Traffic Sign Detection and Prediction System

Abstract - In this study, we present a 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 dataset benchmark, consisting of 43 distinct traffic sign classes, was pre-processed and utilized to train a CNN model. The trained model was then integrated with an ESP32 camera module to facilitate real-time live web streaming and prediction on video data. Simultaneously, a YOLOv9 model was trained on a different traffic sign recognition dataset. We obtained key performance metrics such as Precision-Recall (PR) curves, confusion matrix, and F1 curves to evaluate its performance. The results demonstrated the model's high accuracy and robust predictive capabilities on the test dataset.

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 findings provide valuable insights into the suitability of CNN and YOLO models for advanced driver-assistance systems (ADAS) and autonomous vehicles, paving the way for further advancements in traffic sign detection and recognition technology.

Keywords - Traffic Sign Recognition, Traffic Sign Detection, Convolutional Neural Networks, ESP32 camera, FTDI Module, YOLO V9 model IOT integration, Autonomous Vehicles , Contour Detection, Open CV , Advanced driver-assistance systems (ADAS).

I. Introduction:

The increasing need for intelligent transportation systems and autonomous vehicles has created a demand for reliable and efficient traffic sign detection and recognition systems. Accurate recognition of traffic signs is crucial for enhancing road safety, providing essential information to drivers, and enabling autonomous vehicles to navigate complex traffic environments effectively. This project addresses these needs by developing and comparing two advanced machine learning models for traffic sign detection and recognition: Convolutional Neural Networks (CNN) and You Only Look Once version 9 (YOLOv9).

In this study, we explore the potential of these models in real-time traffic sign recognition applications. We trained a CNN model on the German traffic dataset benchmark, which includes 43 distinct traffic sign classes. The trained model was then integrated with an ESP32 camera module to provide real-time live web streaming and make predictions on video data, simulating real-world driving conditions. The use of the ESP32 camera module offers a cost-effective and flexible solution for capturing and processing live video streams, making it highly suitable for real-time applications.

Additionally, we trained a YOLOv9 model on another traffic sign recognition dataset and evaluated its performance using key metrics such as Precision-Recall (PR) curves, confusion matrix, and F1 scores. The results highlighted the model's high accuracy and robust predictive capabilities on the test dataset.

Compared to existing methods, our approach leverages the latest advancements in deep learning and real-time video processing to enhance the accuracy and efficiency of traffic sign detection and recognition. The integration of the ESP32 camera module with the CNN model provides a practical and scalable solution for real-time applications, offering significant improvements over traditional methods that may rely on more expensive and less flexible hardware.

II. Objectives

Data Preprocessing and Model Training:

Preprocess the German Traffic Sign Recognition Benchmark (GTSRB) dataset for optimal CNN training. Train a CNN model to classify 43 different traffic sign classes with high accuracy.

Real-Time Traffic Sign Recognition:

Integrate the trained CNN model with an ESP32 camera module.

Develop a system to provide real-time live video streaming and traffic sign predictions.

Advanced Object Detection with YOLO v9:

Train a YOLO v9 model on a separate traffic recognition dataset.

Evaluate the YOLO v9 model using performance metrics such as precision-recall curves, F1 scores, and confusion matrices.

***Performance Comparison**:*

Compare the efficacy of the developed models with conventional traffic sign recognition methods.

Demonstrate improvements in accuracy, speed, and real-time application capabilities.

Comparative Analysis :

Compared to conventional traffic sign recognition methods that rely on traditional computer vision techniques and less sophisticated machine learning models, our approach offers several advantages: The CNN model trained on the GTSRB dataset achieves higher classification accuracy by learning complex features from large datasets. Integrating the trained model with the ESP32 camera module enables live video streaming and instant traffic sign predictions, crucial for real-time applications like autonomous driving and ADAS. The YOLO v9 model enhances object detection with greater speed and accuracy. Our comprehensive evaluation using multiple performance metrics ensures reliable and accurate traffic sign recognition.

III. Literature Review

Overview of Existing Traffic Sign Recognition and Detection Methods

Traffic sign recognition and detection are critical for autonomous driving and advanced driver-assistance systems (ADAS). 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.

Previous Work on Traffic sign detection and recognition

The paper provides an overview of road and traffic sign detection and recognition. It discusses the characteristics of road signs, challenges in detection and recognition, techniques for image segmentation based on color and shape analysis, and methods for sign recognition and classification[1]. Traffic sign recognition is challenging for computer systems, despite being easy for humans, due to advancements in image-processing and machine-learning algorithms. Using a semi-supervised learning approach, the paper proposes combining global and local features for traffic sign recognition in an Internet-of-things-based transport system, demonstrating better performance than existing methods [2]. Traffic Sign Recognition (TSR) system developed using Machine Learning and Internet of Things (IoT) technologies to enhance road safety. Utilizes image processing techniques and Convolutional Neural Networks (CNN) for accurate and real-time recognition of traffic signs to assist drivers [3]. Vehicular automation heavily relies on the detection of traffic signs, with a significant growth in this field observed. The paper proposes a prototype utilizing Convolutional Neural Network (CNN) to detect traffic signs and prevent violations in real-time, enhancing road safety by denying the user control if traffic rules are violated [4]. An advanced driving assistance system (ADAS) was developed to enhance road safety by recognizing traffic signs in real-time within the car, aiming to reduce the increasing number of road accidents in Malaysia. The system utilized TensorFlow algorithm for high accuracy in object recognition, implemented on a Raspberry Pi 3 processor, which analysed real-time video recordings from a Raspberry Pi camera NoIR to detect various traffic signs, achieving over 90% accuracy and reliability with acceptable delay [5]. Automatic detection and recognition of traffic signs is essential for managing traffic-sign inventory efficiently with minimal human effort. The paper introduces a novel approach using a convolutional neural network (CNN) for detecting and recognizing 200 traffic-sign categories, achieving less than 3% error rates, suitable for practical applications in traffic-sign inventory management [6]. The paper focuses on exploring the impact of active galactic nuclei (AGN) on their host galaxies and the surrounding circumgalatic medium. Researchers used optical integral-field spectroscopy to study three nearby AGNs, finding complex morphologies and kinematics in the gas surrounding these objects [7].

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 and ADAS.

Using CNN and YOLO Integrated with ESP32 Technology

Our approach leverages the strengths of both CNNs and YOLO, integrated with the ESP32 camera module for real-time traffic sign recognition and detection. The CNN model trained on the GTSRB dataset ensures high classification accuracy, while YOLO v9 provides robust and fast object detection. The ESP32 camera module facilitates live video streaming, making real-time predictions feasible. This integration addresses the need for a low-cost, efficient, and accurate traffic sign recognition system suitable for real-world applications.

Difficulties to Be Taken into Consideration

Several challenges must be considered in this approach:

Computational Constraints: The ESP32 has limited processing power and memory compared to traditional computing platforms, necessitating efficient model optimization and compression techniques.

Real-Time Performance: Achieving low latency in predictions is crucial for real-time applications. This requires optimizing both the CNN and YOLO models for faster inference.

Environmental Variability: Traffic signs can appear under varying lighting conditions, occlusions, and perspectives. Ensuring robustness to these variations is essential for reliable performance.

Integration Challenges: Seamlessly integrating the CNN and YOLO models with the ESP32 camera module and ensuring smooth video streaming and prediction processes pose significant technical challenges.

IV. Methodology

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.

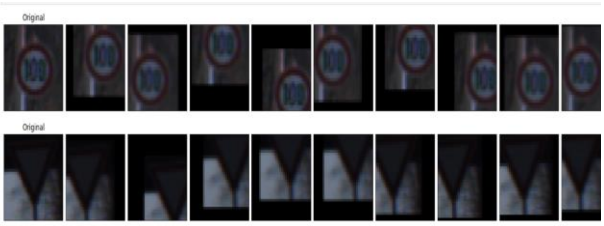
Preprocessing Data

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.

Testing Process

For CNN model

We assessed the CNN's performance using a real-time web stream captured by an ESP32 camera module. This testing evaluated how well the model identified and classified traffic signs in dynamic, real-world conditions.

For YOLOV9 model

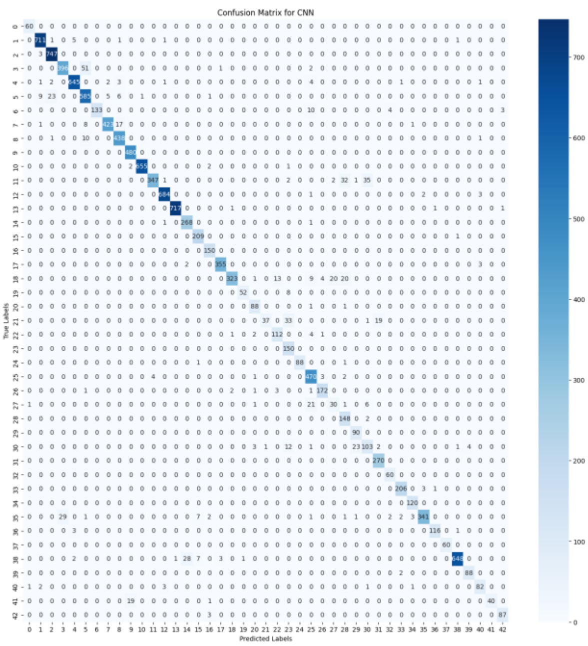
The YOLOv9 model was tested on the specific testing dataset available within the dataset repository. This testing phase verified the model's ability to detect and classify traffic signs according to their respective classes.

V. Experimental Results

Performance Metrics

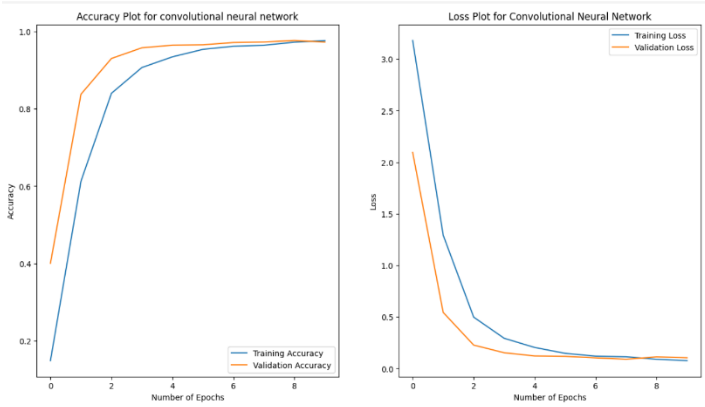
For CNN model

Confusion matrix: Is a table used to evaluate the performance of a classification model by displaying the counts of true positives, true negatives, false positives, and false negatives.



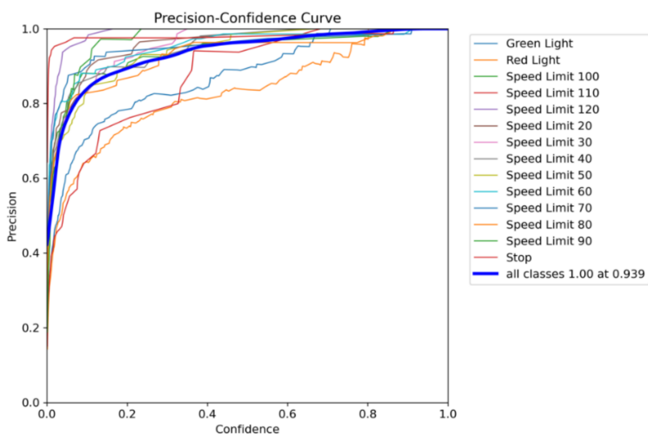
Modal accuracy: Refers to the accuracy of a classifier based on the most frequently predicted class, also known as the mode, across all predictions.

Model loss: Refers to the function that measures the error between the predicted output and the actual target values during model training.

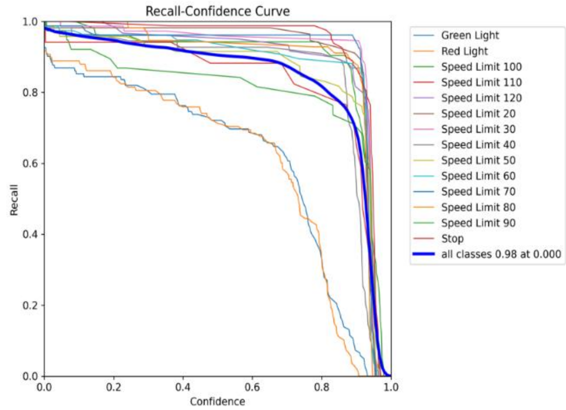


For YOLOV9 model

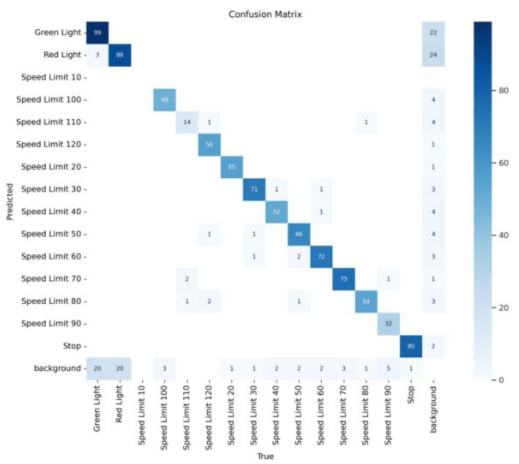
The Precision-Confidence Curve shows the precision of the YOLO model for various traffic sign classes at different confidence levels. Each line represents a class, with the bold line indicating overall performance, achieving a precision of 1.00 at a 0.939 confidence level. This curve helps assess the model's accuracy and determine the optimal confidence threshold for reliable traffic sign detection.



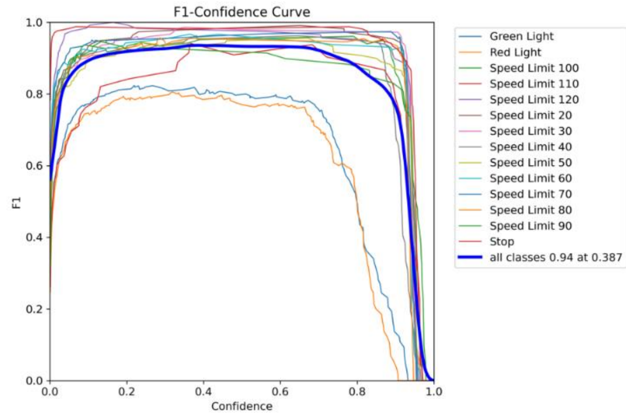
The Recall-Confidence Curve illustrates the recall of the YOLO model for various traffic sign classes at different confidence levels. Each line represents a specific class, with the bold line showing the overall recall, achieving 0.98 at a confidence level of 0.000. This curve is essential for evaluating the model's ability to correctly identify traffic signs across varying confidence thresholds, helping to determine the optimal balance between recall and confidence for effective traffic sign detection.

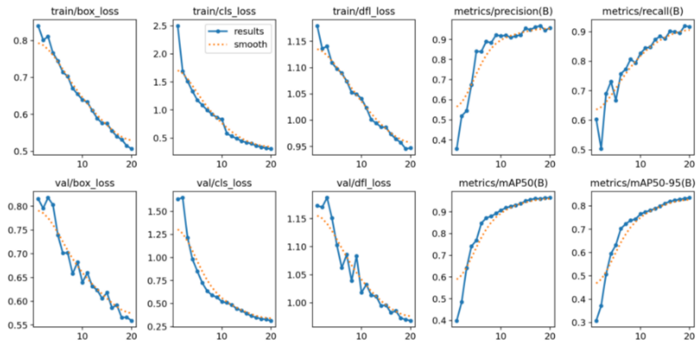


Confusion matrix: Is a table used to evaluate the performance of a classification model by displaying the counts of true positives, true negatives, false positives, and false negatives.



The F1-Confidence Curve displays the F1 scores of the YOLO model for various traffic sign classes across different confidence levels. Each line represents a specific class, with the bold line indicating the overall F1 score, achieving 0.94 at a confidence level of 0.387. This curve is crucial for assessing the model's balance between precision and recall, helping to identify the optimal confidence threshold that maximizes the F1 score for effective traffic sign detection.





Comparative Analysis:

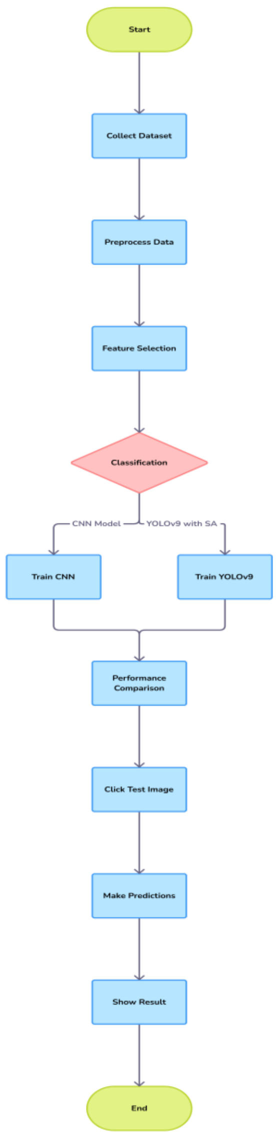
Precision and Recall: The YOLO model excels in precision and recall, making it highly reliable for real-time traffic sign detection where both accuracy and the ability to detect all relevant signs are crucial.

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.

VI. Algorithm Flowchart



VII. DISCUSSION

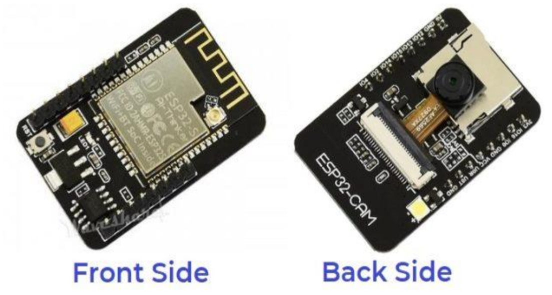
Real Time Testing Results

CNN MODEL

For Hardware Implementation , Components used are :

1. ESP32 CAM
2. FTDI MODULE
3. USB CABLE
4. JUMPER WIRES

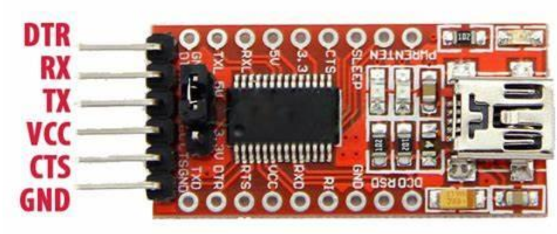
ESP32 camera



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.

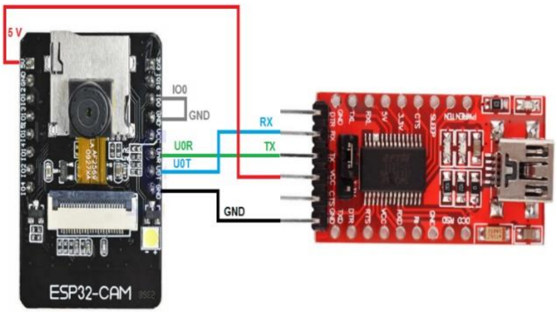
FTDI Module



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 :



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 stream 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.

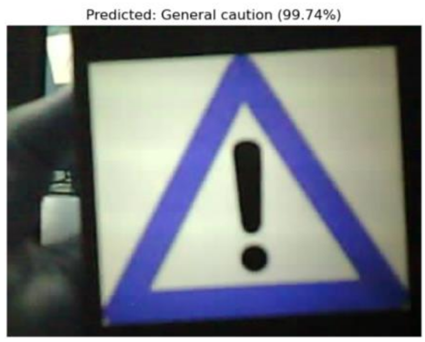
Display:

- 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:

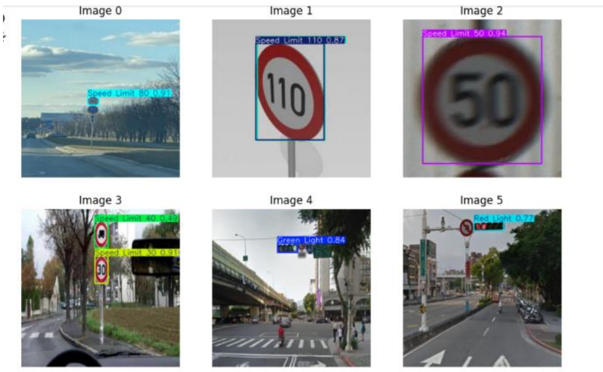
- Demonstrates the feasibility of using ESP32 cameras for real-time traffic sign detection.
- Highlights the practical deployment of deep learning models in dynamic environments for traffic monitoring and autonomous driving systems.

Output



YOLOv9 model

We made predictions on test data available in the dataset itself to identify model's accuracy. It detected , made contour boxes around the relevant traffic signs and made correct predictions indicating model is just working fine.



VIII Future Work and Improvements

Future enhancements to this traffic sign recognition and detection system include integrating more advanced deep learning models, such as newer versions of YOLO and deeper CNNs, to improve accuracy and speed. Optimizing the integration with the ESP32 camera will enable faster real-time processing and reduced latency. Implementing sophisticated data augmentation techniques will increase the robustness of the models. Enhancing the system's adaptability to varying lighting conditions, weather changes, and occlusions will improve its reliability in diverse environments. Expanding the dataset and model capabilities to recognize traffic signs from different regions and languages will make the system applicable globally. Developing a user-friendly GUI will facilitate easier interaction and visualization of real-time recognition and detection results. Integrating the system with advanced driver-assistance systems (ADAS) will improve safety and real-time decision-making. Finally, implementing a continuous learning framework will allow periodic updates with new data, ensuring better adaptability and accuracy over time. These improvements aim to enhance the overall performance, efficiency, and usability of the traffic sign recognition and detection system.

IX. Conclusion

This study evaluated the performance of Convolutional Neural Network (CNN) and YOLOV9 with special attention module models for traffic sign detection.

CNN Model:

- Achieved high training and validation accuracy quickly.
- Demonstrated effective error minimization without overfitting.

YOLO Model:

- Excelled in precision, recall, and F1 score, making it highly reliable for real-time detection.
- Successfully integrated with an ESP32 camera for practical, real-time traffic sign detection.

Comparative Analysis:

- The CNN model is efficient for quick training and generalization.
- The YOLO model is superior for real-time applications requiring high accuracy and reliability.

In summary, both models have distinct advantages: the CNN for efficient training and the YOLOv9 for high-precision, real-time detection. The choice depends on the specific application needs, with YOLO being ideal for scenarios demanding real-time performance. This study highlights the potential of deep learning models in enhancing traffic sign detection systems for safer and more efficient traffic management.

X. References

[1]ROAD AND TRAFFIC SIGN DETECTION AND RECOGNITION - Hasan FLEYEH1, Mark DOUGHERTY2

[2]Traffic sign recognition by combining global

and local features based on semi-supervised

classification ,ISSN 1751-956X - Zhenli He1, Fengtao Nan1

[3]A Research Paper on Traffic Sign Recognition with Machine Learning and IOT - Mahesh Singh1, Lakhvindra Singh2, Manish Sharma3, Kashish Gupta4, Giriraj Khandelwal5

[4]IoT based Traffic Sign Detection and Violation Control -Baratsanjeevi T1, Deepakkrishna S2, Harrine M P3, Sharan S4, and Dr.Prabhu E5

[5]Real-time traffic sign detection and recognition using Raspberry Pi - Ida Syafiza Binti Md Isa1, Choy Ja Yeong1, Nur Latif Azyze bin Mohd Shaari Azyze2

[6]Deep Learning for Large-Scale Traffic-Sign Detection and Recognition - Domen Tabernik and Danijel Sko ˇca

[7]Towards Real-Time Traffic Sign Detection and Classification - Yi Yang, Hengliang Luo, Huarong Xu, and Fuchao Wu.

[8]German Traffic Sign Recognition Benchmark [dataset](CNN)

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