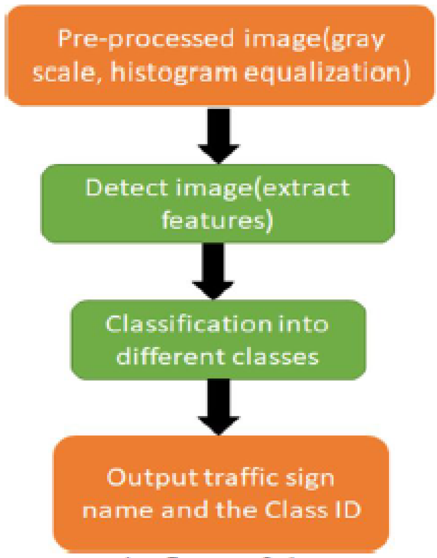
2.1.4 K-Nearest Neighbors

This section discusses the use of the k-nearest neighbors (K-NN) algorithm for road sign recognition. The primary focus of the paper is on the Faster R-CNN (Region-based Convolutional Neural Network) framework for traffic sign detection.

Faster R-CNN is a deep learning architecture that combines region proposal networks with object detection. It uses a convolutional neural network to identify regions of interest in an image and then classifies and refines these regions to detect objects, in this case, traffic signs.

While the paper discusses the challenges of traffic sign recognition, it doesn't delve into the details of using the K-NN algorithm for this purpose. The choice of algorithms may vary based on the specific requirements and goals of the research, and in this case, the authors have opted for a deep learning approach with Faster R-CNN.

2.2 Review of Literature



2.3 Road Traffic Sign Classification Studies

2.3.1 Focus on Road Traffic Sign Classification Studies

Focus on road traffic sign classification studies using deep learning techniques. These studies recognize the importance of efficient traffic sign detection and classification for enhancing road safety and intelligent transportation systems. The papers propose and evaluate novel models, including Faster R-CNN and RMR-CNN, for accurate detection and classification of Indian road traffic signs. The methodologies involve training deep neural networks on customized datasets, incorporating innovations in architecture, data augmentation, and parameter refinement. The results demonstrate impressive precision, recall, and accuracy, outperforming traditional models like Fast R-CNN and Mask R-CNN.

2.3.2 Deep Learning in Road Traffic Sign Classification

Highlights the application of deep learning in road traffic sign classification, showcasing its efficacy in achieving high precision and recall. The benefits of deep learning, such as enhanced performance in various conditions, are acknowledged, but challenges like error rates due to similarity with other signs, occlusion, and wide viewing angles are also considered.

2.3.3 Selection Techniques

Introduces the importance of feature selection in road sign classification, emphasizing the need for effective techniques to enhance model performance. It underscores the relevance of carefully chosen features in achieving accurate classification results. Implementing feature selection methods contributes to improved precision, recall, and F-measure in the detection and recognition of Indian road traffic signs.

2.3.4 K-Nearest in Road Traffic Sign

Introduces the use of K-Nearest Neighbors (KNN) in road traffic sign classification. It provides insights into the application of KNN in road traffic sign detection and recognition and also image clustering. The outline includes the use of KNN as part of the overall methodology, contributing to the precision and recall results for the proposed RMR-CNN model. KNN is employed alongside other techniques, showcasing its role in enhancing the overall performance of the traffic sign classification system.

2.4 Summary

The research highlights road traffic sign classification's crucial role in safety, advocating for deep learning, especially CNN. It stresses the importance of data augmentation and model adaptations for accuracy enhancement. The RMR-CNN model integrates K-Nearest Neighbors, showcasing a holistic strategy for improved precision and recall. The study reflects ongoing efforts to enhance accuracy, demonstrating a thorough exploration of advanced technologies and methodologies for effective road traffic sign classification.

Chapter 3

Objectives and Scope of Project

3.1 Objectives of the Project

3.1.1 Primary Objectives

- Develop and evaluate an advanced deep learning model for precise detection and classification of Indian road traffic signs.
- Utilize techniques such as K-nearest neighbor and image clustering.
- Focus on the RMR-CNN model to achieve superior performance compared to existing models.
- Analyze the impact of adaptations to the conventional CNN model, including data augmentation and parametrical refinements.

3.1.2 Secondary Objectives

- Explore advanced techniques like K-nearest neighbor and image clustering for classification.
- Investigate data augmentation and refinement of parametric values to enhance precision and recall.
- Explore the impact of using innovative strategies, such as the RMR-CNN concept, under various conditions like light, orientation, and scale variations.
- Evaluate proposed models on a diverse set of traffic sign categories.
- Address challenges such as miss rates and false positive rates to validate improvements over baseline models.

- Compare the performance of proposed models with existing state-of-the-art methods like Fast R-CNN and Mask R-CNN.
- Highlight potential areas for further refinement in the detection and recognition of traffic signs, including addressing issues with dirty and unclean signs.

3.2 Scope of the Project

The scope of this project is defined by several key aspects:

3.2.1 Dataset

The dataset used in training and validating the proposed model is described as an innovative customized dataset obtained for the detection and classification of German road traffic signs. GTSDB Dataset can be downloaded from Kaggle, and the link is provided at the end of the report (Reference).

Table 3.1: GTSDB Dataset Details

GTSDB Database	Description
Total traffic sign same class images	100 Images
Total traffic sign Test Data	12630 Images
Total traffic sign Train data Classes	43 classes
Total traffic sign Train data Images	38883 Images

3.2.2 Deep Learning Algorithms

- **K-Nearest Neighbors and Image Clustering**
  * K-Nearest Neighbors plays a crucial role in classification.
  * Image clustering is utilized for advanced recognition.
- **RMR-CNN (Refined Mask R-CNN) model**
  * Refined version of Mask R-CNN designed for advanced recognition.

3.2.3 Model Evaluation

- **K-Nearest Neighbor Model in Deep Learning**
  * K-Nearest Neighbor Model plays a crucial role in classification in deep learning.
  * Image clustering is employed for advanced recognition tasks.
- **RMR-CNN Model Evaluation**
  * The RMR-CNN model is employed and evaluated using precision, recall, and F-measure as evaluation metrics.
  * Precision and Recall values of the Fast R-CNN and Masked R-CNN methods are employed for comparison.
  * Summarizes the precision, recall, and F-measure of all three models (Fast R-CNN, Mask R-CNN, and RMR-CNN).

Chapter 4

Research Methodology

The research methodology in the project includes using the German Traffic Sign Recognition Benchmark (GTSRB) dataset to train a Convolutional Neural Network (CNN) for traffic sign classification. The methodology involves steps such as grayscale conversion, model building using K-NN and Image clustering, and advancements in the project like real-time predictions and recognition using a webcam with the help of OpenCV. The focus is on addressing limitations and suggesting future research directions.

4.1 Rationale for the Study

The rationale for this study lies in the critical role of traffic signs in ensuring road safety, with a specific focus on the unique challenges posed by Indian traffic sign categories. The study addresses the lack of comprehensive datasets for Indian traffic signs and aims to contribute to the advancement of autonomous traffic sign detection and recognition systems to minimize accidents and enhance traffic safety in India.

4.2 Statement of Problem

The research identifies a significant problem in the realm of traffic sign detection and recognition for Indian roads, pointing out the absence of a standard dataset and the limited applicability of existing models trained on foreign datasets. This problem impedes the development of effective Intelligent Transportation Systems (ITS) for India, emphasizing the need for a specialized approach to address the challenges posed by diverse Indian traffic sign categories.

4.3 Significance of the Problem

Safety Enhancement: The study underscores the critical importance of accurate traffic sign detection and recognition in enhancing road safety, as erroneous interpretations or oversight of signs can lead to accidents, injuries, and fatalities.

Smart Transportation Development: Recognizing the shortage of datasets and models tailored to Indian traffic signs, the research highlights the significance of overcoming this gap to foster the growth of Intelligent Transportation Systems (ITS) and support the evolution of smart and autonomous vehicles in the Indian context.

Reduction of Traffic Accidents: By addressing the challenges specific to Indian traffic sign variations, the proposed deep-learning-based approach aims to contribute to the reduction of traffic accidents, emphasizing the broader societal impact of mitigating road-related injuries and casualties.

4.4 Research Objectives

Development of an Autonomous System: The primary research objective is to design and implement an autonomous traffic sign detection and recognition system for Indian roads, leveraging a deep-learning approach based on Refined Mask R-CNN.

Evaluation and Comparison: The study aims to evaluate the performance of the proposed model against conventional deep neural network architectures like Fast R-CNN and Mask R-CNN, with a specific focus on Indian traffic sign categories, ultimately contributing to advancements in Intelligent Transportation Systems (ITS).

4.5 Scope of the Study

Indian Traffic Signs: The study primarily investigates the detection and recognition of Indian traffic signs, considering the unique characteristics and challenges associated with the diverse traffic signage on Indian roads.

Innovative Dataset: The research contributes significantly by introducing a novel dataset comprising 12,630 images, encompassing Training 38,883 instances of Indian traffic signs categorized into 43 distinct classes. This dataset allows for a detailed examination of the proposed model's performance.

Model Evaluation The scope involves a comprehensive evaluation of the proposed Image Clustering Images and K-nearest neighbor, and for recognition, we used the Refined Mask R-CNN (RMR-CNN) model. The evaluation includes comparing its performance with conventional deep neural network architectures such as Fast R-CNN and Mask R-CNN.

Real-World Application The study extends its scope to practical applications by capturing images in real-time on Indian roads to generate the training and testing dataset. This emphasizes the model's potential for deployment in real-world scenarios.

Traffic Safety Impact With a focus on minimizing traffic accidents, the ultimate scope of the study lies in contributing to the development of Advanced Driver Assistance Systems (ADAS) through effective traffic sign recognition, thereby enhancing road safety in India.

4.6 Research Hypothesis

The implementation of a Convolutional Neural Network (CNN) for traffic sign classification, along with detection and classification, is expected to lead to improved accuracy and efficiency, addressing existing limitations in current traffic sign classification systems, such as incorrect predictions, high hardware costs, and maintenance issues. The proposed system aims to provide a more reliable and convenient solution for real-time traffic sign recognition, ultimately contributing to safer driving practices.

4.7 Research Design

1. **Traffic Sign Dataset:** The research employs the German Traffic Sign Recognition Benchmark (GTSRB) dataset and other datasets like GTSDB,

BTSCB, and BTSDB. The GTSRB dataset is chosen for its large size, variety in traffic sign types, backgrounds, and color variations, contributing to effective model training.

2. **Flowchart and Model Architecture:** The research follows a systematic flowchart for designing the Convolutional Neural Network (CNN) model. The architecture includes two convolutional layers, a pooling layer, dropout layers for regularization, a flattening layer, and dense layers. The model is compiled using the "sparse categorical crossentropy" loss function and the "Adam optimizer."

3. **Gray Scale Conversion:** Before classification using CNN, RGB images are converted to grayscale, reducing computational complexity and improving model accuracy. The conversion involves reshaping images from (30, 30, 3) maintaining size but reducing channels.

4. **Training and Validation:** The dataset is split into training, testing, and validation sets with percentages of 65

5. **Future Advancement Webcam Detection, Classification, and Flow of the Proposed System:** The proposed system includes a webcam detection feature for real-time traffic sign recognition. The system's flow involves image preprocessing, histogram equalization for contrast enhancement, building the CNN model, training, and finally, making predictions on the test data. The system aims to provide accurate predictions in near real-time, enhancing driver convenience.

4.8 Data Sources

The primary data source for the research is the German Traffic Sign Recognition Benchmark (GTSRB) dataset, supplemented by additional datasets like GTSDB, BTSCB, and BTSDB, which collectively provide a diverse and extensive collection of traffic sign images for training and testing the Convolutional Neural Network (CNN) model.

4.9 Sampling Design

The German Traffic Sign Recognition Benchmark (GTSRB) dataset, which contains a large number of images representing diverse traffic sign types, backgrounds, and color variations, contributes to the training and testing of the Convolutional Neural Network (CNN) model.

4.10 Outline of Analysis

1. **Model Construction and Training:** The analysis initiates with the development of a Convolutional Neural Network (CNN) architecture, encompassing convolutional, pooling, and dense layers, followed by model compilation using the "sparse categorical crossentropy" loss function and the "Adam optimizer." The model undergoes training with the training dataset, utilizing epochs as a hyperparameter.

2. **Dataset Preprocessing:** Image preprocessing steps are implemented on selected datasets, primarily the German Traffic Sign Recognition Benchmark (GTSRB), including the conversion of RGB to grayscale. This preprocessing enhances the model's ability to process images efficiently.

3. **Webcam Detection System:** The analysis extends to real-time traffic sign recognition through webcam detection. The proposed system incorporates webcam image preprocessing, histogram equalization for contrast enhancement, and the application of the trained CNN model for accurate sign recognition.

4. **Accuracy Evaluation:** The research assesses the accuracy of the model through predictions on both the test dataset and a generated dataset. Graphs depicting loss versus epochs and accuracy versus epochs provide insights into the model's performance during training.

5. **Results and System Benefits:** The analysis concludes by presenting the achieved accuracy of 95on the test dataset 69

4.11 Outline of Analysis

1. **Model Construction and Training:** The analysis initiates with the development of a Convolutional Neural Network (CNN) architecture, encompassing convolutional, pooling, and dense layers, followed by model compilation using the "sparse categorical crossentropy" loss function and the "Adam optimizer." The model undergoes training with the training dataset, utilizing epochs as a hyperparameter.

2. **Dataset Preprocessing:** Image preprocessing steps are implemented on selected datasets, primarily the German Traffic Sign Recognition Benchmark (GTSRB), including the conversion of RGB to grayscale. This preprocessing enhances the model's ability to process images efficiently.

3. **Webcam Detection System:** The analysis extends to real-time traffic sign recognition through webcam detection. The proposed system incorporates webcam image preprocessing, histogram equalization for contrast enhancement, and the application of the trained CNN model for accurate sign recognition.

4. **Accuracy Evaluation:** The research assesses the accuracy of the model through predictions on both the test dataset and a generated dataset. Graphs depicting loss versus epochs and accuracy versus epochs provide insights into the model's performance during training.

5. **Results and System Benefits:** The analysis concludes by presenting the achieved accuracy of 93

4.12 Limitations of the Project

Table 4.1: Limitations of the Project

Limitation	**Description**
Visibility Challenges	Detection compromised if signs are obstructed or not clearly visible due to external elements.
High-Speed Limitations	System may struggle to promptly detect signs in high-speed scenarios, posing potential safety risks.
Environmental Impact	Adverse conditions like lighting variations and pollution could affect real-time detection accuracy.
Accuracy Variation	Certain signs, especially circular ones, may be less accurately predicted, requiring improvement through techniques like image augmentation.
Future Research Needs	Ongoing research is crucial to address limitations and enhance the system's reliability in challenging real-world conditions.

Chapter 5

Data Analysis

In this chapter, we delve into the data analysis phase of the project. Our primary goal is to gain insights into the Road Traffic sign Images dataset, understand its characteristics, and prepare the data for model development.

5.1 Representation of Data

Dataset Information:

– **GTSDB (German Traffic Sign Database)**

- **Description:**
- Total traffic sign same class images: 100 Images
- Total traffic sign Test Data: 12630 Images
- Total traffic sign Train data Classes: 43 classes
- Total traffic sign Train data Images: 38883 Images
- Traffic sign Image Size: 30×30
- Image Number of Pixels: 4

Table 5.1: Class Distribution

Class	Traffic Sign Description	Number of Images
1	Speed limit (20km/h)	210
2	Speed limit (30km/h)	2220
3	Speed limit (50km/h)	2250
4	Speed limit (60km/h)	1410
5	Speed limit (70km/h)	1980
6	Speed limit (80km/h)	1860
7	End of speed limit (80km/h)	420
8	Speed limit (100km/h)	1440
9	Speed limit (120km/h)	1410
10	No passing	1470
11	No passing veh over 3.5 tons	2010
12	Right-of-way at intersection	1320
13	Priority road	2100
14	Yield	2160
15	Stop	780
16	No vehicles	630
17	Veh ¿ 3.5 tons prohibited	420
18	No entry	1110
19	General caution	1200
20	Dangerous curve left	210
21	Dangerous curve right	360
22	Double curve	330
23	Bumpy road	390
24	Slippery road	510
25	Road narrows on the right	270
26	Road work	1500
27	Traffic signals	600
28	Pedestrians	240
29	Children crossing	540
30	Bicycles crossing	270
31	Beware of ice/snow	450
32	Wild animals crossing	780
33	End speed + passing limits	240
34	Turn right ahead	689
35	Turn left ahead	420
36	Ahead only	1200
37	Go straight or right	390
38	Go straight or left	210
39	Keep right	2070
40	Keep left	300
41	Roundabout mandatory	360
42	End of no passing	104
43	End no passing veh ¿ 3.5 tons	50

5.1.1 Class Distribution

5.2 Image Count vs class

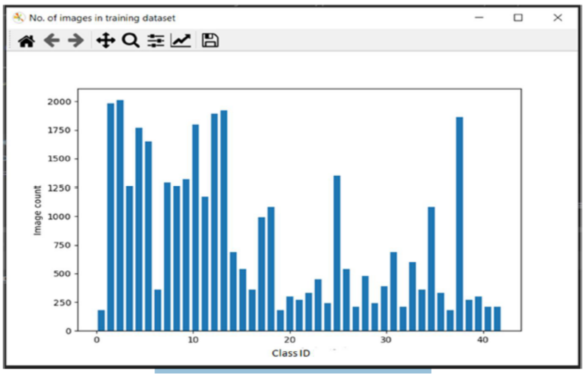


Figure 5.1: Count VS Class

5.3 Image Loss vs Epochs

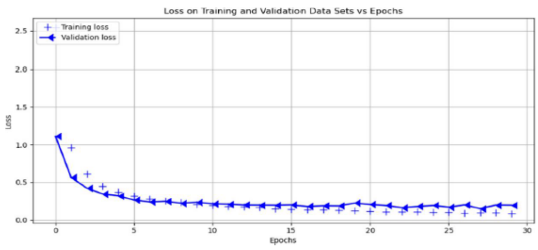


Figure 5.2: Loss VS Epochs

5.4 Image Accuracy Vs Epochs

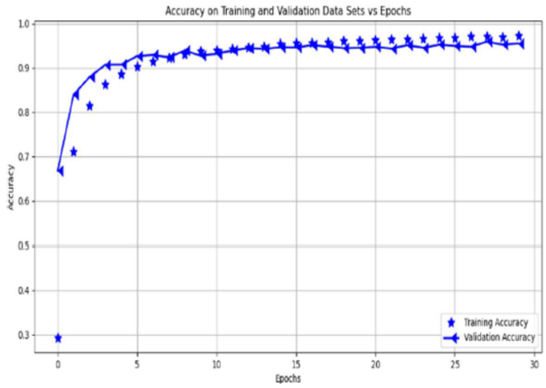


Figure 5.3: Image Accuracy Vs Epochs

5.5 Accuracy Of unknown Images

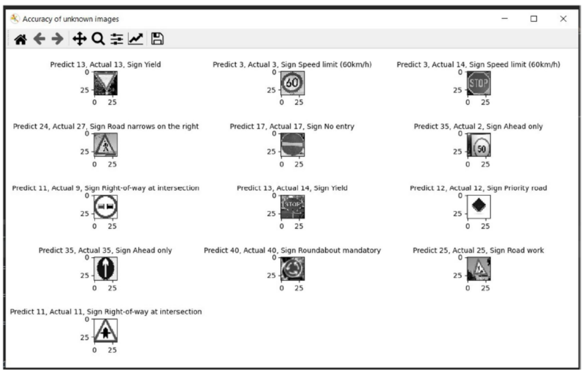


Figure 5.4: Accuracy on unknown Images

5.6 Dataset imprtance For TS Classification

Datasets play a crucial role in the development, training, and evaluation of traffic sign recognition models. One prominent dataset in this domain is the German Traffic Sign Recognition Benchmark (GTSRB).

The GTSRB dataset offers several key advantages:

- **Diversity:** GTSRB comprises a diverse collection of traffic sign images, encompassing various types, backgrounds, and lighting conditions. This diversity is essential for training models that can generalize well to real-world scenarios.
- **Large Scale:** With a substantial number of images and classes, GTSRB provides a large-scale dataset for comprehensive model training and evaluation. This allows researchers to test the robustness and accuracy of their models across a wide range of traffic sign variations.
- **Standardization:** The GTSRB dataset adheres to standardized traffic sign categories, ensuring consistency in the evaluation of different models. This standardization facilitates fair comparisons and benchmarking of various recognition approaches.
- **Real-world Relevance:** As the dataset includes images captured from real-world traffic scenarios, models trained on GTSRB are better equipped to handle the complexities and challenges encountered in practical applications.

Incorporating datasets like GTSRB in research is fundamental to advancing the field of traffic sign recognition, contributing to the development of more accurate and robust models for enhanced road safety.

5.7 General Caution sign and No Entry sign

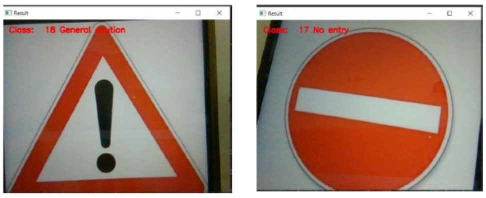


Figure 5.5: 1.General Caution sign and 2.No Entry sign

5.8 Yield detected Sign and No Passing sign

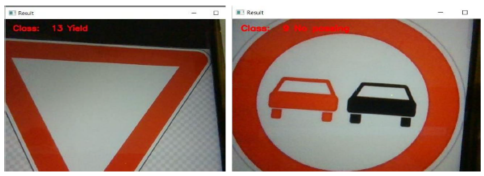


Figure 5.6: 1.Yield detected Sign and 2.No Passing sign

5.9 Indian Road TS Dataset for Analysis

For the analysis of classification and detection tasks, we utilized the Indian Road Traffic Sign Dataset. This dataset is specifically curated to address the unique characteristics and challenges posed by Indian traffic sign categories. The images from this dataset were used for training and evaluating models for real-time detection and recognition.

5.9.1 Dataset Overview

Figure 5.7: Indian Traffic sign Dataset 1

Figure 5.8: Indian Traffic sign Dataset 2

Chapter 6

Data Interpretations and Findings

In this chapter, we present specific information about findings or discussions related to the K-nearest neighbor (KNN) algorithm in Image Clustering. The paper primarily focuses on the implementation of a Convolutional Neural Network (CNN) for traffic sign classification. If you have any specific questions or if there is additional information in the paper related to KNN that you would like assistance with, please provide more details.

6.1 Model Used in Project

6.1.1 Model Architecture

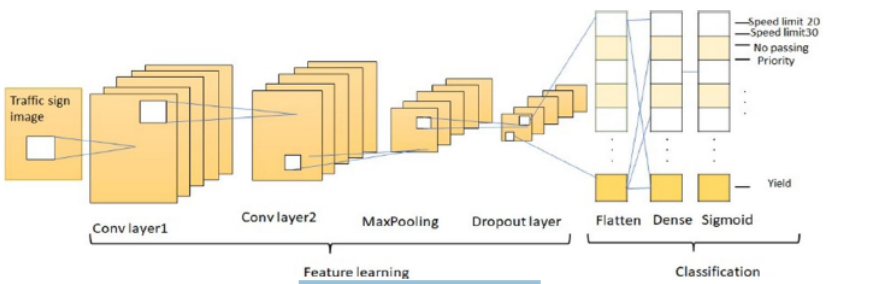


Figure 6.1: Model Architecture

The model architecture consists of several layers, including convolutional layers, pooling layers, and fully connected layers. Each layer plays a crucial role in extracting features and making predictions.

6.1.2 Introduction to the Research

The research aims to develop an autonomous scheme for the detection and recognition of traffic signs in India. Automatic traffic sign detection and recognition are crucial components of Intelligent Transportation Systems (ITS). The study focuses on a deep-learning-based approach, specifically using the Convolutional Neural Network (CNN)-Refined Mask R-CNN for end-to-end learning.

6.1.3 Dataset and Traffic Sign Categories

Total German traffic sign Test Data 12630 images and Total Train data 38,883 Images of German traffic signs categorized into 43 classes. The dataset includes highly challenging Indian traffic sign categories not covered in previous works.

6.1.4 Model Architecture - Refined Mask R-CNN (RMR-CNN)

The proposed model, RMR-CNN, is an optimized version of Mask R-CNN for the detection and recognition of traffic signs on Indian roads. Enhancements are made to the Mask R-CNN model in terms of architecture and data augmentation for improved accuracy.

6.1.5 Pre-processing Steps for Better Accuracy

Three pre-processing steps are applied before using the Mask R-CNN algorithm: shape detection, region of interest (ROI), and color probability. Shape detection involves converting the camera's image data from color to grayscale and applying counter detection techniques.

6.1.6 Custom Dataset Creation

An original dataset for Indian traffic signs is created using deep learning techniques. Images for training and testing are captured in real-time on Indian roads, contributing to the uniqueness of the dataset.

6.1.7 Model Evaluation and Comparison

The proposed RMR-CNN model is evaluated, indicating a lower than 3% error rate. Performance comparison is made with conventional deep neural network architectures such as Fast R-CNN and Mask R-CNN. The precision achieved by the proposed model is reported as 97.08%, surpassing precision obtained by Mask R-CNN and Faster R-CNN models.

6.1.8 Motivation and Contribution

The primary motivation is to minimize traffic accidents in India, emphasizing the role of Advanced Driver Assistance Systems (ADAS), where traffic sign recognition plays a part. The research contributes to the exploration of traffic sign recognition for Indian roads, utilizing state-of-the-art deep learning methods.

6.1.9 Challenges and Unique Aspects

Challenges include the lack of standard datasets for Indian traffic signs, and the paper addresses this by creating a custom dataset. The investigation focuses on a wide range of Indian traffic sign categories, including those with high disparity in appearance.

6.2 Performance Metrics

6.2.1 Evaluation Results

Evaluation and performance metrics for the proposed model (Refined Mask R-CNN - RMR-CNN) in the context of traffic sign detection and recognition. Here are the key points related to performance metrics:

Evaluation Results

- The evaluation results indicate a lower than 3% error rate.

– Precision is highlighted as a key metric for assessing the performance of the proposed model.

Model Comparison

– RMR-CNN's performance is compared with conventional deep neural network architectures such as Fast R-CNN and Mask R-CNN.

Precision Metric

– The proposed RMR-CNN model achieved a precision of 97.08%.

– This precision is reported to be higher than the precision obtained by Mask R-CNN and Faster R-CNN models.

Dataset Details

– The evaluation is conducted on an innovative dataset consisting of 50,000 images with 10,000 instances of German traffic signs grouped into 43 categories.

6.3 Feature Importance

6.3.1 Importance for Road Traffic Sign Classification and Recognition

Color Information

1. Different traffic signs often have distinct colors. The model may assign higher importance to color features, helping distinguish between, for example, a red stop sign and a blue information sign.

Shape and Symbol Recognition

1. The shapes and symbols present on the traffic signs are crucial for classification. Feature importance might be attributed to the recognition of specific shapes or symbols associated with different sign categories.

Size and Aspect Ratio

1. The size and aspect ratio of the detected object in the image can provide important cues for classification. Some signs may be characterized by specific dimensions, and the model might learn to recognize these patterns.

Texture and Pattern Recognition

1. Texture and patterns within the sign, such as stripes or patterns on a warning sign, can contribute to accurate classification. Feature importance may be assigned to the detection.

Texture-Based Characteristics

1. Spatial Arrangement: The relative spatial arrangement of different elements within the sign could be informative. For instance, the location of symbols or text within the sign may be considered important for classification.
2. Contextual Information: Contextual features, such as the surrounding environment or the presence of other road elements, could influence the classification. The model might assign importance to understanding the context in which the sign appears.
3. Deep Learning Representations: In deep learning models, especially convolutional neural networks (CNNs), lower layers might automatically learn to extract features like edges and colors, while higher layers focus on more abstract features. Understanding the learned representations can provide insights into feature importance.

4. Traffic Sign Specific Features: Features specifically designed to capture unique characteristics of traffic signs, such as specific edge patterns or characteristic shapes, may be deemed important for classification.

6.4 Interpretation of Findings

6.4.1 K-Nearest Neighbors (k-NN)

Table 6.1: Overview of K-Nearest Neighbors (k-NN)

Type	Supervised Learning (can be used for both classification and regression)
Algorithm	Instance-based, non-parametric
Principle	Classifies a data point based on how its neighbors are classified.
Distance Metric	Typically Euclidean distance, but other distance metrics can be used.

Table 6.2: How K-Nearest Neighbors (k-NN) Works

Training	The algorithm stores all training examples.
Prediction (Classification)	For a given data point, it identifies the k-nearest data points in the training set.
Voting (Classification)	Classifies the data point by majority class among its k-nearest neighbors.
Prediction (Regression)	For regression, it may take the average of the values of k-nearest neighbors.

Table 6.3: Parameters and Key Considerations for K-Nearest Neighbors (k-NN)

Parameters	**k:** Number of neighbors to consider. It's a hyperparameter. **Distance Metric:** Defines the distance between data points.
Key Considerations	**Choice of k:** Small values of k lead to more flexible models but might be sensitive to noise. Larger k makes the model more robust but less sensitive to local patterns. **Impact of Distance Metric:** Different distance metrics can have a significant impact on the results.

6.5 Results

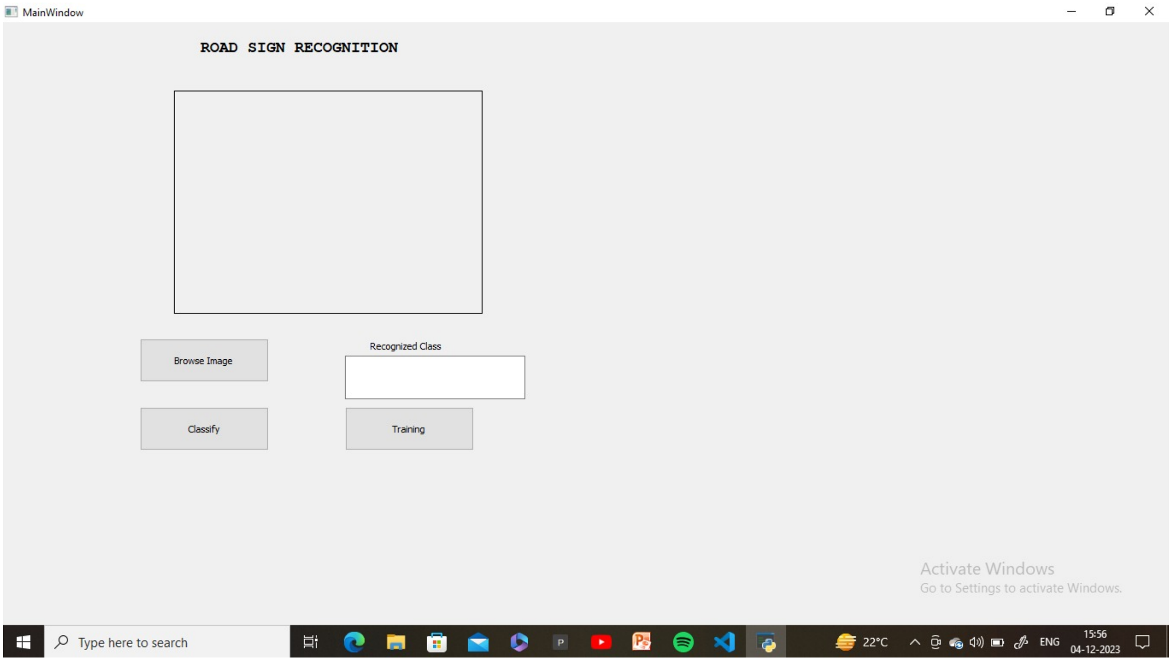


Figure 6.2: Initial picture

Road Traffic Sign Classification Using Neural Network

- Code for model building used in our project:-

```
from PyQt5 import QtCore, QtGui, QtWidgets
from keras.models import Sequential, load_model
from keras.layers import Conv2D, MaxPool2D, Dense, Flatten, Dropout
from keras.utils import to_categorical
from sklearn.model_selection import train_test_split
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
import os
import tensorflow as tf  # Import TensorFlow

# Suppress TensorFlow warnings
tf.compat.v1.logging.set_verbosity(tf.compat.v1.logging.ERROR)

data = []
labels = []
classes = 43
cur_path = os.getcwd()  # To get the current directory

classs = {1: "Speed limit (20km/h)",
2: "Speed limit (30km/h)",
3: "Speed limit (50km/h)",
4:"Speed limit (60km/h)",
5:"Speed limit (70km/h)",
```

6:"Speed limit (80km/h)",

7:"End of speed limit (80km/h)",

8:"Speed limit (100km/h)",

9:"Speed limit (120km/h)",

10:"No passing",

11:"No passing veh over 3.5 tons",

12:"Right-of-way at intersection",

13:"Priority road",

14:"Yield",

15:"Stop",

16:"No vehicles",

17:"Veh > 3.5 tons prohibited",

18:"No entry",

19:"General caution",

20:"Dangerous curve left",

21:"Dangerous curve right",

22:"Double curve",

23:"Bumpy road",

24:"Slippery road",

25:"Road narrows on the right",

26:"Road work",

27:"Traffic signals",

28:"Pedestrians",

29:"Children crossing",

30:"Bicycles crossing",

31:"Beware of ice/snow",

32:"Wild animals crossing",

33:"End speed + passing limits",

34:"Turn right ahead",

35:"Turn left ahead",

36:"Ahead only",

```
37:"Go straight or right",

38:"Go straight or left",

39:"Keep right",

40:"Keep left",

41:"Roundabout mandatory",

42:"End of no passing",

43: "End no passing veh > 3.5 tons"}


# Retrieving the images and their labels

print("Obtaining Images & its Labels.............")

for i in range(classes):

path = os.path.join(cur_path,"D:\Road sign recognization by using deeep learning\dataset\Train", str(i))

images = os.listdir(path)


for a in images:

try:

image = Image.open(os.path.join(path, a))

image = image.resize((30, 30))

image = np.array(image)

data.append(image)

labels.append(i)

print("{0} Loaded".format(a))

except:

print("Error loading image")

print("Dataset Loaded")


# Converting lists into numpy arrays

data = np.array(data)

labels = np.array(labels)
```

```
print(data.shape, labels.shape)

# Splitting training and testing dataset

X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.2,
random_state=42)


print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)


# Converting the labels into one hot encoding

y_train = to_categorical(y_train, 43)

y_test = to_categorical(y_test, 43)



class Ui_MainWindow(object):

def setupUi(self, MainWindow):

MainWindow.setObjectName("MainWindow")

MainWindow.resize(800, 600)

self.centralwidget = QtWidgets.QWidget(MainWindow)

self.centralwidget.setObjectName("centralwidget")

self.BrowseImage = QtWidgets.QPushButton(self.centralwidget)

self.BrowseImage.setGeometry(QtCore.QRect(160, 370, 151, 51))

self.BrowseImage.setObjectName("BrowseImage")

self.imageLbl = QtWidgets.QLabel(self.centralwidget)

self.imageLbl.setGeometry(QtCore.QRect(200, 80, 361, 261))

self.imageLbl.setFrameShape(QtWidgets.QFrame.Box)

self.imageLbl.setText("")

self.imageLbl.setObjectName("imageLbl")

self.label_2 = QtWidgets.QLabel(self.centralwidget)

self.label_2.setGeometry(QtCore.QRect(110, 20, 621, 20))

font = QtGui.QFont()

font.setFamily("Courier New")

font.setPointSize(14)
```

```
font.setBold(True)
font.setWeight(75)
self.label_2.setFont(font)
self.label_2.setObjectName("label_2")
self.Classify = QtWidgets.QPushButton(self.centralwidget)
self.Classify.setGeometry(QtCore.QRect(160, 450, 151, 51))
self.Classify.setObjectName("Classify")
self.label = QtWidgets.QLabel(self.centralwidget)
self.label.setGeometry(QtCore.QRect(430, 370, 111, 16))
self.label.setObjectName("label")
self.Training = QtWidgets.QPushButton(self.centralwidget)
self.Training.setGeometry(QtCore.QRect(400, 450, 151, 51))
self.Training.setObjectName("Training")
self.textEdit = QtWidgets.QTextEdit(self.centralwidget)
self.textEdit.setGeometry(QtCore.QRect(400, 390, 211, 51))
self.textEdit.setObjectName("textEdit")
MainWindow.setCentralWidget(self.centralwidget)
self.menubar = QtWidgets.QMenuBar(MainWindow)
self.menubar.setGeometry(QtCore.QRect(0, 0, 800, 26))
self.menubar.setObjectName("menubar")
MainWindow.setMenuBar(self.menubar)
self.statusbar = QtWidgets.QStatusBar(MainWindow)
self.statusbar.setObjectName("statusbar")
MainWindow.setStatusBar(self.statusbar)

self.retranslateUi(MainWindow)
QtCore.QMetaObject.connectSlotsByName(MainWindow)

self.BrowseImage.clicked.connect(self.loadImage)
self.Classify.clicked.connect(self.classifyFunction)
self.Training.clicked.connect(self.trainingFunction)
```

```
def retranslateUi(self, MainWindow):

_translate = QtCore.QCoreApplication.translate

MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))

self.BrowseImage.setText(_translate("MainWindow", "Browse Image"))

self.label_2.setText(_translate("MainWindow", "          ROAD SIGN RECOGNITION"))

self.Classify.setText(_translate("MainWindow", "Classify"))

self.label.setText(_translate("MainWindow", "Recognized Class"))

self.Training.setText(_translate("MainWindow", "Training"))


def loadImage(self):

fileName, _ = QtWidgets.QFileDialog.getOpenFileName(None, "Select Image", "",

"Image Files (*.png *.jpg *jpeg *.bmp);;All Files (*)")  # Ask for file

if fileName:  # If the user gives a file

print(fileName)

self.file = fileName

pixmap = QtGui.QPixmap(fileName)  # Setup pixmap with the provided image

pixmap = pixmap.scaled(self.imageLbl.width(), self.imageLbl.height(),
QtCore.Qt.KeepAspectRatio)  # Scale pixmap

self.imageLbl.setPixmap(pixmap)  # Set the pixmap onto the label

self.imageLbl.setAlignment(QtCore.Qt.AlignCenter)  # Align the label to center


def classifyFunction(self):

model = load_model('D:\Road sign recognization by using deeep
learning\my_model.h5')

print("Loaded model from disk")

path2 = self.file

print(path2)

test_image = Image.open(path2)

test_image = test_image.resize((30, 30))

test_image = np.expand_dims(test_image, axis=0)

test_image = np.array(test_image)
```

```
result = np.argmax(model.predict(test_image), axis=-1)[0]

sign = classs[result + 1]

print(sign)

self.textEdit.setText(sign)


def trainingFunction(self):

self.textEdit.setText("Training under process...")

model = Sequential()

model.add(Conv2D(filters=32, kernel_size=(5, 5), activation='relu',
input_shape=X_train.shape[1:]))

model.add(Conv2D(filters=32, kernel_size=(5, 5), activation='relu'))

model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Dropout(rate=0.25))

model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))

model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))

model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Dropout(rate=0.25))

model.add(Flatten())

model.add(Dense(256, activation='relu'))

model.add(Dropout(rate=0.5))

model.add(Dense(43, activation='softmax'))

print("Initialized model")


# Compilation of the model

model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])


history = model.fit(X_train, y_train, batch_size=32, epochs=5,
validation_data=(X_test, y_test))

model.save("my_model.h5")
```

```
plt.figure(0)

plt.plot(history.history['accuracy'], label='training accuracy')

plt.plot(history.history['val_accuracy'], label='val accuracy')

plt.title('Accuracy')

plt.xlabel('epochs')

plt.ylabel('accuracy')

plt.legend()

plt.savefig('Accuracy.png')


plt.figure(1)

plt.plot(history.history['loss'], label='training loss')

plt.plot(history.history['val_loss'], label='val loss')

plt.title('Loss')

plt.xlabel('epochs')

plt.ylabel('loss')

plt.legend()

plt.savefig('Loss.png')

self.textEdit.setText("Saved Model & Graph to disk")



if __name__ == "__main__":

import sys


app = QtWidgets.QApplication(sys.argv)

MainWindow = QtWidgets.QMainWindow()

ui = Ui_MainWindow()

ui.setupUi(MainWindow)

MainWindow.show()

sys.exit(app.exec_())
```

6.5.1 Output Interface

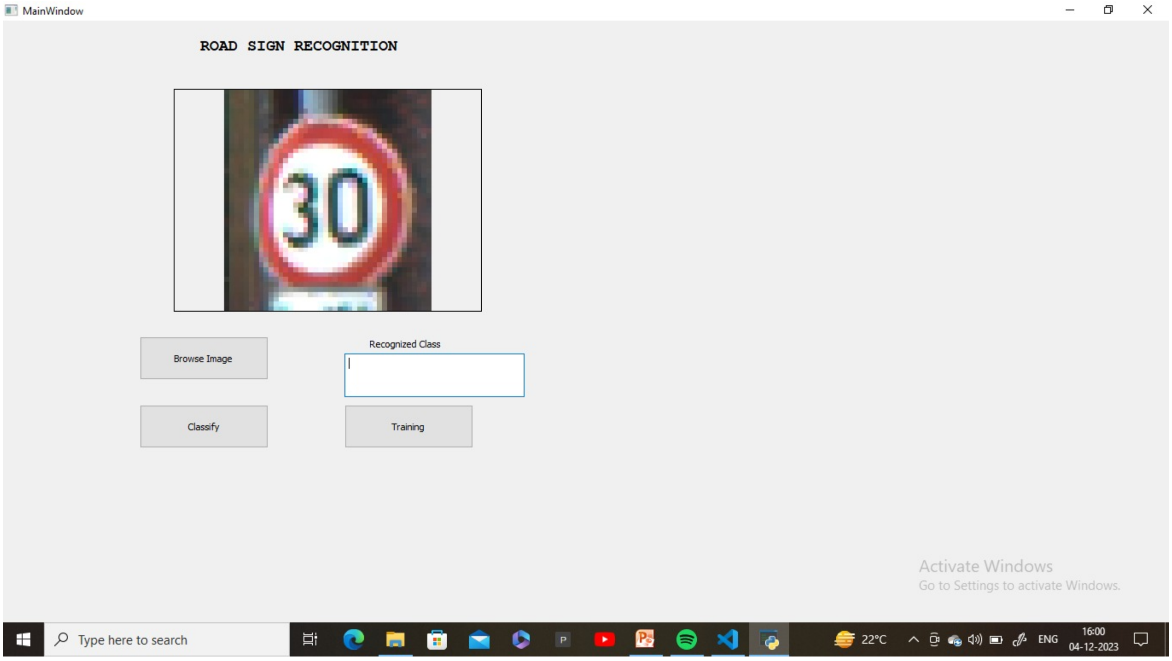


Figure 6.3: Browsing Images

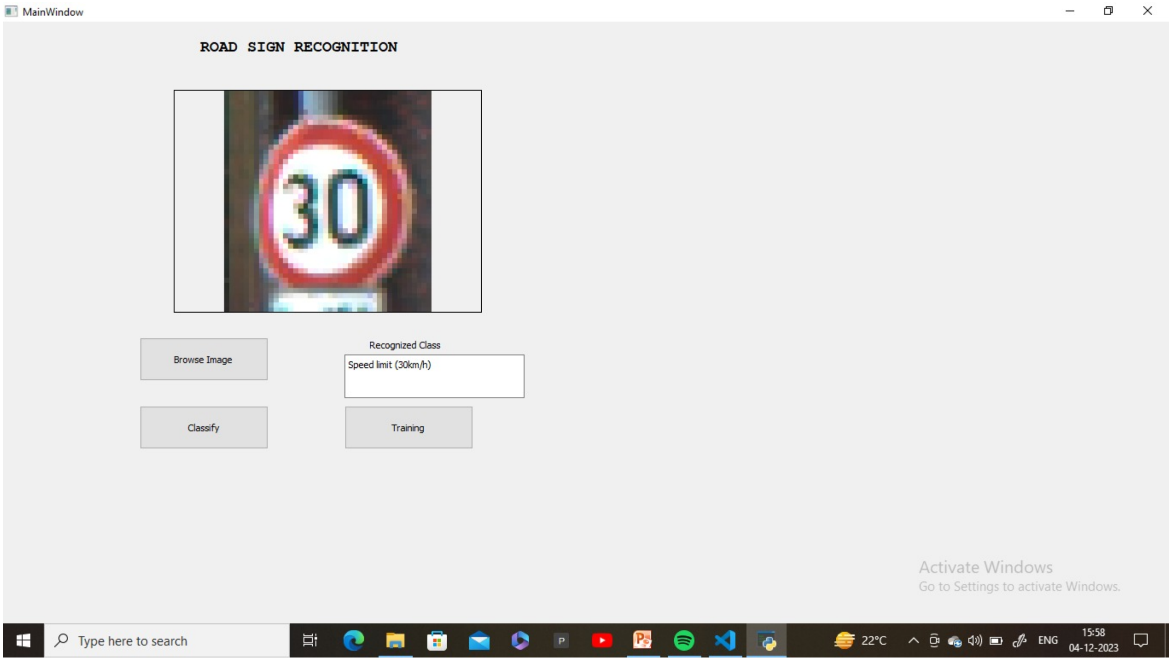


Figure 6.4: Classify Images

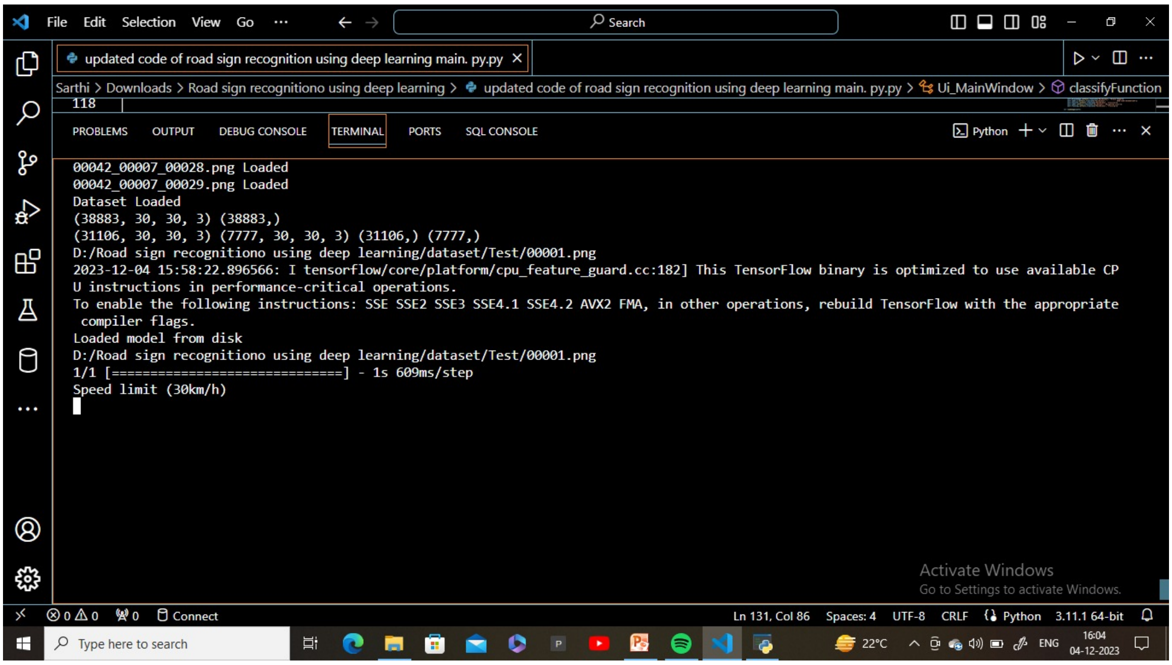


Figure 6.5: Code output Image 01

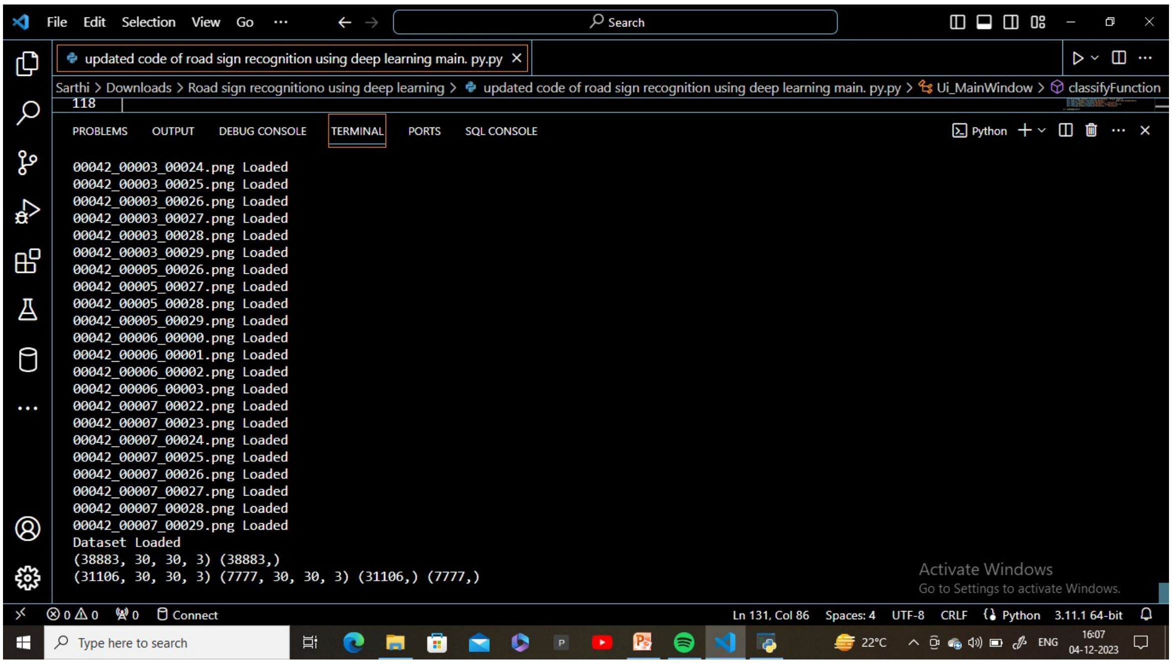


Figure 6.6: Code output Image 02

Chapter 7

Conclusion

7.1 Conclusions:

1. The proposed traffic sign classification system using Convolutional Neural Networks (CNN) is simple, highly accurate, and offers real-time capabilities.
2. It significantly contributes to driver safety, convenience, and the development of smarter cars in automatic driving assistance systems.

7.2 Summary of Key Findings:

1. Impressive accuracy of 93% is achieved.
2. Real-time webcam detection provides quick results, enhancing driver convenience.
3. Minimal hardware requirements highlight practical applicability.
4. Potential applications in building smarter cars, especially in future automated driving vehicles.

7.3 Contributions:

1. Introduction of a high-accuracy traffic sign classification system with a CNN.
2. Real-time webcam detection, model simplicity, and minimal hardware requirements contribute practically to automatic driving assistance technologies.

7.4 Implications:

1. The system holds significant potential for enhancing driver safety, convenience, and the development of smarter cars in automatic driving assistance systems.

7.5 Suggestions:

1. Future research on obstructed sign detection.
2. Improvement in system performance for high-speed scenarios.
3. Consideration of environmental conditions for enhanced system robustness.

7.6 Future Research:

1. Ongoing research to address limitations and improve robustness in real-world conditions.

Chapter 8

Bibliography

1. https://doi.org/10.1109/TITS.2015.2496795Huang, Z., Yu, Y., Gu, J., Liu, H., 2017. An efficient method for traffic sign recognition based on extreme learning machine. IEEE Trans. Cybern. 47 (4), 920â933.

2. https://doi.org/10.1109/TITS.2018.2801560Lin, T.-Y., DollĂAr, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. Proc. Comput. Vis. Pattern Recognit., 936â944

3. https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-signKaggle - German Traffic Sign Recognition Benchmark

4. https://en.wikipedia.org/wiki/Intelligent$_{t}ransportation_{s}ystem$ $IntelligentTransportationSystemonWikipedia$

5. https://en.wikipedia.org/wiki/Advanced$_{d}river - assistance_{s}ystems$ $AdvancedDriverAssistanceSystemsonWikipedia$