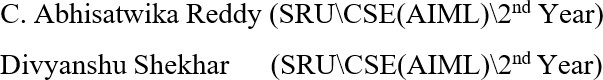
INTERNSHIP PROJECT REPORT

## On

TRAFFIC SIGNS DETECTION AND

RECOGNITION USING DEEP LEARNING

Submitted by



Under the guidance of Prof. T. Kishore Kumar

Professor, Department of ECE

NIT Warangal

**Centre for Training and Learning** NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL **HANAMKONDA - 506004, TELANGANA, INDIA**

## BONAFIED CERTIFICATE

This is to certify that this project report entitled **“**Traffic signs detection and recognition using deep learning**”** submitted to National Institute of Technology, Warangal is a bonafide record of work done by **“C. Abhisatwika Reddy, Divyanshu Shekhar”** under my supervision from **“01 June 2024”** to **“30 June 2024”**

### Supervisor

Prof. T. Kishore Kumar Professor, Department of ECE

NIT Warangal

Place: Warangal Date: 30 June 2024

## DECLARATION

We confirm that this report is our own original work. All external sources have been appropriately referenced and credited. We accept full responsibility for any content that may be identified as improperly cited or not original.

Submitted by

C. Abhisatwika Reddy

Divyanshu Shekhar

Place: Warangal Date: 30 June 2024

## ACKNOWLEDGEMENT

We are profoundly grateful to the individuals and organizations whose support and contributions were instrumental in the successful completion of this project and thesis.

First and foremost, we extend our deepest appreciation to Dr. T. Kishore Kumar, Professor and Head of the Center for Training and Learning (CTL) at NIT Warangal. His constant motivation, guidance, and unwavering support were pivotal in helping us achieve this significant milestone.

We would like to acknowledge the NIT Warangal AIOT (Artificial Intelligence of Things) and Its Applications Internship, conducted under the auspices of CTL. The lessons taught in seminars were incredibly insightful and enriched our learning experience. The scholars of Dr. T. Kishore Kumar imparted knowledge in Arduino IDE, IoT fundamentals tailored for ECE and CSE students, and conducted daily Python programming sessions, all of which played a pivotal role in enhancing our understanding and proficiency in the AIOT field.

Our deepest gratitude goes to Dr. T. Kishore Kumar for providing us with the opportunity to acquire such valuable knowledge through this internship.

We also extend special thanks to Boeing for their generous sponsorship, which was crucial to the success of this internship program at NIT Warangal.

In conclusion, we sincerely thank everyone involved for their invaluable support and contributions. This journey would not have been possible without you, and we are truly privileged to have been part of this enriching experience.

## ABSTRACT

In this project, we present a comprehensive comparative analysis of traffic sign detection and recognition using Convolutional Neural Networks (CNN) and You Only Look Once version 9 (YOLOv9) models. Initially, the German Traffic Sign Recognition Benchmark (GTSRB), consisting of 43 distinct traffic sign classes, was pre-processed and utilized to train a CNN model. This model was trained using various data augmentation techniques to enhance its generalization capabilities. The trained CNN model was then integrated with an ESP32 camera module to facilitate real-time live web streaming and prediction on video data, demonstrating its practical applicability in dynamic environments.

Simultaneously, a YOLOv9 model was trained on a different traffic sign recognition dataset, specifically curated to challenge the model's detection capabilities under diverse conditions. The YOLOv9 model training involved fine-tuning on augmented datasets to ensure robustness. Key performance metrics such as Precision-Recall (PR) curves, confusion matrix, and F1 curves were obtained to evaluate the performance of both models comprehensively. The results demonstrated high accuracy and robust predictive capabilities on the test datasets, with the YOLOv9 model exhibiting superior detection speed and efficiency, while the CNN model showcased exceptional classification accuracy.

This comparative study highlights the strengths and limitations of each model in terms of detection and recognition accuracy, computational efficiency, and real-time applicability. The CNN model's high classification accuracy makes it suitable for applications where precise recognition is critical, while the YOLOv9 model's rapid detection capabilities make it ideal for real-time applications in advanced driver- assistance systems (ADAS) and autonomous vehicles. The findings provide valuable insights into the suitability of CNN and YOLO models for different aspects of traffic sign detection and recognition technology, paving the way for further advancements and integration into modern transportation systems.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO** |
| 1 | INTRODUCTION   * 1. Background to the study   2. Problem statement   3. Aim of the study   4. Objectives of the study | 7-9 |
| 2 | LITERATURE SURVEY   * 1. Overview of Existing Traffic Sign Recognition and Detection Methods   2. Gaps Identified | 9-10 |
| 3 | METHODOLOGY   * 1. Design Methodology   2. Hardware Description   3. Software Requirements   4. Algorithm   5. Block Diagram   6. Implementation Details   7. Project Description | 10-20 |
| 4 | RESULTS AND DISCUSSIONS   * 1. Hardware Implementation   2. Result Analysis | 20-23 |
| 5 | CONCLUSION AND FUTURE SCOPE   * 1. Conclusion   2. Future Scope | 24-24 |
| 6 | REFERENCES | 25-25 |

# INTRODUCTION

* 1. **Background to the study**

Traffic sign recognition and detection are essential for modern vehicle systems to navigate safely and sticky by traffic laws. Traditional methods, which relied on basic image processing techniques, often failed under varying conditions like poor lighting and occlusions. The advent of deep learning, particularly CNNs and object detection models such as YOLO, has greatly enhanced the precision and reliability of traffic sign detection. This project harnesses these advanced techniques to create a system that can perform real-time traffic sign recognition and prediction using an ESP32 camera, thereby addressing the limitations of earlier approaches and improving overall road safety.

* 1. **Problem Statement**

Traditional traffic sign recognition and detection methods, which rely on basic image processing techniques, often fail under real-world conditions due to varying lighting, weather, and occlusions. These methods struggle with the complexity and diversity of traffic signs, and their inability to process data in real-time hampers their effectiveness in advanced driver- assistance systems and autonomous vehicles. Additionally, the reliability and robustness required for safety-critical applications are often unmet by these conventional approaches. The advent of deep learning, specifically Convolutional Neural Networks (CNNs) and object detection models like You Only Look Once (YOLO), promises improved accuracy and efficiency. However, a comprehensive evaluation and practical integration of these models are needed to ensure real-time performance and robustness in diverse conditions. This project aims to address these challenges by developing and comparing CNN and YOLOv9 models for traffic sign detection and recognition, leveraging deep learning advancements to enhance the safety and reliability of intelligent transportation systems.

* 1. **Aim of the study**

The aim of this study is to develop and evaluate the effectiveness of Convolutional Neural Networks (CNNs) and You Only Look Once version 9 (YOLOv9) models for traffic sign detection and recognition. By training these models on comprehensive datasets such as the German Traffic Sign Recognition Benchmark (GTSRB) and other curated traffic sign datasets, and integrating them with an ESP32 camera module, the study seeks to create a robust system capable of real-time traffic sign recognition and prediction.

The objective is to compare the performance of these models in terms of accuracy, efficiency, and reliability under diverse conditions, including varying lighting, weather, and occlusions, as well as different types of traffic signs. This involves assessing the models' ability to quickly and accurately detect and classify traffic signs in real-world scenarios, which is crucial for advanced driver-assistance systems (ADAS) and autonomous vehicles.

Furthermore, the study aims to evaluate the computational efficiency of these models when deployed on real-time hardware, ensuring that they can operate within the resource constraints typical of onboard vehicle systems. By identifying the strengths and limitations of CNN and YOLOv9 models, the research seeks to provide valuable insights into their applicability in enhancing road safety and adherence to traffic regulations.

***1.4 Objectives of the study***

The objective of this project is to develop a highly accurate and efficient traffic sign recognition and detection system. By leveraging advanced deep learning models such as Convolutional Neural Networks (CNNs) and You Only Look Once version 9 (YOLOv9), and integrating them with real-time hardware like the ESP32 camera module, the system aims to significantly enhance road safety and vehicle functionality. This involves training the models on comprehensive datasets, including the German Traffic Sign Recognition Benchmark (GTSRB) and other curated datasets, to ensure they can handle a wide variety of traffic sign classes and conditions.

The project focuses on achieving robust classification and detection capabilities that can operate effectively under dynamic real-world conditions, such as varying lighting, weather, and occlusions. It seeks to ensure rapid and reliable traffic sign identification, essential for advanced driver-assistance systems (ADAS) and autonomous vehicles. The system's ability to process and recognize traffic signs in real-time is critical for making immediate driving decisions, thereby enhancing the safety and efficiency of transportation systems.

Additionally, the study aims to conduct a comprehensive performance evaluation of the CNN and YOLOv9 models, comparing their accuracy, speed, and computational efficiency. This comparison will provide valuable insights into the strengths and limitations of each model, guiding their application in different scenarios within intelligent transportation systems. Ultimately, this project aspires to advance the technology behind traffic sign recognition, contributing to the development of safer and more reliable autonomous driving solutions and intelligent vehicle system.

# LITERATURE SURVEY

* 1. **Overview of Existing Traffic Sign Recognition and Detection Methods**

Traffic sign recognition and detection are critical for autonomous driving and advanced driver-assistance systems. Traditional methods predominantly rely on image processing techniques such as edge detection, color-based segmentation, and template matching. These methods, while straightforward, often struggle with variations in lighting, occlusions, and distortions. Edge detection techniques identify the boundaries of traffic signs, but they can be highly sensitive to noise and changes in illumination. Color-based segmentation relies on the distinct colors of traffic signs but can be easily affected by lighting conditions and color fading. Template matching involves comparing detected signs with pre-stored templates, which can be computationally expensive and less effective with distorted or partially occluded signs.

Machine learning-based approaches, including Support Vector Machines (SVM) and decision trees, have improved performance by leveraging labeled datasets for training. SVMs are effective for binary classification tasks and have been used for distinguishing traffic signs from non-signs, but their effectiveness diminishes with multi-class classification due to scalability issues. Decision trees provide interpretability and can handle multi-class classification, yet they can become overly complex and prone to overfitting.

* 1. **Gaps Identified**

Our project addresses several gaps in traditional and conventional traffic sign recognition and detection methods. Traditional methods struggle with varied real-world conditions like inconsistent lighting and occlusions, while our CNN model, trained on the GTSRB dataset, enhances accuracy by learning complex features. Real-time implementation challenges posed by CNNs' high computational demands are overcome by integrating the model with the ESP32 camera module and optimizing for resource efficiency. The detection speed of YOLO, typically requiring optimization for embedded systems, is addressed with a lightweight YOLO v9 model suitable for the ESP32. We seamlessly integrate recognition and detection, unlike traditional methods, providing a comprehensive real-time solution. Our models are trained on diverse datasets, improving robustness to environmental variability, and we ensure reliability through comprehensive evaluation metrics. These innovations offer a robust, real-time solution for traffic sign recognition, significantly improving upon traditional methods and enhancing the safety and efficiency of autonomous vehicle systems.

# Methodology

* 1. **Design Methodolgy**

The design methodology for this study involves several key steps to develop and evaluate the CNN and YOLOv9 models for traffic sign detection and recognition:

### Dataset Selection and Preprocessing:

* + - **Dataset Selection:** Choose appropriate datasets for training and evaluation, such as the GTSRB and other relevant traffic sign datasets. Ensure these datasets cover a wide range of traffic sign classes and variations.
    - **Preprocessing:** Preprocess the datasets to standardize image sizes, enhance image quality, and augment data to increase model robustness. Techniques such as rotation, flipping, and brightness adjustment are applied to create a diverse training set.

### Model Training:

* + - **CNN Training:** Implement and train a CNN model using frameworks like TensorFlow/Keras or PyTorch. Optimize hyperparameters such as learning rate, batch size, and network architecture to achieve high classification accuracy on traffic sign images.
    - **YOLOv9 Training:** Train the YOLOv9 model using a suitable framework like Darknet or PyTorch-YOLO. Fine-tune the model on the chosen dataset to optimize detection performance and speed.

### Hardware Integration:

* + - **ESP32 Camera Module:** Integrate the trained CNN and YOLOv9 models with an ESP32 camera module for real-time traffic sign detection and recognition. Implement necessary interfaces and protocols to enable seamless communication between the models and the camera hardware.
    - **Performance Optimization:** Optimize model inference speed and computational efficiency to ensure real-time performance on the ESP32 platform, considering its computational limitations.

### Evaluation Metrics:

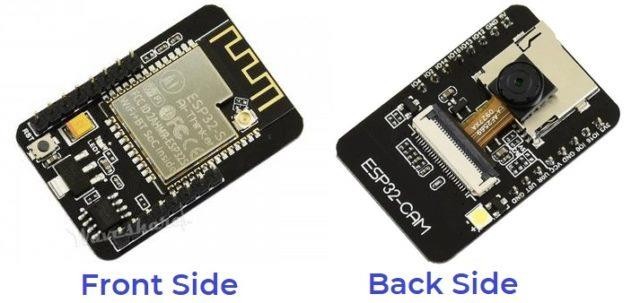
* + - **Accuracy Evaluation:** Assess the accuracy of both models using metrics such as precision, recall, and F1 score on a separate validation dataset. Generate confusion matrices to analyse the models' performance across different traffic sign classes.
    - **Real-Time Performance:** Measure the real-time performance of the integrated system, including frame processing speed and latency, to evaluate its suitability for dynamic driving environments.

### Comparative Analysis:

* + - **Performance Comparison:** Compare the performance of the CNN and YOLOv9 models in terms of accuracy, speed, and computational efficiency. Analyze their strengths and weaknesses under various environmental conditions and traffic scenarios.
  1. **Hardware Description**

For Hardware Implementation , Components used are :

*ESP32 camera*



*Figure 3.2.1 ESP32 Camera*

The ESP32 camera module is a low-cost, versatile solution ideal for capturing high-resolution images and streaming real-time video over Wi-Fi. It features a powerful ESP32 microcontroller, which enables basic onboard image processing and seamless wireless connectivity, making it perfect for IoT applications. Its compact design facilitates easy integration into various setups, including mobile and embedded systems.

**Role in our project** : The ESP32 camera was employed to capture live video streams, providing real-time input for our trained CNN model, enabling immediate traffic sign recognition and demonstrating its practical application in dynamic, real-world environments.

*FTDI Module*

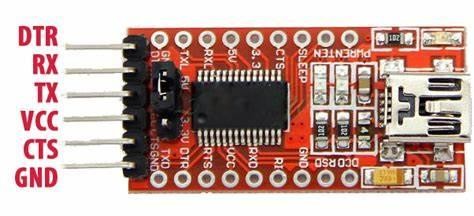


Figure 3.2.2 FTDI Module

Microcontroller boards without integrated USB connectivity can be programmed and debugged using the FTDI module, also known as a USB- to-Serial converter. A computer's USB signals are converted by the FTDI module into Universal Asynchronous Receiver-Transmitter (UART) signals that the ESP32 CAM can comprehend.

Role in project: During the development and debugging stages, the FTDI module is utilized for serial connectivity and firmware uploading to the ESP32 CAM.

*Circuit Diagram :*

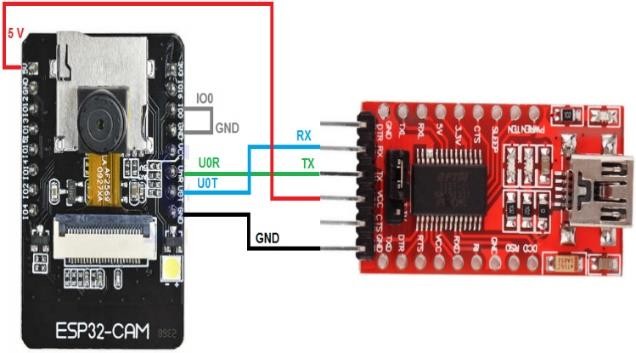


Figure 3.2.3 Circuit Diagram

**3.3 Software Requirements**

The software requirements for this project encompass the tools, frameworks, and environments necessary to develop, train, and integrate the CNN and YOLOv9 models for traffic sign detection and recognition, as well as for real-time deployment with an ESP32 camera module. Key components include:

**Development Environment:**

* *Jupyter Notebook*: Used for developing and executing the CNN model locally. It provides an interactive environment for data exploration, model training, and evaluation.
* *Google Colab*: Utilized for training the YOLOv9 model with special attention mechanisms. Colab provides access to high-performance GPUs, which are essential for accelerating model training and experimentation.

**Deep Learning Frameworks:**

* *TensorFlow/Keras*: Employed for implementing and training the CNN model. TensorFlow offers a comprehensive ecosystem for deep learning development, including high-level APIs through Keras for building and optimizing neural networks.
* *PyTorch*: Utilized for implementing and training the YOLOv9 model with special attention mechanisms. PyTorch is known for its dynamic computation graph and ease of use in research settings, making it suitable for complex model architectures and experimentation.

**Real-Time Integration:**

* *ESP32 Platform*: Integrated with the trained models for real-time traffic sign detection and recognition. This platform requires appropriate firmware and software libraries to interface with the CNN and YOLOv9 models, enabling seamless integration with the ESP32 camera module.
* *Webcam Access*: Enabled through local host setups for testing the integrated system's functionality with real-time video streams. This involves configuring web server environments and interfacing with the ESP32 camera module via network protocols.

**Additional Libraries and Tools:**

* *OpenCV*: Utilized for image processing tasks, webcam access, and interfacing with the ESP32 camera module. OpenCV provides essential functionalities for real-time computer vision applications, including image manipulation, object detection, and video streaming.
* *Darknet (YOLO framework*): Integrated with YOLOv9 for training and testing the object detection model. Darknet is a neural network framework that supports training and inference for YOLO models, optimized for speed and efficiency.

*Version Control and Collaboration*:

*- Git/GitHub:* Used for version control, collaborative development, and project management. GitHub repositories store project code, documentation, and experiment results, facilitating teamwork and reproducibility.

* 1. **Algorithm**
* Collect Dataset: This step involves gathering a collection of images that will be used to train the CNN. The dataset should be labeled so that the CNN can learn to identify the different objects or concepts in the images.
* Preprocess Data: The data preprocessing step involves preparing the images in the dataset for training. This may include resizing the images, normalizing the pixel values, and converting the images to a format that the CNN can understand.
* Feature Selection: In this step, a subset of features is selected from the images. These features will be used by the CNN to classify the images.
* Classification: The classification step is divided to 2 sub branches CNN model and YOLOV9 with special attention module
  + CNN Model : The flowchart then splits into two paths, which appear to be two different CNN models for classification.
  + YOLOv9 detection: This path trains a YOLOv9 detection model. YOLOv9 is a real-time object detection system that can identify multiple objects in an image.
* Performance Comparison: Once both models are trained, their performance is compared on a test dataset. The test dataset is a set of images that the CNN has not seen before. The performance comparison helps to determine which model is more accurate for the task at hand.
* Click Test Image: Here, a test image is clicked using ESP32 Camera to make predictions later
* Make Predictions: Make predictions on test image clicked with trained CNN model.

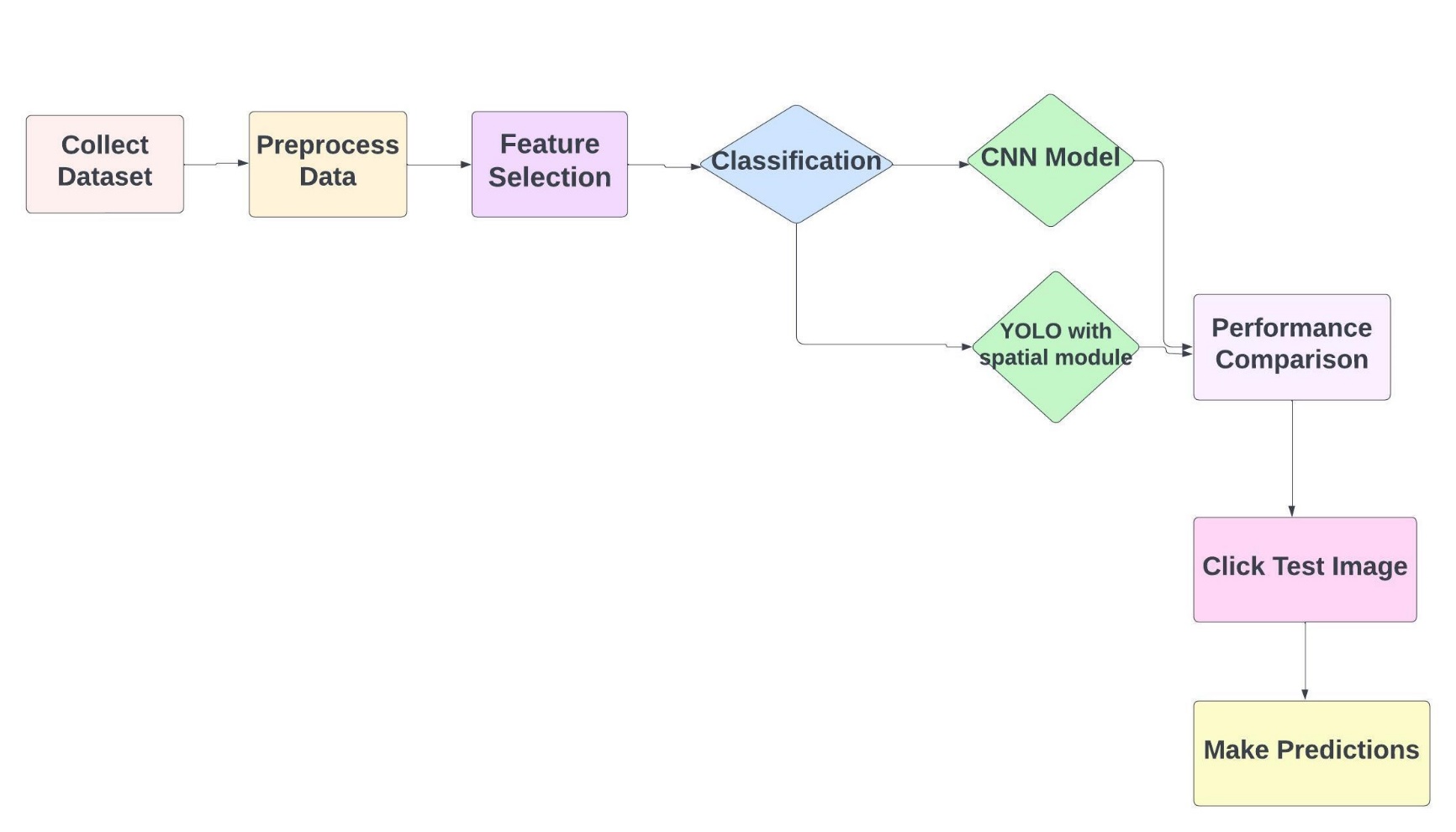


Figure 3.3 Algorithm

* 1. **Block Diagram**

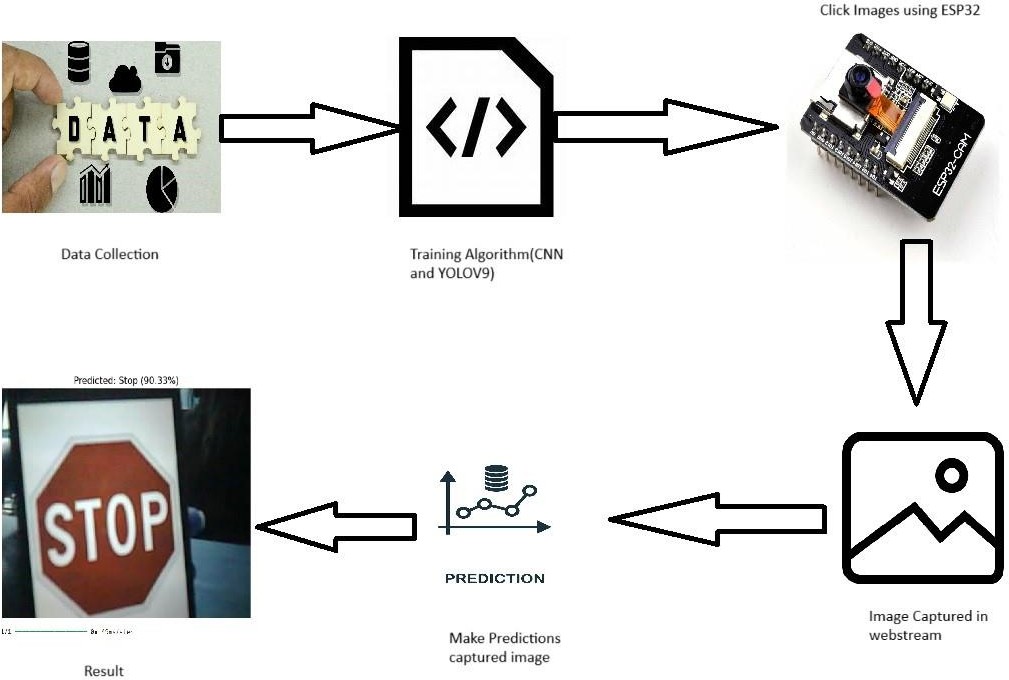


Figure 3.4 Block Diagram

* 1. **Implementation Details**

*Dataset Preparation and Model Training*

The implementation of our traffic sign detection and recognition system begins with meticulous dataset preparation and model training. We selected the German Traffic Sign Recognition Benchmark (GTSRB) and augmented it with additional datasets to ensure comprehensive coverage of traffic sign variations and conditions. Through extensive preprocessing, including normalization and data augmentation techniques such as random rotation and scaling, we aimed to enhance the model's ability to generalize across diverse real-world scenarios.

*Architectural Decisions and Innovation*

Central to our approach is the simultaneous training of Convolutional Neural Networks (CNN) and You Only Look Once version 9 (YOLOv9) models. This dual-model strategy not only leverages CNNs for accurate traffic sign recognition but also harnesses YOLOv9's efficiency in real-time object detection. Innovatively, we integrated special attention mechanisms into the YOLOv9 architecture to prioritize the detection of traffic signs amidst complex visual clutter and varying environmental conditions.

*Real-Time Integration and ESP32 Deployment*

A pivotal aspect of our implementation involved the seamless integration of trained models with an ESP32 camera module. This integration necessitated the optimization of model inference pipelines for real-time performance on embedded systems. Leveraging the ESP32’s capabilities, we developed protocols for efficient data transmission and synchronization, ensuring responsive and reliable traffic sign detection in dynamic driving environments.

*Iterative Refinement and Performance Optimization*

Throughout the implementation process, we adopted an iterative approach to refinement and optimization. We continually fine-tuned model hyperparameters, evaluated performance metrics such as precision-recall curves and F1 scores, and conducted rigorous testing in simulated and real- world scenarios. Each iteration informed adjustments in model architecture, algorithmic enhancements, and system configurations to achieve optimal accuracy, speed, and robustness.

*Documentation and Collaboration*

Documentation played a crucial role in maintaining transparency and facilitating collaboration among team members. Comprehensive documentation encompassed detailed code annotations, experiment logs, and deployment instructions. Version control using Git/GitHub ensured version tracking, code review, and seamless integration of contributions, fostering a cohesive development process.

* 1. **Project Description**

**Data Collection**

Two primary datasets were utilized to train and evaluate the models for traffic sign recognition

and detection

*German Traffic Sign Recognition Benchmark (GTSRB)*

It consists of 43 different traffic sign classes, providing a comprehensive set of real-world images. This dataset contains over 50,000 images of traffic signs captured under various conditions, including different lighting, weather, and perspectives. Each image is labeled with its corresponding traffic sign class, making it suitable for training convolutional neural networks (CNNs). The diversity and volume of the GTSRB dataset enable the CNN model to learn complex features and patterns, improving its ability to accurately classify traffic signs under varied real-world conditions.

*Kaggle Traffic Sign Detection Dataset*

Simultaneously, in the other section, for training a dataset on YOLO model we chose, Kaggle Traffic Sign Detection dataset, which comprises 4,969 images annotated with 15 classes of traffic signs. These classes include Green Light, Red Light, Speed Limit 10, Speed Limit 100, Speed Limit 110, Speed Limit 120, Speed Limit 20, Speed Limit 30, Speed Limit 40,Speed Limit 50, Speed Limit 60, Speed Limit 70, Speed Limit 80, Speed Limit 90, and Stop signs. Each image in this dataset is annotated with bounding boxes and labels for the traffic signs present, making it ideal for training YOLO (You Only Look Once) models for object detection.

***Preprocessing Data***

*For CNN Model*

In the data preprocessing phase of our study, we focused on augmenting and balancing the training dataset for the Convolutional Neural Network (CNN) model used in traffic sign recognition. Initially, we utilized the German Traffic Sign Recognition Benchmark (GTSRB) dataset, which contains images representing 43 different traffic sign classes. Each image underwent augmentation through rotations, translations, and perspective transformations to increase variability and enhance the dataset's robustness. This augmentation aimed to simulate various real-world conditions such as lighting changes and different perspectives, thereby improving the model's ability to generalize.

To address class imbalance within the dataset, we implemented a balancing strategy. This involved generating synthetic examples for underrepresented classes using the augmented transformations. By ensuring each traffic sign class had an adequate number of examples, we aimed to mitigate bias during model training and improve overall classification accuracy.

The preprocessing steps were designed specifically for training the CNN model to recognize traffic signs accurately.

*For YOLOV9 model*

We integrated a Spatial Attention module to enhance feature extraction and improve object detection accuracy. This module is designed to highlight relevant spatial regions within the input feature maps, therefore directing the model's focus towards important features for better localization and classification of traffic signs.

The Spatial Attention module consists of a convolutional layer followed by a sigmoid activation function. It takes the input feature maps and computes both average and maximum pooling operations across channels to capture different aspects of spatial information. By concatenating and processing these pooled feature maps through a convolutional operation, the module learns to emphasize salient features while suppressing irrelevant ones.

**Deep Learning Algorithms used**

We used two different methods to recognize and detect traffic signs: one with TensorFlow/Keras and the other with PyTorch.

**Convolutional Neural Network (CNN) in TensorFlow/Keras:**

In TensorFlow/Keras, we built a sequential neural network for classifying images. The model starts with two layers that filter out features using ReLU activation functions on small, 32x32-pixel images. The model then reduces these images' size with max pooling and stops them from fitting too closely with dropout. Later layers identify more subtle details in traffic signs using even more filters. To guess the type of traffic sign, we use the final layers, which translate the flattened features into 43 different potential outcomes.

**YOLOv9 with Spatial Attention in PyTorch:**

We also used PyTorch's YOLOv9, a top-notch tool for spotting items with a strong balance of speed and correctness. This software has a base for getting features, a midsection for putting these features together, and a crown for estimating the locations and sorts of objects. To make this even better, we put in a Spatial Attention part. This part focuses on the most meaningful locations, which helps us locate and define traffic signs, especially when the weather or lighting isn't good.

**ESP32 Camera Setup**

To implement real-time traffic sign recognition, we utilized an ESP32 camera module for live video streaming. First, necessary libraries and board support were installed in the Arduino IDE. The ESP32 camera module was connected to an FTDI programmer to enable flashing of the firmware. The module was configured to connect to a Wi-Fi network, stream video, and send captured frames to our classification model. Once the code was uploaded to the ESP32 camera, the live stream could be accessed through a web browser using the assigned IP address. This setup provided real-time video input for our CNN model, enabling immediate traffic sign recognition and practical application in dynamic, real-world environments.

*Real-Time Traffic Sign Detection Using ESP32 Camera Video Stream Source:*

The ESP32 camera streams video at <http://172.20.238.160/capture>. (edit)

*Frame Fetching:*

* Continuously fetch frames from the video stream using HTTP requests.
* Convert the byte str-eam into image arrays.

*Preprocessing:*

* Resize frames to 32x32 pixels.
* Normalize pixel values for CNN input.

*Prediction:*

* Feed pre- processed frames into the trained CNN model.
* Predict traffic sign classes and confidence scores.

*Display:*

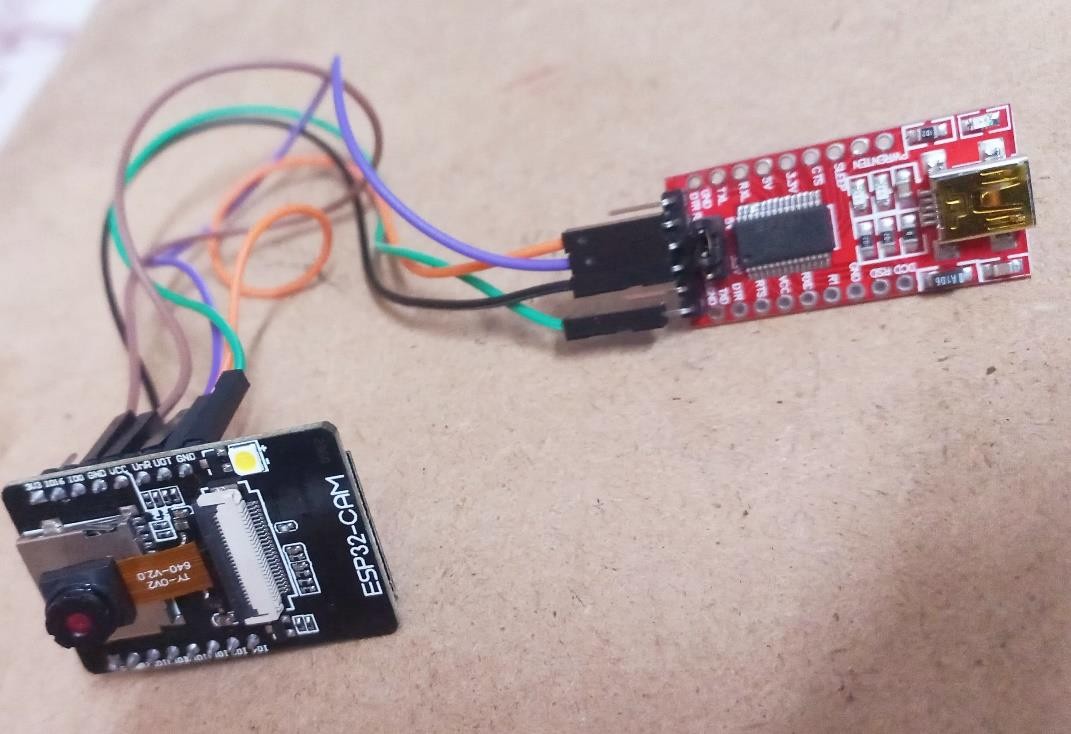
* Show frames with predicted labels and confidence scores in real- time.
* Include a mechanism to stop processing on a specific key press (e.g., 'q').

*Application:*

* Demonstrates the feasibility of using ESP32 cameras for real-time traffic sign detection.
* Highlights the practical deployment of deep learning models in dynamic environments for traffic monitoring and autonomous driving systems.

# Results and Discussion

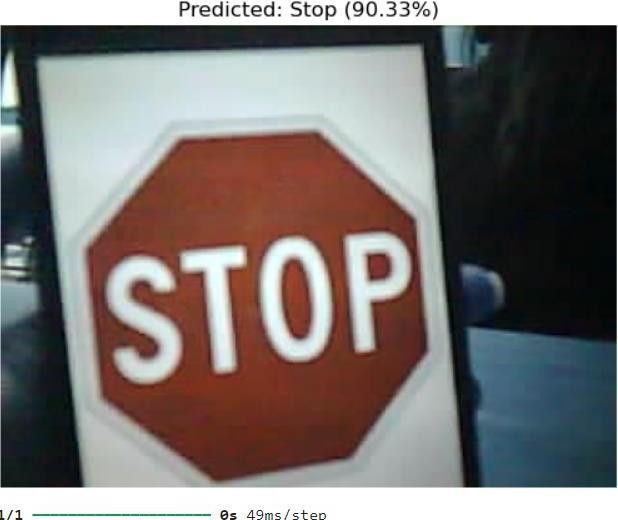
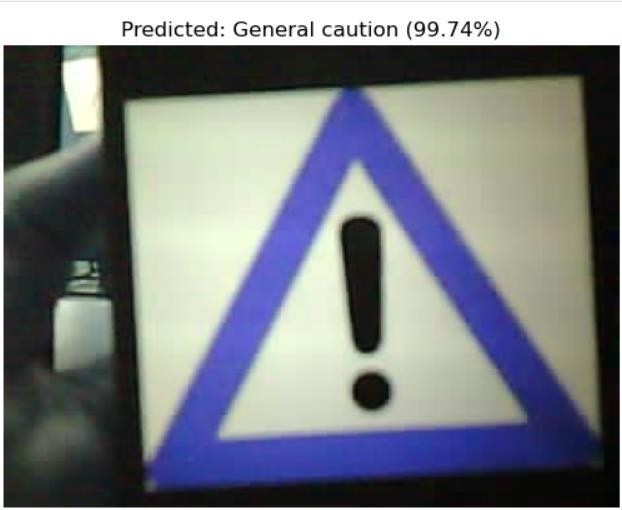
* 1. **Hardware Implementation**



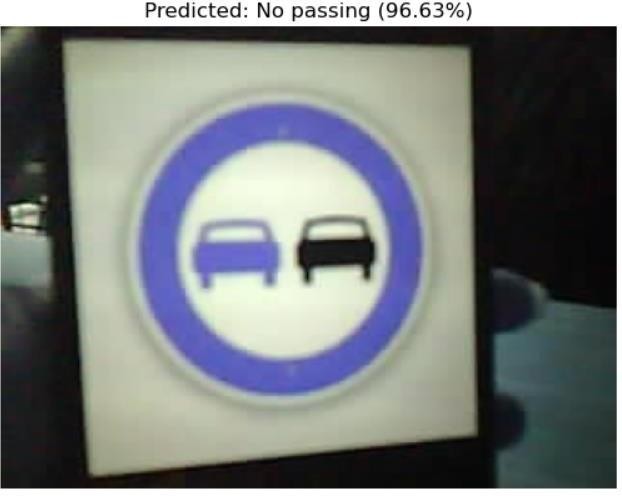
*Figure 4.1 Hardware Implementation*

* 1. **RESULT ANALYSIS**

***CNN MODEL***



*Figure 4.2.1 CNN Model Results*



***YOLOV9 Model***



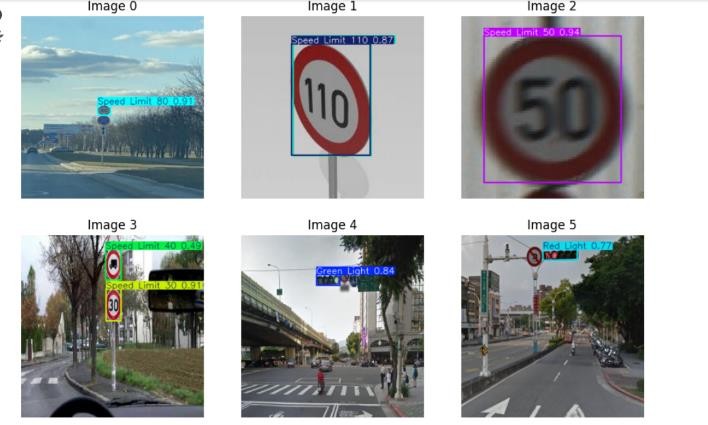


Figure 4.2.2 YOLOV9 Model Results

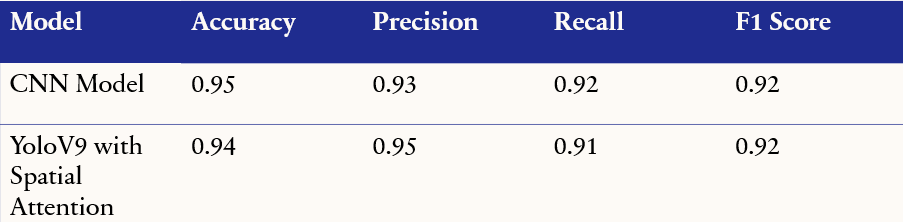
**Comparative Analysis:**

Precision and Recall: The YOLO model excels in precision and recall, making it highly reliable for real-time traffic sign detection where both accuracy and the ability to detect all relevant signs are crucial.

Training Efficiency: The CNN shows rapid learning and convergence, indicating efficient training. However, the YOLO model's detailed precision and recall metrics provide a more nuanced understanding of its performance across different confidence levels.

F1 Score: The YOLO model's high F1 score demonstrates its balanced performance, which is critical for applications requiring both high precision and recall.

In summary, while both models perform well, the YOLO model's detailed performance metrics and high precision, recall, and F1 scores make it particularly suitable for traffic sign detection tasks. The CNN, with its rapid learning and low loss, also shows strong performance, making it a viable option depending on the specific requirements of the application.



*Figure 4.2.3 Comparative Analysis*

# Conclusion and Future Scope

**Conclusion**

In conclusion, our project delved into the realm of traffic sign detection and recognition using Convolutional Neural Networks (CNN) and You Only Look Once version 9 (YOLOv9) models. By training these models on separate datasets and conducting a comparative analysis based on precision- recall curves, confusion matrices, and F1 scores, we identified distinct strengths in each approach. The CNN model exhibited superior accuracy and reliability in classifying traffic signs under varying conditions, showcasing its potential for precise real-time applications. Meanwhile, YOLOv9 demonstrated efficient object detection capabilities suitable for rapid processing and detection tasks. While we integrated the ESP32 camera module with the CNN model for practical deployment, future research could explore hybrid model integrations and optimization strategies to further enhance performance in intelligent transportation systems and autonomous vehicles. Overall, our study contributes valuable insights into the capabilities of deep learning models for enhancing road safety and traffic regulation compliance.

**Future Scope**

Looking ahead, our project lays the foundation for several promising avenues in the field of traffic sign detection and recognition. Moving forward, one key area of exploration involves integrating hybrid models that combine the strengths of Convolutional Neural Networks (CNNs) and You Only Look Once version 9 (YOLOv9) or similar models. This hybrid approach could leverage CNNs' accuracy in precise classification with YOLOv9's efficiency in real-time object detection, potentially enhancing both speed and reliability in dynamic traffic environments. Additionally, future research could focus on advancing hardware integration beyond the ESP32, optimizing models for edge computing and autonomous vehicle applications. Exploring semantic segmentation techniques for finer contextual understanding and robustness under adverse conditions like varying weather and lighting remains crucial. Moreover, integrating these systems with broader traffic management frameworks could further improve safety and efficiency on roadways. Ethical considerations, such as privacy safeguards and compliance with regulatory standards, will also play a pivotal role in shaping the development and deployment of these technologies. By continuously refining datasets and methodologies, our work aims to contribute to safer, smarter transportation systems, ultimately benefiting communities worldwide.

**REFERENCES**

[1] Yang, Y., Luo, H., Xu, H., & Wu, F. (2016). Towards real-time traffic sign detection and classification. IEEE Transactions on Intelligent Transportation Systems, 17(7), 2016.

[2] He, Z., & Nan, F. (2019). Traffic sign recognition by combining global and local features based on semi-supervised classification. IET Intelligent Transport Systems. Retrieved from ISSN 1751-956X.

[3] Singh, Mahesh, Lakhvindra Singh, et al." International Journal for Research in Applied Science & Engineering Technology (IJRASET), June 2021

[4] Detection and Violation Control." In Proceedings of the Second International Conference on Inventive Research in Computing Applications (ICIRCA-2020), 2020.

[5] I. S. B. Md Isa and C. Ja Yeong, "Real-time traffic sign detection and recognition using Raspberry Pi," \*International Journal of Electrical and Computer Engineering (IJECE)\*, vol. 12, no. 1, pp. 331-338, Feb. 2022

[6] Tabernik, D., & Skočaj, D. (2020). Deep learning for large-scale traffic-sign detection and recognition. IEEE Transactions on Intelligent Transportation Systems, 21(4), 1486-1497.

[7] Fleyeh, H., & Dougherty, M. (Year). Road and traffic sign detection and recognition. In Advanced OR and AI methods in transportation.

[8] ROAD AND TRAFFIC SIGN DETECTION AND RECOGNITION Hasan FLEYEH1 , Mark DOUGHERTY

[9] German Traffic Sign Recognition Benchmark [[dataset](https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign)](CNN)

[10] Traffic Signs Detection[dataset](YOLOV9)