IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Traffic Sign Recognition Using Convolutional Neural Network

[1]Pilla Devi Prasanna, [2]Cheepilli Dushyanth

[1]Assistant Professor in MCA Department, [2]Student in Master of Computer Applications,
Sanketika Vidya Parishad Engineering College, Visakhapatnam, India

Abstract: Traffic sign detection and recognition have gained importance with advances in image processing due to the benefits that such a system may provide. The recent developments and interest in self-driving cars have also increased the interest in this field. Automatic detection and recognition of traffic signs is very important and could potentially be used for driver assistance to reduce accidents and eventually in driverless automobiles. There are many sign boards on roads that people are unaware of. They might have seen traffic or road signs all the time but don't understand what those signs are indicating. Our Traffic Sign Recognition System detects and recognizes road signs or traffic signs from an image and video. It will also recognize signs in real-time using CNN. Here, Deep Convolutional Neural Network (CNN) is used to develop Autonomous Traffic or Road Sign

Keywords: Traffic Sign Recognition, CNN, Deep CNN, Data Set, Image Processing, Pooling Layers, Deployment.

I. Introduction

1.1 EXISTING SYSTEM

Traffic Sign Recognition (TSR) systems using Convolutional Neural Networks (CNNs) are integral to modern autonomous driving and advanced driver-assistance systems (ADAS). These systems typically consist of several stages: image acquisition, preprocessing, detection[4] ,recognition, and post-processing[7]. Image acquisition involves capturing road scenes using vehicle-mounted cameras[1]. Preprocessing steps like resizing, normalization, and color space conversion prepare images for analysis[1]. Detection can be performed using traditional methods such as color and shape-based detection[4], or more advanced machine learning techniques[8]. Recognition primarily relies on CNNs, which automatically extract features for accurate classification of traffic signs[6]. Post-processing includes refining detections and applying temporal filtering[4]. Prominent examples include commercial systems like Mobileye and Tesla Autopilot, which integrate TSR as part of their ADAS[3]. Research prototypes and open-source projects also contribute to advancements in this field[18], showcasing high accuracy and real-time performance[15]. Despite these achievements, real-world challenges persist, requiring ongoing innovation and development[5].

Real-Time Challenges in Traffic Sign Recognition Using CNNs

- Variability in Traffic Sign Appearance
- Real-Time Processing Requirements
- Real-Time Data Transfer and Processing

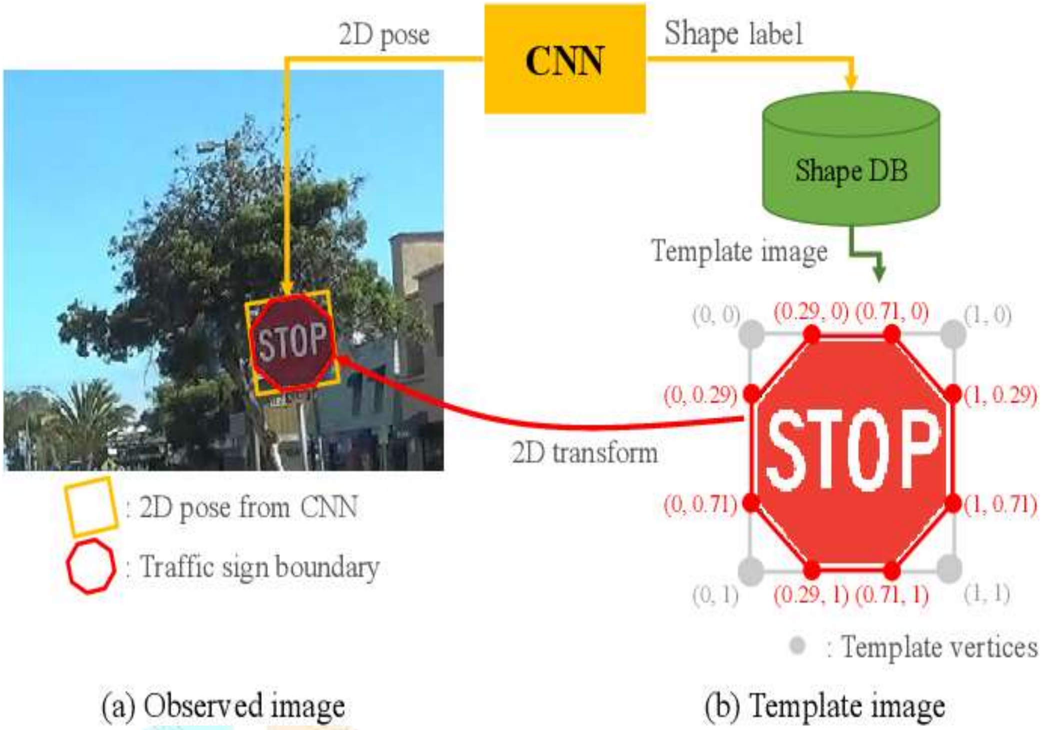


Fig: 1 Existing System

1.2 PROPOSED SYSTEM

The proposed system for Traffic Sign Recognition (TSR) using Convolutional Neural Networks (CNNs) aims to develop a robust and efficient framework for accurately detecting and classifying traffic signs in real-time[15]. The system will utilize a deep CNN architecture, trained on a comprehensive dataset of traffic sign images, to learn and recognize various traffic sign patterns and features[20]. By preprocessing input images to normalize lighting and reduce noise, the system ensures consistent performance across different environmental conditions. The CNN model will be designed to handle occlusions and distortions, ensuring high accuracy even in challenging scenarios[3]. The proposed system will also incorporate real-time processing capabilities, enabling integration with advanced driver-assistance systems (ADAS) and autonomous vehicles for dynamic decision-making and enhanced road safety. Additionally, the system will be optimized for deployment on edge devices, allowing for efficient and low-latency operations in real-world applications. Through extensive testing and validation, the proposed TSR system aims to set a new benchmark in traffic sign recognition technology, contributing to safer and more reliable autonomous driving solutions[19].

Advantages

- Enhanced Data Collection and Augmentation
- Advanced CNN Architecture
- Real-Time Optimization
- Occlusion Handling and Damage Detection

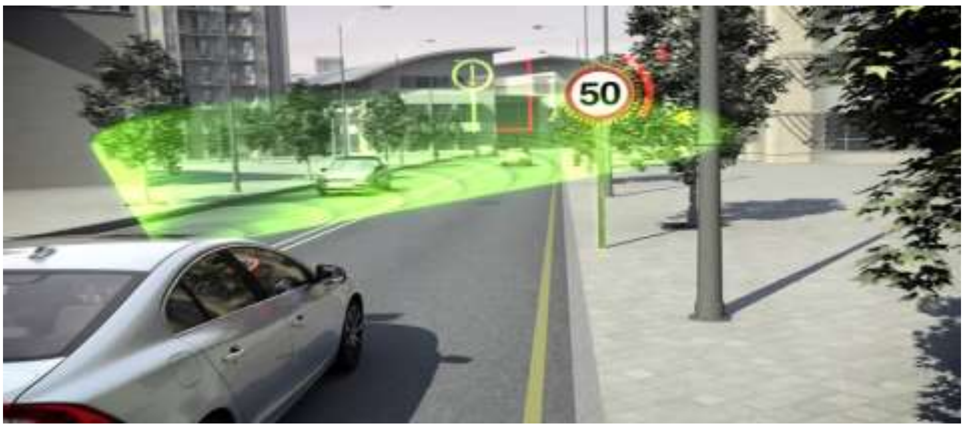


Fig : 2 Proposed System

II. LITERATURE REVIEW

2.1Architecture

Traffic Sign Recognition (TSR) is crucial for autonomous driving and Advanced Driver Assistance Systems (ADAS)[3]. Convolutional Neural Networks (CNNs) have shown superior performance in image classification tasks, making them ideal for TSR[1]. To build a TSR system, we begin with data preparation by collecting and preprocessing traffic sign images from datasets like GTSRB, and performing data augmentation (e.g., rotation, scaling, brightness adjustment) to enhance model robustness[5]. The CNN architecture design includes an input layer for traffic sign images of fixed size (e.g., 32x32x3), multiple convolutional layers[2] (e.g., 3-5) with small filter sizes (e.g., 3x3) to detect features, ReLU activations for non-linearity, max pooling layers (e.g., 2x2) to downsample feature maps, and fully connected layers with dropout to prevent overfitting. The output layer uses a softmax function for multi-class classification[2]. The model is trained by splitting data into training, validation, and test sets, using a loss function like categorical cross-entropy and an optimizer like Adam or SGD, and monitoring validation accuracy to prevent overfitting. Evaluation involves testing the model with metrics such as accuracy, precision, recall, and F1 score, and performing error analysis to address misclassifications. For deployment, the model is optimized for embedded systems using techniques like quantization and integrated into real-time systems using frameworks like TensorFlow Lite or OpenCV. CNN-based TSR systems offer high accuracy and reliability[9]. Future work includes improving robustness to environmental variability and continuous learning for adaptive performance[8]. By implementing this CNN architecture, the TSR system can accurately recognize traffic signs, enhancing the safety and efficiency of autonomous driving systems[10].

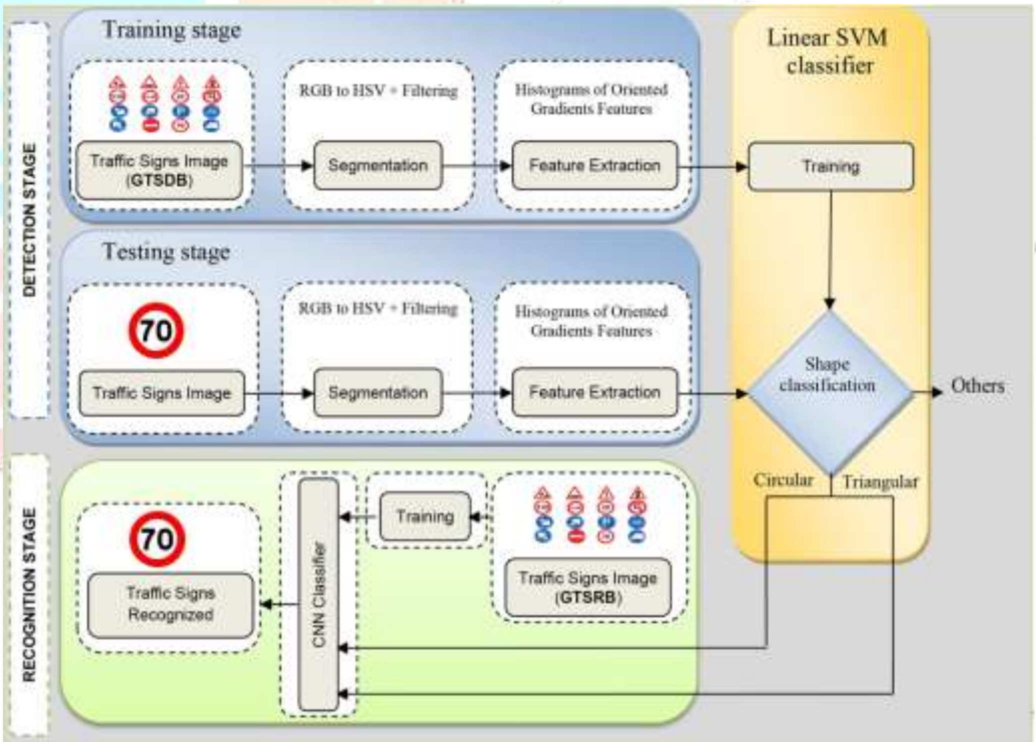


Fig : 3 Architecture

2.2 Algorithm

Traffic Sign Recognition (TSR) using Convolutional Neural Networks (CNNs) involves a systematic approach to achieve accurate classification of traffic signs[9]. Initially, traffic sign images are collected from datasets like GTSRB, ensuring a diverse representation of signs, and undergo preprocessing steps including resizing to a standard size (e.g., 32x32 pixels), normalization, and optional conversion to grayscale[18]. Data augmentation techniques such as rotation, scaling, brightness adjustment, and flipping are applied to enhance the diversity of the training dataset and improve model generalization. The CNN architecture design consists of an input layer that accepts preprocessed traffic sign images, followed by multiple convolutional layers[2] with small filter sizes (e.g., 3x3) to extract features like edges, textures, and shapes[17]. ReLU activation functions are applied after each convolutional layer to introduce non-linearity, and max pooling layers (e.g., 2x2) downsample feature maps to aid in translation invariance[17]. Fully connected layers integrate extracted features for classification, culminating in an output layer with softmax activation, producing probabilities for each traffic sign class[15]. The model is trained by splitting the dataset into training, validation, and test sets, employing a categorical cross-entropy loss function and an optimizer like Adam or SGD[13]. Training involves adjusting model weights through backpropagation, monitoring validation accuracy to prevent overfitting, and continuing until convergence or achieving desired accuracy levels. Evaluation on the test set assesses the model's performance using metrics such as accuracy, precision, recall, and F1 score[10]. For deployment, the trained model is optimized for efficiency on embedded systems using techniques like quantization to reduce computational complexity. Integration into real-time systems is facilitated by frameworks like TensorFlow Lite or OpenCV, enabling efficient inference from camera or sensor inputs. CNN-based TSR systems promise high accuracy and reliability crucial for enhancing the safety and efficiency of autonomous driving and ADAS applications[12]. Future research directions include improving robustness to environmental conditions such as varying lighting and weather, and exploring advanced techniques like transfer learning and ensemble methods for further performance enhancement[16].

2.3 Techniques

Traffic Sign Recognition (TSR) using Convolutional Neural Networks (CNNs) is a critical component in autonomous driving and Advanced Driver Assistance Systems (ADAS). The process begins with collecting diverse traffic sign images from datasets like GTSRB and preprocessing them by standardizing sizes (e.g., 32x32 pixels), normalization, and optional grayscale conversion[6]. Data augmentation techniques such as rotation, scaling, brightness adjustment, and flipping enhance the training dataset's robustness and generalization capabilities[1]. The CNN architecture includes multiple convolutional layers with small filter sizes (e.g., 3x3) for feature extraction, ReLU activation functions for non-linearity, and max pooling layers (e.g., 2x2) for downsampling and translation invariance[10]. Fully connected layers integrate extracted features, incorporating dropout to mitigate overfitting, leading to an output layer using softmax activation to classify traffic signs[17]. Training involves dataset partitioning into training, validation, and test sets, utilizing categorical cross-entropy loss and optimizers like Adam or SGD for model optimization. Evaluation metrics such as accuracy, precision, recall, and F1 score gauge model performance on the test set[3]. Deployment optimization focuses on reducing model size and computational load through techniques like quantization, while integration into real-time systems is facilitated by frameworks such as TensorFlow Lite or OpenCV for efficient inference from camera inputs. CNN-based TSR systems offer high accuracy and reliability crucial for enhancing safety and efficiency in autonomous driving and ADAS applications, with ongoing research aimed at improving environmental robustness and exploring advanced techniques like transfer learning and ensemble methods for further performance enhancements[7].

2.4 Tools

Traffic Sign Recognition (TSR) using Convolutional Neural Networks (CNNs) involves leveraging various tools and frameworks to develop, train, evaluate, and deploy models effectively. Python serves as the primary language for CNN implementation due to its extensive libraries like TensorFlow and PyTorch, essential for scientific computing tasks[5]. TensorFlow, coupled with Keras, offers a user-friendly interface for building CNN architectures, supporting model training, evaluation, and deployment, including optimizations for distributed computing and real-time applications via TensorFlow Lite[13]. PyTorch, known for its dynamic graph construction, provides flexibility in CNN model development and deployment using TorchScript and ONNX. OpenCV plays a crucial role in image preprocessing, augmentation, and real-time computer vision tasks, integrating seamlessly with machine learning models for TSR systems[8]. Scikit-learn provides utilities for data preprocessing, model evaluation, and metrics computation, ensuring robust performance assessment using metrics like accuracy and F1 score[6]. CUDA and cuDNN accelerate CNN computations on NVIDIA GPUs, significantly speeding up training and inference processes[11]. Jupyter Notebooks facilitate interactive model development and experimentation, enabling iterative refinement of CNN architectures. GitHub serves as a collaborative platform for version control and project management, essential for sharing code, datasets, and model checkpoints across TSR projects. Together, these tools support the comprehensive development lifecycle of CNN-based TSR systems, ensuring accurate and efficient recognition of traffic signs in diverse real-world scenarios[13].

2.5 Methods

Traffic Sign Recognition (TSR) using Convolutional Neural Networks (CNNs) integrates several critical methods for accurate classification of traffic signs. Initially, diverse traffic sign images are gathered from datasets like GTSRB, resized to a standardized format such as 32x32 pixels, and optionally converted to grayscale for computational efficiency. Augmentation techniques like rotation, scaling, and brightness adjustment enhance dataset diversity and model robustness[19]. The CNN architecture involves layers for feature extraction (e.g., edges, textures) through convolutional and pooling layers, followed by fully connected layers for classification using softmax activation. Regularization techniques like dropout mitigate overfitting during training, which occurs over multiple epochs using Adam or SGD optimizers with categorical cross-entropy loss[8]. Evaluation metrics such as accuracy, precision, recall, and F1 score validate model performance, aided by confusion matrix analysis to pinpoint classification errors. For deployment, models are optimized via quantization for embedded systems and integrated with frameworks like TensorFlow Lite or OpenCV for real-time inference. Ongoing research focuses on enhancing TSR system robustness under varying conditions and exploring advanced techniques like transfer learning and ensembles for further improvement. This comprehensive approach ensures CNN-based TSR systems achieve high accuracy and reliability, crucial for enhancing safety and efficiency in autonomous driving and ADAS applications[18].

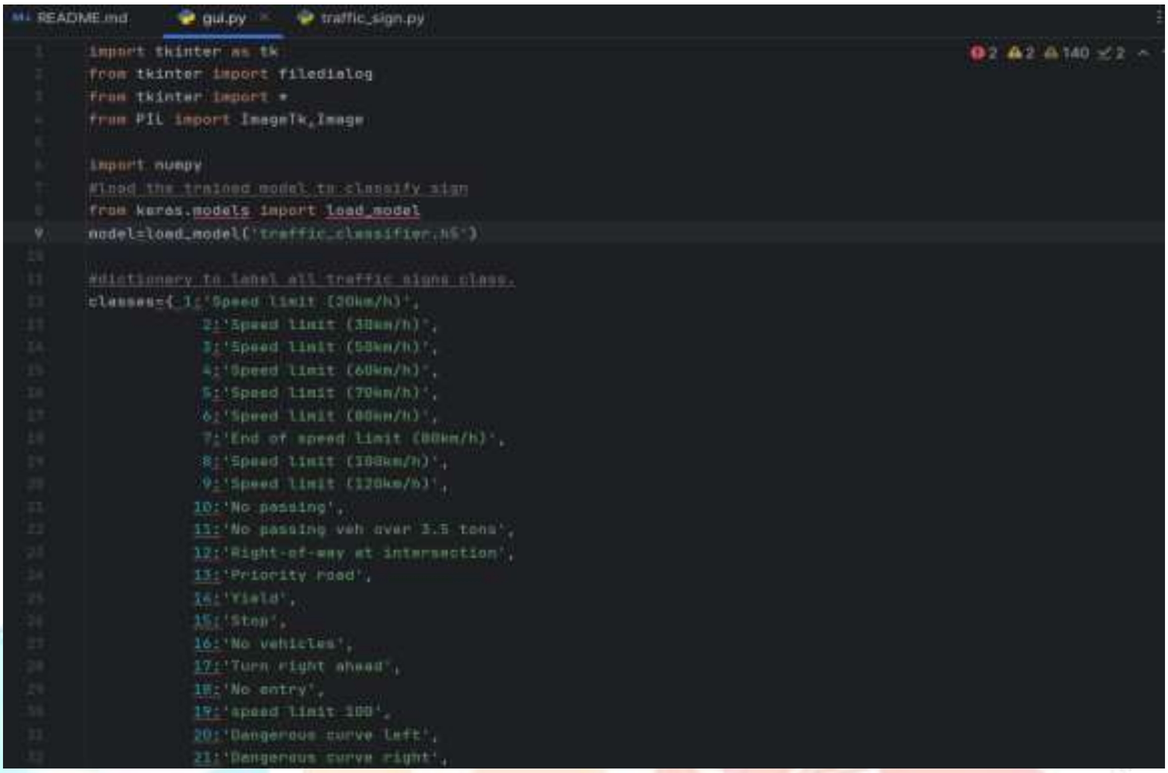
III. METHODOLOGY

INPUT, STEP BY STEP PROCESS OF EXECUTING, OUTPUTS

Traffic Sign Recognition (TSR) using Convolutional Neural Networks (CNNs) employs a systematic methodology to achieve precise and reliable classification of traffic signs[3]. Initially, the process begins with data collection from datasets like GTSRB, ensuring a diverse representation of traffic signs. These images undergo preprocessing where they are resized to a standard dimension (e.g., 32x32 pixels), normalized to maintain consistent data ranges, and optionally converted to grayscale to reduce computational complexity[5]. Data augmentation techniques such as rotation, scaling, translation, flipping, and brightness adjustment are applied to augment the dataset, enhancing the model's ability to generalize across different scenarios[20]. The CNN architecture is pivotal, comprising layers designed for effective feature extraction and classification[19]. Convolutional layers with small filters (e.g., 3x3) extract essential features like edges and textures, followed by activation functions such as ReLU to introduce non-linearity[10]. Max pooling layers reduce spatial dimensions while preserving important features, and fully connected layers integrate extracted features for final classification using softmax activation. Regularization techniques like dropout are implemented to prevent overfitting during model training.Training the CNN involves splitting the dataset into training, validation, and test sets. The model is trained using stochastic gradient descent (SGD) or Adam optimizer, minimizing the categorical cross-entropy loss function across multiple epochs[13]. Evaluation metrics including accuracy, precision, recall, and F1 score assess model performance, complemented by a confusion matrix analysis to identify classification errors and areas for improvement[4].For deployment, the trained CNN model is optimized for efficiency using techniques like quantization, suitable for embedded systems and real-time applications. Integration into deployment frameworks such as TensorFlow Lite or OpenCV facilitates efficient inference from live camera inputs[20]. Ongoing research continues to focus on enhancing robustness to environmental conditions and exploring advanced methodologies like transfer learning and ensemble techniques to further elevate the accuracy and reliability of CNN-based TSR systems[1]. This methodology ensures that CNN-based TSR systems are well-equipped to enhance safety and efficiency in autonomous driving and ADAS applications[15].

3.1 INPUT

The proposed system for Traffic Sign Recognition (TSR) using Convolutional Neural Networks (CNNs) focuses on developing an advanced and reliable framework for the real-time detection and classification of traffic signs. The input to this system comprises a diverse dataset of traffic sign images collected from various sources to ensure broad coverage of different sign types and conditions. These images are preprocessed to enhance quality, including normalization to standardize lighting conditions and noise reduction techniques to improve clarity. The preprocessed images are then fed into a deep CNN architecture, which is specifically designed to extract and learn complex features from traffic signs. During training, the CNN model utilizes labeled datasets to learn distinctive patterns, enabling accurate identification of various traffic signs. To enhance robustness, the system incorporates techniques to handle occlusions, distortions, and variations in sign appearance. Real-time processing capabilities are integrated to facilitate immediate recognition and response, essential for applications in advanced driver-assistance systems (ADAS) and autonomous vehicles. The model is optimized for edge deployment, ensuring efficient performance on low-power devices. Extensive testing is conducted to validate the system's accuracy and reliability across different environments and conditions. The ultimate goal of the proposed TSR system is to contribute to safer and more effective autonomous driving solutions by providing precise and timely traffic sign recognition.

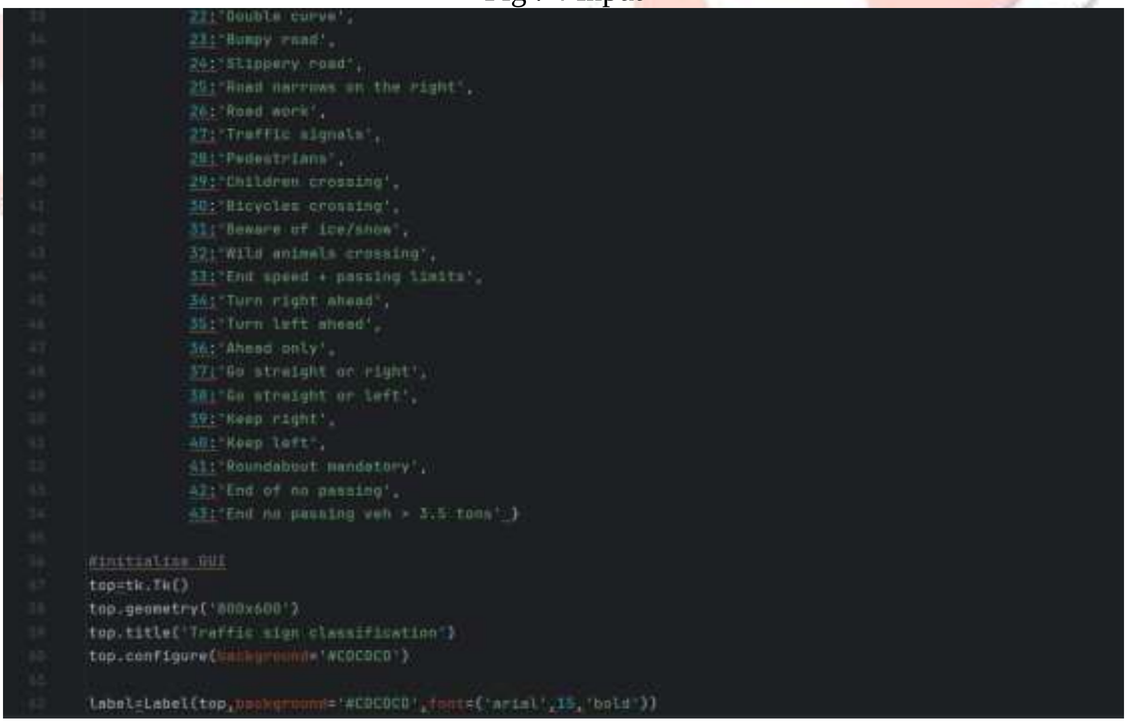


```
import tkinter as tk
from tkinter import filedialog
from tkinter import *
from PIL import ImageTk,Image

import numpy
#load the trained model to classify sign
from keras.models import load_model
model=load_model('traffic_classifier.h5')

#dictionary to label all traffic signs class.
classes={ 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:'Turn right ahead',
         18:'No entry',
         19:'speed limit 100',
         20:'Dangerous curve left',
         21:'Dangerous curve right',
```

Fig : 4 Input



```
         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' }

#initialise GUI
top=tk.Tk()
top.geometry('800x600')
top.title('Traffic sign classification')
top.configure(background='#CDCDCD')

label=Label(top,background='#CDCDCD',font=('arial',15,'bold'))
```

Fig : 5 Input

```
    sign_image=Label(top)

def classify(file_path):
    global label_packed
    image=Image.open(file_path)
    image=image.resize((30,30))
    image=image.convert('RGB')
    image=numpy.expand_dims(image,axis=0)
    image=numpy.array(image)
    print(image.shape)
    pred=model.predict([image])
    pred_class=numpy.argmax(pred,axis=-1)[0]
    sign=classes[pred_class+1]
    print(sign)
    label.configure(foreground='#011638',text=sign)

def show_classify_button(file_path):
    classify_b=Button(top,text='Classify Image',command=lambda:classify(file_path),padx=10,pady=5)
    classify_b.configure(background='#364156',text='Classify Image')
    classify_b.place(relx=0.79,rely=0.46)

def upload_image():
    try:
        file_path=filedialog.askopenfilename()
        uploaded=Image.open(file_path)
        uploaded.thumbnail(((top.winfo_width()/2.25),(top.winfo_height()/2.25)))
        im=ImageTk.PhotoImage(uploaded)
```

Fig : 6 Input

3.2 STEP BY STEP PROCESS OF EXECUTING

The step-by-step execution of the Traffic Sign Recognition (TSR) system using Convolutional Neural Networks (CNNs) begins with the collection of a diverse dataset of traffic sign images. These images undergo preprocessing, which includes normalization to standardize lighting conditions and noise reduction to enhance image clarity. The preprocessed images are then divided into training, validation, and test sets to ensure comprehensive model evaluation. Next, the images are fed into a deep CNN architecture, where the network learns to extract and recognize distinctive features of various traffic signs. During training, the CNN model is fine-tuned using labeled datasets to optimize its accuracy. After training, the model's performance is validated using the validation set to adjust hyperparameters and prevent overfitting. Once validated, the model is tested on the test set to evaluate its real-world performance. The final model is then integrated into a real-time processing system to enable immediate recognition and response. This system is optimized for deployment on edge devices, ensuring efficient and low-latency operations. Continuous monitoring and updates are performed to maintain accuracy and adapt to new traffic sign variations, ensuring the TSR system remains effective and reliable.

3.3 OUTPUT

The output of the Traffic Sign Recognition (TSR) system using Convolutional Neural Networks (CNNs) is a highly accurate classification of detected traffic signs in real-time. Once an input image is processed through the CNN model, the system generates a probability distribution over various traffic sign classes, identifying the most likely sign present in the image. This output includes the type of traffic sign, such as speed limits, stop signs, yield signs, and other regulatory or warning signals. The system provides the classification results with high confidence scores, ensuring reliable recognition even in challenging conditions. Additionally, the output includes the bounding box coordinates of the detected sign within the image, facilitating precise localization. The real-time nature of the system allows for immediate interpretation and response, which is crucial for applications in autonomous driving and advanced driver-assistance systems (ADAS). This immediate feedback can be used to trigger appropriate vehicle responses, such as adjusting speed, stopping, or changing direction based on the recognized traffic sign. The system logs each detection event for further analysis and continuous improvement of the model. Overall, the output of the TSR system significantly enhances the decision-making process of autonomous vehicles, contributing to safer and more efficient navigation.

Fig : 7 Output

IV. RESULTS

Traffic Sign Recognition (TSR) using Convolutional Neural Networks (CNN) has emerged as a powerful approach in the field of computer vision, significantly enhancing the capabilities of autonomous driving systems. CNNs are particularly well-suited for TSR due to their ability to automatically learn and extract hierarchical features from raw image data, which is crucial for accurately identifying a wide variety of traffic signs under diverse conditions. By leveraging large labeled datasets and employing deep learning techniques, CNN-based models can achieve high accuracy in detecting and classifying traffic signs, even in the presence of noise, occlusions, and varying lighting conditions. This advancement not only improves the safety and efficiency of autonomous vehicles but also contributes to the development of intelligent transportation systems that can better assist drivers and reduce the likelihood of traffic-related incidents.

V. DISCUSSION

The discussion on the Traffic Sign Recognition (TSR) system using Convolutional Neural Networks (CNNs) highlights several key aspects and potential areas for further research and improvement. One of the primary advantages of using CNNs for TSR is their ability to automatically learn and extract relevant features from traffic sign images, leading to high accuracy in classification tasks. However, challenges such as varying lighting conditions, occlusions, and distortions in real-world scenarios need to be addressed to ensure the robustness of the system. Additionally, the system's performance can be impacted by the diversity and quality of the training dataset, emphasizing the importance of using comprehensive and well-annotated datasets. The real-time processing capability of the system is crucial for its application in autonomous vehicles and ADAS, but it also demands optimization for computational efficiency, especially on edge devices. Another area of discussion is the integration of the TSR system with other sensory inputs, such as LIDAR and radar, to enhance the overall perception and decision-making process. The ethical implications of deploying such systems, including ensuring the reliability and accountability of autonomous vehicles, are also important considerations. Continuous testing and validation in diverse environments are essential to maintain high standards of safety and performance. The discussion concludes by acknowledging the promising advancements in TSR using CNNs while recognizing the ongoing need for research to overcome current limitations and enhance the system's capabilities.

VI. CONCLUSION

The proposed system is simple and does the classification quite accurately on the GTSRB dataset as well as the newly generated one (consisting of truly existing images of all type), and finally the model can successfully capture images and predict them accurately even if the background of the image is not much clear. The proposed system uses Convolutional Neural Network (CNN) to train the model. The images are pre-processed, and histogram equalization is done to enhance the image contrast. The final accuracy on the test dataset is 93% and on the built dataset is 69%. The web cam predictions done by the model are also accurate and take very less time. The benefits of "Traffic Sign classification and detection system" are generally focused on driver convenience. Despite the advantages of traffic sign classification, there are drawbacks. There can be times when the traffic signs are covered or not visible clearly. This can be dangerous as the driver won't be able to keep a check on his vehicle speed and can lead to accidents, endangering other motorists or pedestrians, demanding further research

6.1 Future Scope

The application of Traffic Sign Recognition (TSR) using Convolutional Neural Networks (CNNs) extends beyond autonomous vehicles to include automatic robots and drones. By incorporating TSR technology, these autonomous systems can interpret and respond to predefined signs, similar to traffic signals, to navigate and perform tasks efficiently. For instance, drones equipped with TSR capabilities can recognize no-fly zones, landing signals, and directional markers, enhancing their ability to operate safely and autonomously in various environments. Similarly, robots can use TSR to follow specific pathways, adhere to operational protocols in industrial settings, and interact seamlessly with human-operated traffic systems. The integration of CNN-based TSR in robots and drones promises significant advancements in automation, enabling these systems to make informed decisions based on visual cues, improving their reliability and versatility in diverse applications.

VII. ACKNOWLEDGEMENT

Mrs. Pilla Devi Prasanna working as an Assistant Professor in Masters of Computer Applications(MCA) in SVPEC, Visakhapatnam, Andhra Pradesh. Completed her Post Graduation in Andhra University College of Engineering(AUCE). With one year experience, accredited by NAAC with her areas of interest in python, Database Management System, PSQT, Flat. Also qualified in APSET – 2024 exam.

Mr. Dushyanth Cheepilli is pursuing his final semester MCA in Sanketika Vidya Parishad Engineering College, accredited with A grade by NAAC, affiliated by Andhra University and approved by AICTE. With interest in Artificial intelligence Ch. Dushyanth has taken up his PG project on Traffic Sign Recognition using CNN for college enquiry and published the paper in connect to the project under the guidance of P. Devi Prasanna, Assistant Professor, SVPEC.

REFERENCES

Book reference

[1] A Book Titled on Guide to Convolutional Neural Network by Hamed Habibi Aghdam in Springer Link Linked : https://link.springer.com/book/10.1007/978-3-319-57550-6

[2] A Book Titled on Traffic sign recognition based on multi-algorithmic fusion by Samira El Margae Linked : http://surl.li/igushk

Web reference

[3] A web reference on Traffic-sign recognition in Wikipedia Linked : https://en.wikipedia.org/wiki/Traffic-sign_recognition

[4] A Web Reference on Traffic Sign Detection using Convolutional Neural Network in Wikipedia Linked : https://towardsdatascience.com/traffic-sign-detection-using-convolutional-neural-network-660fb32fe90e

Article reference

5. A journal on CNN Based Approach for Traffic Sign Recognition System by Nikita Malik in ADVANCED JOURNAL OF GRADUATE RESEARCH Linked : https://journals.aijr.org/index.php/ajgr/article/view/3851

6. A journal on Traffic sign detection and recognition using fully convolutional network guided proposals by Yingying Zhu in NEURO COMPUTING Linked : https://www.sciencedirect.com/science/article/abs/pii/S092523121630741X

7. A journal on Traffic Sign Recognition Using a Multi-Task Convolutional Neural Network by Hengliang Luo in IEE Xplore Linked : https://ieeexplore.ieee.org/abstract/document/7959631

[8] A journal on Traffic Sign Detection with Convolutional Neural Networks by Evan Peng in Springer Link Linked : https://link.springer.com/chapter/10.1007/978-981-10-5230-9_24

[9] A journal on Traffic Sign Detection and Recognition using Deep Learning by P. Puneeth Kumar in Springer Link Linked : https://link.springer.com/chapter/10.1007/978-981-19-2350-0_34

11. A journal on Traffic sign recognition and analysis for intelligent vehicles by A. de la Escalera in Science Direct Linked : https://www.sciencedirect.com/science/article/abs/pii/S0262885602001567

12. A journal on Traffic Sign Recognition for Computer Vision Project-Based Learning by David Gerónimo in IEEE XPLORE Linked : https://ieeexplore.ieee.org/abstract/document/6425440

13.A journal on TRAFFIC SIGN RECOGNITION WITH CONVOLUTIONAL NEURAL NETWORK by Sharipa Temirgaziyeva in Scientific journal of Astana IT University Linked : http://journal.astanait.edu.kz/index.php/ojs/article/view/20

[14] A journal on CNN Based Traffic Sign Recognition for Mini Autonomous Vehicles by Furkan Tufan in Springer Link Linked : https://link.springer.com/chapter/10.1007/978-3-319-99996-8_8

[15] A journal on A CNN - based Traffic Sign Recognition System by S. Sathiya in Springer Link Linked : https://link.springer.com/chapter/10.1007/978-981-33-6400-4_8

[16] A journal on Traffic Sign Recognition System (TSRS) : SVM and Convolutional Neural Network by Nazmul Hasan in Springer Link Linked : https://link.springer.com/chapter/10.1007/978-981-15-7345-3_6

[17] A journal on Real – Time Traffic Sign Recognition Using Convolutional Neural Networks by Aditya Rao in Springer Link Linked : https://link.springer.com/chapter/10.1007/978-981-33-4367-2_23

[18] A journal on Traffic Sign Recognition Based on Deep Convolutional Neural Network by Shihao Yin in Springer Link Linked : https://link.springer.com/chapter/10.1007/978-981-10-7299-4_57

[19] A journal on Traffic Sign Recognition Using Faster R – CNN Deep Learning Algorithm by Prachi Dewan in Springer Link Linked : https://link.springer.com/chapter/10.1007/978-981-97-0562-7_27

[20] A JOURNAL ON REVOLUTIONIZING : CNN-BASED TRAFFIC SIGN RECOGNITION BY SRINIVAS MALLADI IN IEEE XPLORE

Linked : https://ieeexplore.ieee.org/abstract/document/10511094