TRAFFIC SIGN AND LIGHT DETECTION USING YOLOV5

A PROJECT REPORT

Submitted by

CHRISTY RAJ (TKM20MCA-2016)

to

The APJ Abdul Kalam Technological University

In partial fulfillment of the requirments for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

Thangal Kunju Musaliar College of Engineering
Kerala

DEPARTMENT OF COMPUTER APPLICATIONS

JULY 2022

DECLARATION

I undersigned hereby declare that the project report on TRAFFIC SIGN AND LIGHT DETECTION USING YOLOV5 submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of DR. NADERA BEEVI S. This submission represents my ideas in my own words and where ideas or words of others have been included,I have adequately and accurately cited and referenced the original sources.I also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Kollam

Date: 12-07-22

Christy Raj

DEPARTMENT OF COMPUTER APPLICATIONS

TKM COLLEGE OF ENGINEERING

C E R T I F I C A T E

This is to certify that, the project report entitled **TRAFFIC SIGN AND LIGHT DETECTION USING YOLOV5** is submitted by **CHRISTY RAJ (TKM20MCA-2016)** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications,is a bonafide record of the project work carried out by her under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor　　　　Head of the Department　　　　External Examiner

ACKNOWLEDGEMENT

First and foremost I thank GOD almighty and my parents for the success of this project. I owe sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

I am extremely grateful to **Dr. Fousia.M.Shamsudeen**, Head of the Department, for providing me with best facilities.

I would like to thank my project guide **Dr. Nadera Beevi s**, Department of Computer Applications, who motivated me throughout the work of my project.

I profusely thank all other faculty members in the department and all other members of TKM College of Engineering, for their guidance and inspirations throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this project.

Christy Raj

ABSTRACT

Detecting traffic signs and lights has been a problem for intelligent vehicles for a long time.Prior to classifying traffic signs and lights, this provides an efficient method for managing the inventory of traffic signs and lights for driver assistance or autonomous vehicles. Object identification models like Fast RCNN and Faster RCNN have been applied to this issue. These approaches' main drawbacks are their slowness and inability to sort in real time. Deep learning was used to create the Convolutional Neural Network (CNN) for visual object detection. This has helped to fix the problems with traditional object recognition. In this proposed plan, CNN and YOLO architecture have been chosen as ways to find and sort things. Here, the newest version of the YOLOV5 was used. The YOLOV5 is mostly used because it is fast, has a simple design, and works well for most object detection tasks. In this system, a method for detecting and identifying traffic signs and lights is proposed. This method, along with deep CNN, is used to classify traffic signs and lights. The proposed system works in real time to find and identify images of traffic lights and signs.

Contents

List of Figures

Chapter 1

Introduction

Statistics from the government show that more than 400 car accidents happen every day in India. Signs and lights on the road keep drivers and pedestrians safe by cutting down on traffic accidents. Since traffic signs also make sure that cars follow the rules, there are less likely to be traffic violations. Using traffic signs is another way to help people find their way. Make it a priority for all road users, including cars and people on foot, to follow traffic signals. People don't pay attention to traffic signals and lights for a number of reasons, such as being tired or not getting enough sleep. People's inability to read traffic signs and signals can be caused by a number of things, such as poor vision, outside influences, and the environment around them. Traffic Sign Recognition (TSR) uses digital photos or video frames to try to figure out what a traffic sign is. Most TSR methods use things you can see, like the colour and layout of traffic lights. In real-time tests, however, the standard TSR algorithms were shown to have problems. For example, they were easily affected by things like lighting, camera angle, roadblocks, vehicle speed, etc. Also, it can be hard to find more than one target because slow recognition makes it easy to ignore visual clues. As computer technology keeps getting better, the limitations of artificial neural networks have become much less of a problem. This has led to a golden age of development for machine learning. To solve all of these problems, many new machine learning techniques and algorithms have been made. TSR used to put a lot of importance on traditional ways of finding objects.

Following the application of manually crafted criteria to discover region proposals, the TSR pipeline typically makes use of classifiers to discover undesirable regions. Over the course of the past few years, the application of deep learning algorithms has made it significantly simpler to locate and recognise targets. Image recognition and object detection research frequently makes use of deep convolutional neural networks, abbreviated as CNNs. These networks are designed to improve precision, speed, and accuracy. Following that, CNN would be able to learn the features directly from the massive datasets, without the need for any preprocessing. Because of this, the requirement for manually crafted features would be removed, and more general features would be added. a prime period of growth and development.

Recent developments in object identification algorithms include SSD, Fast R-CNN, Faster RCNN, R-FCN, and YOLO all use CNN. Here use ”You Look Only Once” (YOLO) single-shot detection network, which has a short propagation delay and good detection performance. Although many contemporary neural networks are accurate, they do not operate in real time, making the utilization of numerous GPUs for training necessary. In the grid, each cell is in charge of finding objects within of it. Due to its accuracy and speed, YOLOv5 is one of the most well-known object identification algorithms. Real-time picture analysis via a front-facing camera in a car is made possible by thistechnology, which also alertsthe driver to potential problems through the vehicle's navigation system.

1.1 Problem Definition

Numerous inventive solutions for detecting traffic signs and lights have been proposed. The majority of them used straightforward models, like CNN, to resolve the problem, however the primary issues with these models are that they are impractical for real-time applications due to extensive computation, low performance, large costs in terms of memory and latencies, and a lengthy training process. The YOLOV5 object detection technique, which separates images into a grid system, can be used to tackle this issue. In the grid, each cell is in charge of finding objects within of it. Due to its accuracy and speed, YOLOv5 is one of the most well-known object identification algorithms. To achieve a high recognition rate and superior performance, convolutional neural networks are employed in conjunction with YOLOv5 architecture. Real-time object detection is provided by YOLOV5 using neural networks.

1.2 Objective

As the project's main goal, we want to:

- Develop a system that is able to detect, recognise, and categorise different traffic signals and signs as they occur in real time.
- The recognition of traffic signs is used to enforce the rules that govern traffic signs and to restrict certain behaviours.
- A quick, accurate, and reliable automatic traffic sign and light detection and recognition system may support and help the driver while also greatly enhancing comfort and safety while driving.

Chapter 2

Related Works

This section discusses numerous research of traffic sign and light identification methods that use deep learning.

2.1 Unified approach for detecting traffic signs and potholes on Indian roads [1].

It is necessary for the Driver Alerting System to be able to automatically locate and identify potholes and other forms of traffic signage. The ability to read traffic signs and locate potholes cannot be learned in a unified fashion at this time. The majority of research on pothole recognition and traffic sign interpretation is based on deep learning approaches, such as convolutional neural networks and long short-term memory, among others. The majority of work is also taking place along international routes, resulting in roads that are in substantially poorer condition than Indian roads. As a result of this research, a method is being developed to locate both potholes and traffic signs on Indian roadways.

The most useful characteristics of traffic signs are identified and paired with the assistance of the Hybrid Qualities, which are derived from the Accelerated Segment Test and the Random Sample Consensus methods. The accelerated segment test, which also provides a way for fast locating corners, makes use of the random sample consensus approach in order to get rid of any mismatched sites that may have been present. Both the enhanced Canny Edge detector and the Contour detection approach, which is modelled after the way in which potholes are discovered by nature, can be used to locate potholes. The Support Vector Machine classifier is then utilised to sort traffic signs and potholes into their proper groupings. The diameters of the potholes are then determined using a bounding box regression model. The suggested unified model surpasses previously developed models in terms of precision, sensitivity, specificity, the Matthews correlation coefficient, and F1-Score values, as determined by the results of the research. This research is leading to the development of a system for locating potholes and traffic signs on Indian roadways. It has been found that the FAST-RANSACSVM approach is superior to both the FasterRCNN method and the DR-CNN method when it comes to locating traffic signs. The MCC and F1-score values for FAST-distinct RANSACSVM are 70 percentage and 90 percentage, respectively, due to the fact that it is superior to other methodologies. In the case of the Faster-RCNN,it reaches a value between 65 and 88 percent. In the case of DR and CNN, it only reaches 49 and 83 percent respectively. It is not realistic to find things in real time because the dataset has few examples of potholes and traffic signs, the model takes a long time to execute, and it takes a long time to find things.

2.2 Deep learning for Large-scale Traffic-Sign Detection and Recognition [2].

For inventory management of traffic signs, the signs must be able to be found and recognised automatically. It makes the best use of human labour and is the most accurate and efficient way to keep track of the number of traffic signs. A great deal of research has been conducted in the field of computer vision on how to recognise and locate traffic signs. Most of the current ways to understand traffic signals meet the needs of complex driver assistance systems and systems that drive themselves.We don't know yet how well the remaining traffic signs will work when they are used to replace the time-consuming work of human traffic-sign inventory management.This is due to the fact that few traffic signs remain (about 50 categories out of several hundred). The problem of discovering and locating traffic sign categories that could be processed automatically was investigated. A convolutional neural network (CNN) technique with autonomous end-to-end learning is utilised for the entire detection and recognition procedure. It makes it easier to read traffic signs and improves the car's overall performance in a number of ways. This method figures out which of the 200 different kinds of traffic signs are in the new dataset. Results are given for very hard types of traffic signs that were left out of previous evaluations. The proposed method evaluates the deep learning method for recognising traffic signs that look different within the same category and have three percent error rates. It can be used in real-world traffic-sign inventory management systems. But this system's biggest problems are that it doesn't work well and uses a lot of memory

2.3 Real Time Embedded traffic sign recognition using Efficient convolutional neural network [3].

They discuss two networks in their analysis: an ENet network for TSC and an EmdNet network for TSD.They show the experiments that were done as the ENet framework was being made. The new ENet has 0.9M more parameters than the most common way of doing things. Using ideas from ENet, they also built the EmdNet backbone network. EmdNet was made with the SDD Framework. It has 6.3M parameters and is similar to MobileNet. The results of these experiments show that it is possible to build a neural network architecture that is accurate, general, and fast enough for embedded traffic sign recognition in real time. Most of the time, they have three long-term goals. First, performance needs to be improved, especially for TSD. The second is to feed the network with movies instead of photos to give it more physical proof. The third and fourth goals are to find out how multitasking affects learning and to make the network better at generalisation and commercialization.

2.4 Traffic Light Recognition with High Dynamic Range Imaging and Deep Learning [4].

Before determining the signal strength, the traffic light identification system, or TLR, first pinpoints the location of the traffic light in the image. TLR is necessary for autonomous vehicles since doing so raises the likelihood of a fatal collision. The computing time, the changeable lighting, and the risk for false positives are the three key issues that a practical TLR system needs to overcome. Their research presents a novel approach to the problem of real-time traffic light detection by utilising deep learning and extremely dynamic images. After been positively recognised in a series of low exposure and dark photos, a deep neural network reliably recognises candidates for traffic signals in high exposure and bright frames. This takes place once the candidates have earned significant recognition. This dual-channel technique can completely utilise the context information in light photos as well as the colour and shape information available in dark images.First, it was demonstrated that a non-parametric multicolor saliency model for the dark channel could simultaneously extract lights in various colour spectrums. On the bright channel, a CNN-based multiclass classifier is then employed to reduce the number of false positives. Temporal trajectory monitoring has significantly improved the overall performance. The number of sites that may be examined is decreased using a mask and a previous detection to speed up the process. This helps to reduce the overall search area that must be covered. Current best practises in object detection and real-time deep learning, which use only bright photographs and could produce more false positives, are outperformed by the suggested method. To ensure its accuracy, the suggested method was evaluated using a huge dual-channel dataset. Their self-driving car has a reliable algorithm that permits it to travel on public roads. The model has some shortcomings, such as a small sample size and average performance.

2.5 Multi-Feature Fusion and Enhancement Single Shot Detector for Traffic Sign Recognition [5].

Modern driver assistance systems rely a lot on being able to recognise and identify road traffic signs in order to give real-time data about how drivers see those signs (ADAS).MF-SSD offers an improved Single Shot Detector (SSD) approach for recognising traffic signs by combining and enhancing numerous aspects. The SSD's ability to locate small targets is first enhanced by a combination of low-level and high-level features. So that they can find the target, they improve the qualities of a number of channels by boosting the important qualities and reducing the less important ones.When utilised in the United States with real-time traffic indicators, the method is effective. The studies' results show that the MF-SSD algorithm is better at finding small traffic signs. Compared to traditional methods, it is more effective, reliable, and accurate at detecting problems in heavy traffic. In order to use it in real time, they will try to improve the algorithm and put the framework to use on the dataset of domestic traffic signs.

2.6 Deep Learning Traffic Sign Recognition in Autonomous Vehicle [6].

A deep learning technique is used to develop a system for recognising traffic signs. Because of its quick response in terms of real-time data reliability, high accuracy, and strong performance, You Only Look Once(YOLOv3) is used. In this study, image preprocessing is used to improve recognition system decision-making in a different environment that includes lighting and weather. It makes sure that the method is secure enough to be put into autonomous vehicles. There will be a comparison between the training and test-related photos. Create a system for autonomous vehicles without sensor fusion as future development.

2.7 An Improved Light-Weight Traffic Sign Recognition Algorithm Based on YOLOv4-Tiny [7].

When lightweight networks are required to recognise traffic signs, they tend to have poor detection accuracy and incorrect placement accuracy. To address these issues, an improved light-weight traffic sign identification method based on YOLOv4-Tiny was developed. For the traffic sign data set, the K-means clustering approach is improved, and a suitable anchor of the correct size is generated. This is intended to improve both the precision of target localization and the rate of detection recall. The large-scale feature map optimization method should be used whenever possible because it enhances the manner in which information regarding small target features is displayed by enhancing the level of network features with low-level data. The best course of action is this one. Additionally, it makes finding far-off and evasive targets much easier. They propose an enhanced NMS technique to filter the prediction box, prevent deleting the prediction results of different targets, and improve the accuracy and recall rate of target detection as a result of the model's post-processing stage omitting targets with significant overlap.This is due to the fact that targets with a lot of overlap were not found in the model's post-processing stage. The experimental results demonstrate that the improved algorithm's mAP and recall, when using the TT100K dataset, are superior to those of the original algorithm by 5.73 and 7.29 percentage points,respectively, when put through the same traffic sign recognition task asthe original YOLOv4-Tiny algorithm. This indicates that the standards for the task's accuracy and real-time performance have been reached. The model's most significant flaws can be summarised as a restricted capacity to extract features and an ineffective utilisation of multi-scale features.

2.8 Automatic Traffic Sign Recognition Artificial Intelligence Deep Learning Algorithm [8].

Convolutional neural networks were chosen because, in their opinion, they offer the best accuracy and usability when compared to other networks for computer vision applications. The main advantage CNN has over its forerunner is that it can identify features automatically and without human intervention. They gave the network a lot of photos of traffic signs in this case, and the algorithm found unique characteristics for each class of traffic sign. In their future research, they can make use of the dataset that includes more traffic sign classifications.

2.9 Fully Automated Traffic Sign Substitution in Real-World Images for Large-Scale Data Augmentation [9].

In this study, they show a way to change photos on traffic signs that is fully automated. Traffic sign images can be made with any symbol and can show a wide range of places and things in the real world. So, the case study can be used to show that there are many problems with recognising natural image data that have simple geometric constraints but look very different because of how they were recorded, how they were made, or because of natural events. In the following research, several questions about the generation process are looked into: The animations on CycleGAN aren't very good because the pictures aren't very clear or aren't very big (the smallest picture samples for substitution were just 2525 pixels in size). So, the overall generating strategy may no longer give satisfactory results, and the estimation of posture and segmentation of the background may also be wrong. If the currently allowed substitutes could be used between classes, the background/traffic sign combinations might be more varied, which would make for a more stable training dataset. It is important to think carefully about whether and how these problems can be fixed.

2.10 Traffic Sign Image Synthesis with Generative Adversarial Networks [10].

They offer a novel method for visual synthesis for traffic signs in this research. For this, a generative network was built using a typical traffic sign template as input and a portion of a background image. The backdrop picture patch in the architecture controls the lighting and visual appearance of the fake traffic signs. To create realistic visuals, this network has been trained using the GAN framework. Experimental results show that their approach, when applied to a limited quantity of real labelled training data, not only generates more realistic images than the traditional approach, but also yields improved classification results. Models' greatest flaw is their time commitment.

2.11 Weakly Supervised Traffic Sign Detection in Real Time Using Single CNN Architecture for Multiple Purposes [11].

They present a novel unsupervised method for finding traffic signs in this study, based on a single convolutional neural network. The classification network is based on MobileNetv2, a very lightweight convolutional architecture that is also employed as a region proposal network.The approach consists of two steps: in the first, MobileNetv2 is asked to recommend which components of a photograph should be identified, and in the second, it is asked to classify traffic signs. At a resolution of 800x1300 milliseconds, it took around 55 milliseconds to accurately process a single image or frame. This strategy eliminates the time that would have been required for dataset annotation when poor supervision is employed. For recognising and detecting traffic signs, the German Traffic Signs Recognition Benchmark (GTSRB) and the German Traffic Signs Detection Benchmark were two well-known datasets.These datasets were used to train and evaluate our proposed technique (GTSDB). In the future, they will be able to use anchor boxes to improve memory, accuracy, and the mAP overall.

Chapter 3

Methodology

3.1 Proposed System

To overcome the limitations of current object detection approaches, images are identified using the YOLOv5 ultralytics open-source object-identification model based on deep learning. YOLO, a convolutional neural network (CNN), is used to identify objects in real-time. It may overcome the difficulty of detecting varied image scales, so boosting the effectiveness of traffic sign and light detection. The current method employs anchor boxes better suited for detection and regression loss functions that are more precise. The method applies a single neural network to the entire image, divides it into areas, and then predicts the bounding boxes and probabilities for each region using a neural network. The proposed model can be employed to detect the item by integrating with laptop cameras. However, deep learning frequently requires a significant quantity of labelled data for training, which can be time-consuming. Transfer learning is a typical strategy for addressing this difficulty, in which a model that has been trained for one task on an existing labelled database is updated for use in a different, unrelated, but still relevant activity. Yolov5's model for autonomously categorising traffic signs is based on transfer learning-derived applications of deep learning and convolution neural networks. The block diagram of the proposed system is depicted in Figure 3.1.

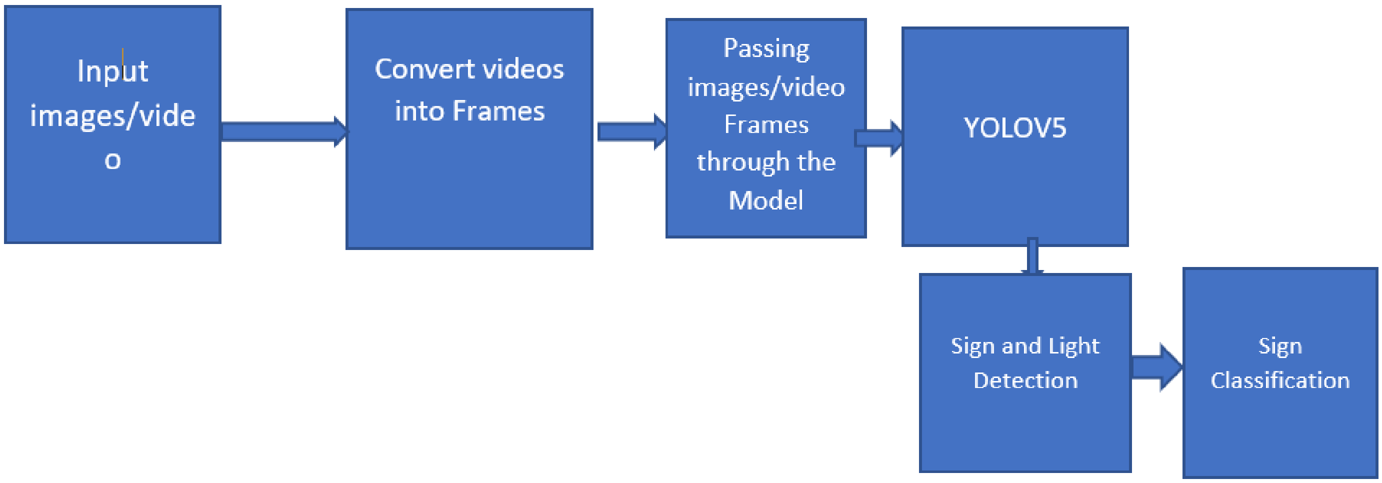

Figure 3.1: Schematic of proposed system

3.2 System Architecture

The basic objective of the project is to locate traffic lights and signs. The traffic lights and signs dataset can be trained using this method. After training the model, the system can recognise signs and lights and classify them. It may even access the webcam and predict future events. Figure 3.2 depicts the system's architecture. The suggested system comprises four significant phases:

- Preparation & Pre-Processing of Dataset.
- Selecting a pre-trained model of yolov5.
- Training the model.
- Testing the model.

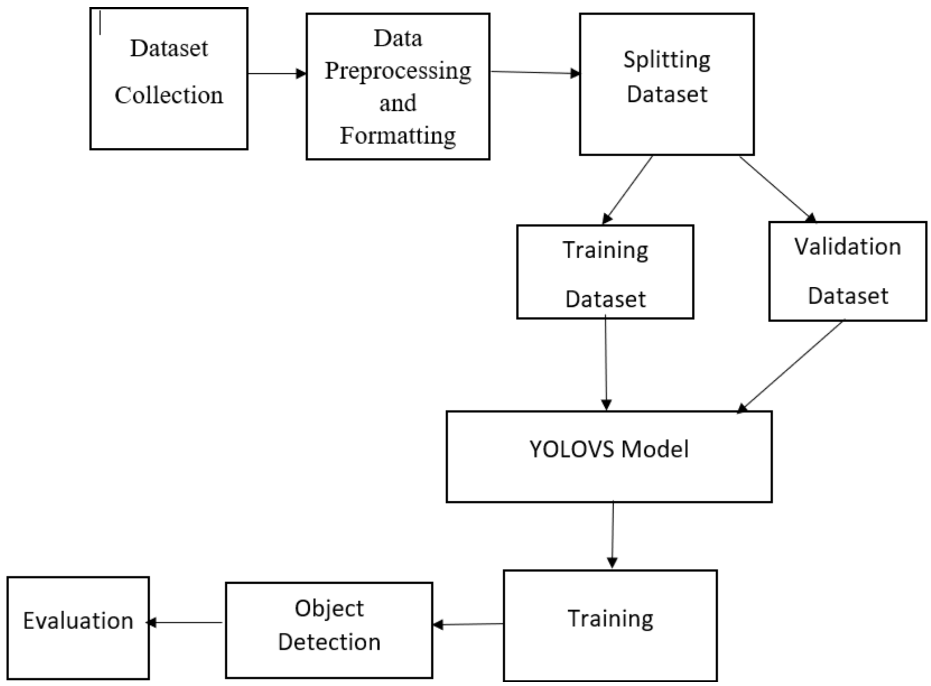

Figure 3.2: System Architecture

3.3 Dataset

The dataset for traffic signs is taken from the analytics Vidhya website. Using this dataset, a model may be created to correctly identify and categorise the traffic sign. This dataset is in YOLO format and contains 1144 photos that are divided into 77 classes. The courses include: Downloading the data is possible at https://www.analyticsvidhya.com/blog/2021/12/how-touse-yolov5-object-detection-algorithm-for-custom-object-detection-an-example-use-case/. The Traffic Light dataset is the coco dataset, as well as the dataset is in YOLO format and it consist of 80 classes.

3.4 Data Preparation and Preprocessing

We preprocess the data before putting it into our model since the data's quality and the knowledge that may be drawn from it directly affect how well our model learns. The process of transforming raw data into something acceptable for our model is known as data preprocessing. It is the first and most crucial step in the process of creating a machine learning model. To acquire better outcomes from the applied model, the data format in machine learning projects must be accurate. To prepare for model training, we must first build the project coco128.yaml file,

```
train: /content/drive/MyDrive/Dataset/images/train #
val: /content/drive/MyDrive/Dataset/images/val  # va
test:  # test images (optional)

# Classes
nc: 77  # number of classes
names: ['200m',
        '50-100m',
        'Ahead-Left',
        'Ahead-Right',
        'Axle-load-limit',
        'Barrier Ahead',
        'Bullock Cart Prohibited',
        'Cart Prohobited',
        'Cattle',
        'Compulsory Ahead',
        'Compulsory Keep Left',
        'Compulsory Left Turn',
        'Compulsory Right Turn',
        'Cross Road',
        'Cycle Crossing',
        'Compulsory Cycle Track',
        'Cycle Prohibited',
        'Dangerous Dip',
        'Falling Rocks',
        'Ferry',
        'Gap in median',
        'Give way',
        'Hand cart prohibited',
        'Height limit',
```

Figure 3.3: coco128.yaml file

describing the locations of the training and validation images, the number of labels, and the names of the labels from our training set. The file should be organised in the manner depicted in figure 3.3.

3.5 Selecting a pre-trained model of yolov5

are a number of YOLOv5 architectures, known as P5 models, that differ primarily in terms of their parameter sizes: YOLOv5n (nano), YOLOv5s (small), YOLOv5m (medium), YOLOv5l (large), and YOLOv5x (extra large). Using 640x640 pixel pictures for training, these architectures perform well. We choose the YOLOV5s version because it is the most straightforward. With only 7.2 million parameters, it is the smallest model in the Yolov5 family and is best suited for inference on the CPU.

3.6 Training the model

Transfer learning is another technique for deep models, particularly CNNs, to improve performance. Transfer learning is the idea of moving past the isolated learning paradigm and applying the skills you've acquired to tackle one problem to others that are similar. The YOLO v5-s model is modified for the training procedure, and the results of the training are applied to the detection procedure. The little model's layers have all been trained in this manner. This suggests that even if the pretrained weights will be loaded, the entire model will be adjusted on a fresh dataset. The model is saved as a pt file after training, and the results of the training can be verified using images or videos.

3.7 Testing the model

Training data and testing data have been created from the selected data. A deep learning model was created using the training data, and it was validated using the testing data. The model, which was created using training data, has been put to the test using test data in order to confirm its accuracy-based performance. Using the test dataset, the created model has been put to the test at this step. Similar to how it processes training data, the system also processes test data. Photos or videos are used in the testing process. For testing, images or videos can be used as input. The built-in model is loaded to start the detection phase, and then bounding boxes and confidence scores are used to start the classification process. Prediction boxes, confidence values, and object classes are displayed as the results. The system classes the object based on that (signs).

3.8 YOLOV5 Architecture

Object detection refers to the process of locating and detecting certain items inside an image. It is one of the primary jobs in computer vision. The YOLO (You Only Look Once) algorithm was created in 2015 with a new angle, and it completes object detection in a single neural network. That led to an explosion in the field of object detection, which led to much more astounding results than in the previous decade. YOLO has so far undergone five iterations of improvement and is considered one of the best object detection algorithms. YOLOv5, also known as YOLOv5, is the most current version of YOLO that was not created by the original author. However, the YOLOv5 outperforms the YOLOv4 in terms of accuracy and speed.

The YOLO model family consists of three primary architectural components.

- Backbone
- Neck
- Head

YOLOv5 Backbone: Key features are extracted from an input image using the backbone model. The CSP (Cross Stage Partial) Network serves as the framework for YOLOv5 to extract useful characteristics from the input image. In the feature extraction procedure, DenseNet CSP is used. DenseNet's foundation is the same for CSP (Cross Stage Partial), with the exception that CSP input is split into two parts, one of which proceeds directly to the subsequent stage without being processed and the other of which passes via dense blocks that perform convolution. After then, the block's convolutional component will be integrated (concat). The SPP (Spatial Pyramid Pooling) layer is one of the additional blocks that are added before employing FPN to enter the neck.

YOLOv5 Neck: Using the Feature Pyramids Network-powered YOLOv5-based PANet, neck style (FPN). The PANet model is used to enable the model's successful generalisation of object scaling. When identifying the same thing in various scales and sizes, this is incredibly helpful. FIt uses PANet to combine the information and send them to Head for prediction.

YOLOv5 Head: The head model completes the final detection by applying anchor boxes to the features and generating an output vector containing bounding boxes, objectivity scores, and class probabilities. Leaky ReLU and Sigmoid are the activation techniques used by YOLOv5. In the middle layer, the Leaky ReLU activation function is used, and Sigmoid is used in the final detection layer. The number of input weights and biases required to activate and deactivate neurons is determined by the artificial neural network's activation function. YOLOV5 uses the objectivity, class probability, and bounding box regression scores to determine the loss value. The architecture of YOLOV5 is shown in Figure 3.4.
Additionally, YOLOv5 makes advantage of the following training methods:

Activation and Optimization: 1.Activation and Optimization: Leaky ReLU and sigmoid activation, as well as SGD and ADAM as optimizer choices, are used in YOLOv5.
2. Loss Function: The loss function consists of binary cross-entropy with logits loss.

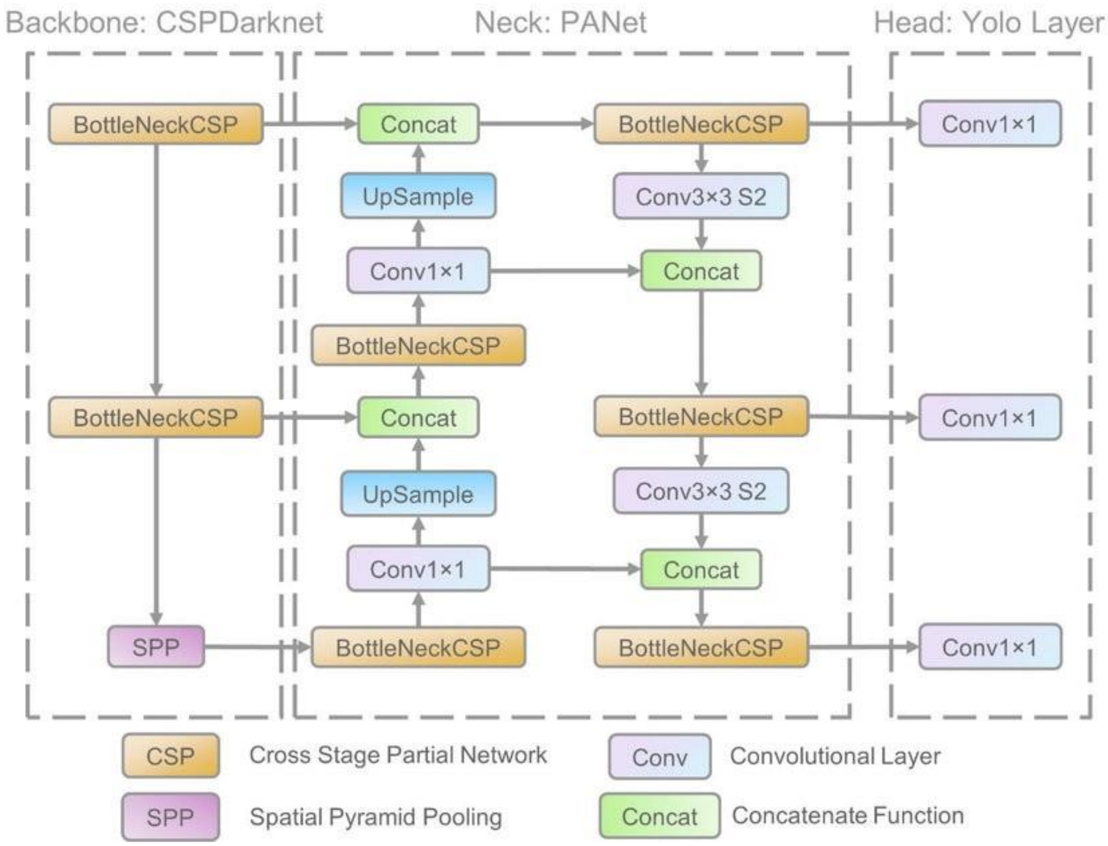

Figure 3.4: YOLOV5 Architecture

3.9 Software Requirement

The software used for the project:

• Python

• Google Colab

• Juypter notebook

1. Python

Python is an interpreted, general-purpose programming language. Python was invented by Guido van Rossum and made available to the public for the first time in 1991. Its design philosophy effectively utilises large amounts of whitespace and lays a major emphasis on code readability. Its object-oriented technique and language features Assist programmers in writing comprehensible code for both small and large projects. Python is equipped with garbage collection and dynamic typing. There is support for procedural, object-oriented, and functional programming paradigms. Python is referred described as ”batteries included” due to its extensive standard library. A garbage collector that detects cycles and Unicode support were two of the many notable new features added to Python 2.0 on October 16, 2000. Python 3.0’s release on 3.12.2008. Significant modifications were made to the language, some of which are not totally backward-compatible. Many of Python’s most crucial features were backported to the 2.6.x and 2.7.x version series. The Python 3 releases feature the 2 to 3 utility, which automates the conversion of Python 2 code to Python 3 at least partially. Python is designed to make reading simple. Its formatting is aesthetically beautiful, and English words are frequently substituted for foreign language punctuation. Similar to many other languages, it does not utilise curly brackets to separate blocks, and semicolons are optional between sentences. It contains fewer syntactic exceptions and special circumstances than C or Pascal. In 1999, Guido van Rossum described his goals for Python:

- Easy and intuitive language just as powerful as those of the major competitor
- Everyone can contribute to its development because it is open source.
- Code that is as easy to understand as simple English

Python was designed to be extremely readable. It uses English terms more often than punctuation and has fewer syntactical elements than other languages. Especially if they work in the web development industry, Python is a fundamental requirement for students and working professionals who wish to become exceptional software engineers. Knowing Python has a number of important advantages, including:

- Python is interactive: when creating programmes, you can sit at a Python prompt and immediately have a discussion with the interpreter.

- Python is Interpreted: At runtime, Python is processed by the interpreter. Your software does not require compilation prior to execution. Similar to PERL and PHP
- Python is Object-Oriented :Python allows object-oriented programming, in which code is contained within objects.
- Python is a Beginner's Language: Python is a fantastic option for novice programmers since it enables the building of a wide range of programmes, from simple text processing to game development to web browsers and the World Wide Web.

2. Google Colab

To provide free access to GPUs and TPUs for anyone who needs it to construct a machine learning or deep learning model, Google created Google Colab. Google Colab is a more sophisticated variation of Jupyter Notebook. A application called Jupyter Notebook allows users to edit and run Notebook documents using either an Integrated Development Environment or a web browser (IDE). Google Colab has all the intriguing features that every modern IDE offers, plus more.

Below is a list of some of the more fascinating aspects.

- Interactive tutorials for learning neural networks and machine learning.
- Without a local setup, write and run Python 3 code.
- Run commands on the terminal from the Notebook. Bring in data from outside sources like Kaggle.
- Your notebooks should be saved to Google Drive.
- Google Drive notebook imports.
- No-cost cloud computing, GPUs, and TPUs.
- Integrate with Open CV, PyTorch, and Tensor Flow.
- Directly import or publish to/from GitHub.

3. Jupyter Notebook

   Fernando Perez created Jupyter notebook as a web-based user interface for the IPython kernel. In an effort to provide an integrated interactive computing environment for many languages, the Notebook project has been redirected to Project Jupyter, which provides a front end for the programming environments Juila and R in addition to Python. You can discover rich text elements in notebook papers, such as images, equations, and HTML-formatted text. A Python or other auxiliary language-written executable file is also included in the notebook. Jupyter Notebook is a prime example of client-server software. The server runs on the local computer, while the software launches the notebook interface in a web browser so that it can be modified and utilised. It is saved as an ipynb file and exported as html, pdf, and LaTex files.

ADVANTAGE OF JUPYTER NOTEBOOK

- In a single location: Jupyter Notebook is a web-based interactive environment that mixes code, text, photos, videos, mathematical equations, charts, and widgets in a single document. You may be familiar with it already.
- Simple to convert: Users of Jupyter Notebook can convert their notebooks to HTML or PDF formats with ease. Additionally, it utilises internet tools such as nbviewer, which renders a publicly accessible notebook directly in the browser.
- Simple to share: Because Jupyter Notebooks are saved as structured text files, they are straightforward to share (JSON format).
- Language independence: Jupyter Notebook is platform-neutral and language-independent due to its representation in the text-based file format JSON (JavaScript Object Notation). The notebook can be processed by any programming language and converted to multiple file formats, including Markdown, HTML, and PDF.

- Interactive code: Jupyter Notebook employs the ipywidgets packages, which provide users with a variety of standard user interfaces for viewing interactive code and data.

DISADVANTAGE OF JUPYTER NOTEBOOK

- Testing long asynchronous jobs is difficult.
- Security is Less
- It runs cell out of order
- Lack of IDE integration, linting, and code-style correction in Jupyter notebook.

Chapter 4

Results and Discussion

The findings of the generated model will be discussed in this section. The model correctly classifies traffic signs after detecting the sign and light.

4.1 Training and Validation results

Following setup, the model needs to be trained. The fundamental foundation is based on the YOLOv5 repository. All YOLOv5 repositories and dependencies must be satisfied. On 1144 annotated photos, dataset training for the traffic sign detection system was done. The training outcome is depicted in figure 4.1.

```
Fusing layers...
Model summary: 213 layers, 7217794 parameters, 0 gradients, 16.4 GFLOPs
                 Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 5/5
                   all        153        333      0.771      0.666      0.701      0.481
           Ahead-Right        153          3      0.642      0.617      0.665      0.551
     Compulsory Ahead        153         14      0.649      0.714      0.817      0.576
  Compulsory Keep Left        153          2          0          0     0.0387      0.031
  Compulsory Left Turn        153          1      0.778          1      0.995      0.298
              Give way        153         20      0.881          1      0.995       0.67
          Height limit        153          1          1          0          0          0
            Humpy Road        153         11      0.759          1      0.995      0.708
    Left hair pin bend        153          1          1          0          0          0
       Left hand curve        153          2          1          0     0.0585     0.0319
           Men at work        153         11      0.801      0.636      0.813      0.593
     Narrow road ahead        153          3      0.607          1      0.995      0.863
            No stoping        153         39       0.84      0.923      0.942      0.707
          One way sign        153          1          1          0          0          0
   Pedestrian crossing        153         46       0.69      0.957      0.887      0.634
      Right hand curve        153          3      0.912      0.667      0.727      0.512
    Right Reverse Bend        153          1          1          0    0.00454    0.00318
          School ahead        153         12       0.97      0.833      0.934      0.496
        Slippery road        153          3      0.342          1      0.995      0.632
           Speed limit        153         82      0.924      0.988      0.982      0.706
                  Stop        153          3      0.643          1      0.995       0.83
Compulsory turn left ahead        153          2          1          0      0.663      0.448
Compulsory right turn ahead        153          6      0.768       0.56      0.637       0.42
                Sign_C        153          2      0.503          1      0.995      0.697
                Sign_T        153          7      0.461          1      0.853      0.563
                Sign_S        153         17      0.718      0.824      0.788      0.457
              No entry        153         13      0.947          1      0.995      0.748
```

Figure 4.1: Training Result

4.2 Performance Metrices

The model's precision, recall, F1, and mean Average Precision scores are shown in the classification report visualizer. The measurements are made using false negatives and genuine positives. In this instance, the terms ”positive” and ”negative” refer to the classes in a binary classification problem. We would regard true and false as both occupied and vacant in the previous scenario. When the projected class and the actual class are both positive, it is said to be a true positive. When the estimated class is positive but the actual class is negative, a false positive occurs. The subsequent conditions, which are listed below, are used to define the metrics:

- Precision

  The capacity of a classifier to avoid classifying something as positive that is truly negative is known as precision.It is defined as the ratio of true positives to all true and false positives for each class.

  Precision = Positive prediction accuracy

$$Precision = TP/(TP + FP) \tag{4.1}$$

- Recall

  The recall of a classifier is its ability to locate each successful occurrence. It is defined as the ratio of true positives to the sum of true positives and false negatives for each class. What portion of all cases that were actually positive were appropriately classified? is another way to phrase this inquiry.

  Recall = Amount of positives that were actually identified as positives

$$Recall = TP/TP + FN \tag{4.2}$$

- F1 Score

  The F1 score is calculated as a weighted harmonic mean and ranges between 0.0 and 1.0 based on recall and precision. Since precision and recall are taken into account when calculating F1 scores, they frequently perform worse than accuracy measurements. When comparing classifier models, the weighted average of F1 is typically recommended rather than overall accuracy.

  f1Score = what proportion of correct positive predictions there were

$$f1Score = 2(Recall * precision)/Recall + Precision. \tag{4.3}$$

- mAP

  mAP (Mean Average Precision) is a recall and accuracy combined measure that is used to determine object correctness. Calculated over recall ranges of 0 to 1, the average precision values are averaged out.

Considering the accuracy of the produced model Precision is the ratio of positive results actually found to the positive outcomes predicted by the classifier. Figure 4.2's precision graph demonstrates that the model is operating effectively as the precision value rises throughout the training period.

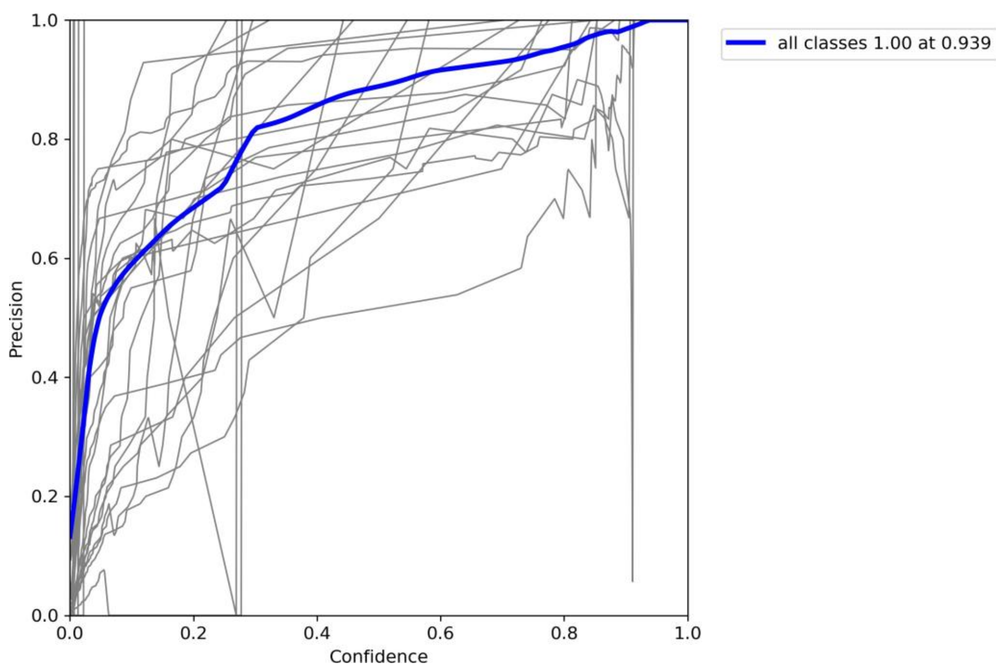

Figure 4.2: The precision of the created model

The recall value of the created model is displayed in figure 4.3. The recall in this instance is the proportion of correctly positive results to all samples that were determined to be positive.

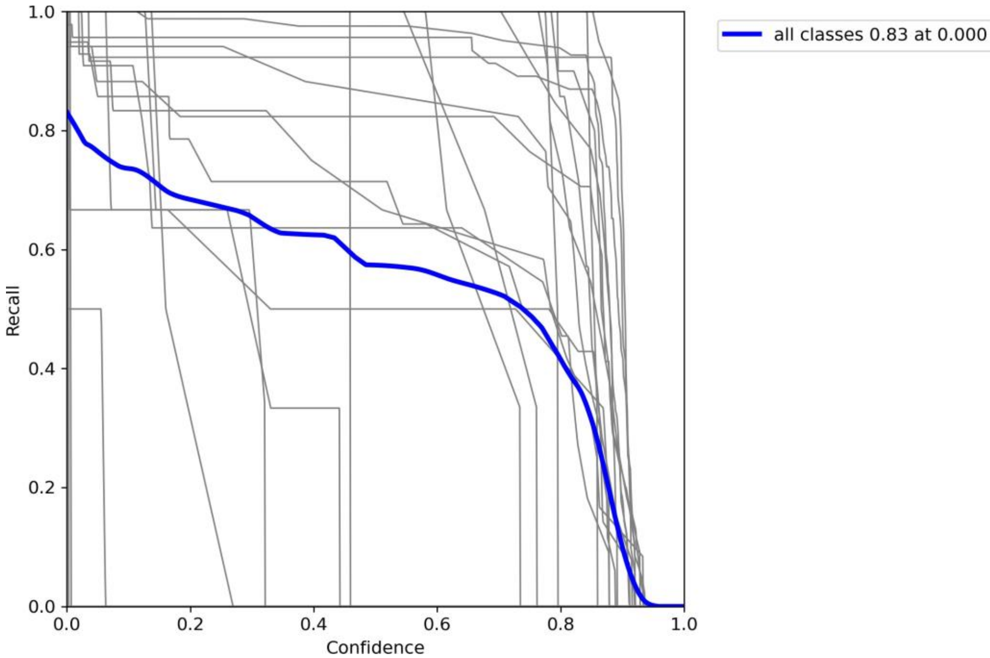

Figure 4.3: Recall graph of the developed model

The correctness of the created model can be evaluated using the precision Recall graph. The tradeoff between precision and recall for various thresholds is depicted by the precision-recall curve. High recall correlates with a large area under the curve, and high precision connects with a low false positive rate, yielding a low false positive rate. The Precision Recall graph for the created model is depicted in Figure 4.4.

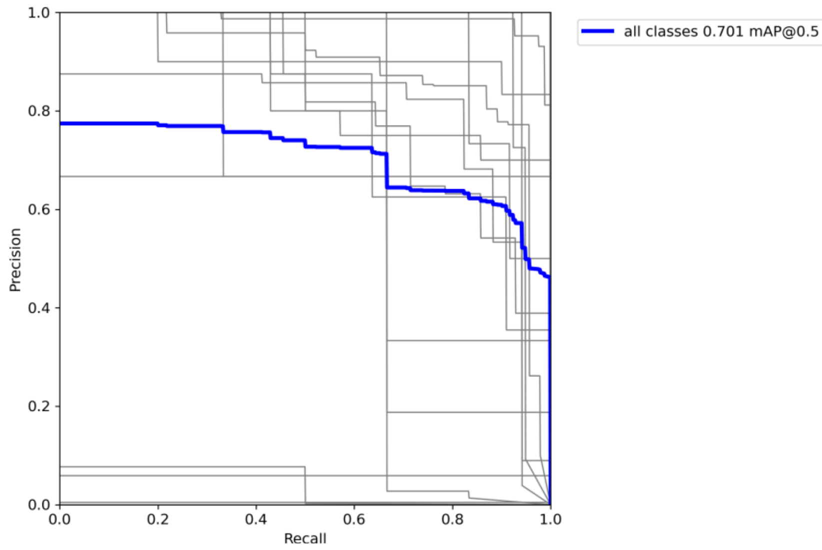

Figure 4.4: Precision Recall graph of the developed model

4.3 Confusion Matrix

A confusion matrix indicates that a classification model, also known as a ”classifier,” typically performs well on a set of test data for which the true values are known.It makes it possible to assess an algorithm’s effectiveness. It makes it simple to identify miscommunication brought on by lessons. For instance, it happens often that one class be categorised as the other. The confusion matrix is used to compute the majority of performance indicators. The confusion matrix’s list of notable terms includes the following:

- True Positive (TP): Both the observation and the prediction are positive.
- False Negative (FN): Positive observation with a predicted negative result.
- True Negative (TN): The observation and prediction are both negative.
- False Positive (FP): When an observation is negative but a positive result is predicted.

The developed model's confusion matrix is displayed in figure 4.5.

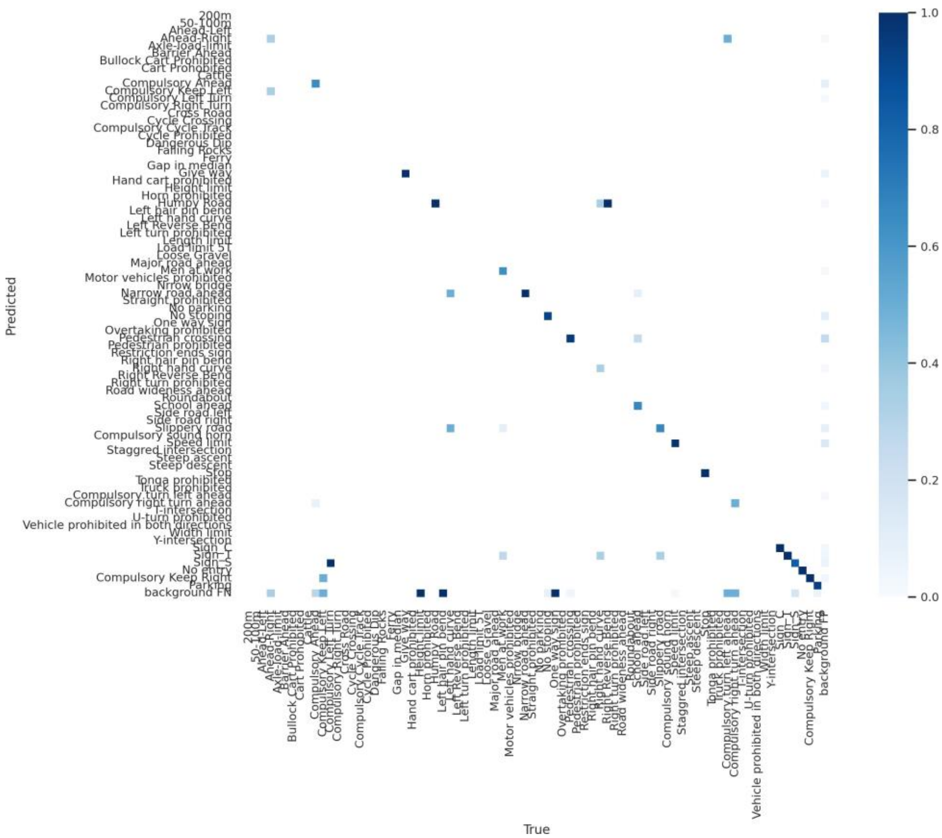

Figure 4.5: Confusion Matrix

4.4 Screenshots

The screenshots of developed model shown as from figure 4.6 to figure 4.10

Figure 4.6: Speed Limit

Figure 4.7: Pedistrian Crossing

Figure 4.8: No Parking

Figure 4.9: Overtaking Prohibited

Figure 4.10: Traffic Light

Chapter 5

Conclusion

The number of automobiles on the road has significantly grown recently. Due to various factors, including this extraordinary expansion, there are more accidents every year. Ignorance of traffic signs and lights is thought to be a major factor in these incidents. By utilising only the vehicle's camera, this system proposes a novel approach based on YOLOV5 enabling applications to detect traffic signs and lights.

5.1 Advantages

The key Features are :

- The model accurately detect the traffic signs and lights and also it efficiently classify the traffic signs .
- Faster Inference speed

5.2 Future Enhancement

We will keep expanding our datasets until we have all of the classifications of traffic signs covered. While this is going on, visual object recognition can be done using more modern models like Mask R-CNN and CapsNet. An effective class of traffic signs with spatial correlations has been identified using a capsule neural network (CapsNet). Deep neural networks place a greater emphasis on the topological relationship between visual components than capsule networks. In the future, we will employ expert evaluation techniques to assess the efficacy of our models from a variety of angles.

References

[1] Satish Kumar Satti, Suganya Devi K., Prasad Maddula, N.V.Vishnumurthy Ravipati, Unified approach for detecting traffic signs and potholes on Indian roads, Journal of King Saud University - Computer and Information Sciences, 2021,https://doi.org/10.1016/j.jksuci.2021.12.006.

[2] D. Tabernik and D. Skočaj, ”Deep Learning for Large-Scale Traffic-Sign Detection and Recognition,” in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 4, pp. 1427-1440, April 2020, doi: 10.1109/TITS.2019.2913588.

[3] X. Bangquan and W. Xiao Xiong, ”Real-Time Embedded Traffic Sign Recognition Using Efficient Convolutional Neural Network,” in IEEE Access, vol. 7, pp. 53330-53346, 2019, doi: 10.1109/ACCESS.2019.2912311.

[4] J. -G. Wang and L. -B. Zhou, ”Traffic Light Recognition With High Dynamic Range Imaging and Deep Learning,” in IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 4, pp. 1341-1352, April 2019, doi: 10.1109/TITS.2018.2849505.

[5] Jin Y, Fu Y, Wang W, Guo J, Ren C, Xiang X (2020)Multi-feature fusion and enhancement single shot detector for traffic sign recognition. IEEE Access 8:38931–38940.

[6] 2. S. M. Alhabshee and A. U. b. Shamsudin, ”Deep Learning Traffic Sign Recognition in Autonomous Vehicle,” 2020 IEEE Student Conference on Research and Development (SCOReD), 2020, pp. 438-442, doi: 10.1109/SCOReD50371.2020.9251034.

[7] L. Wang, K. Zhou, A. Chu, G. Wang and L. Wang, ”An Improved Light-Weight Traffic Sign Recognition Algorithm Based on YOLOv4-Tiny,” in IEEE Access, vol. 9, pp. 124963-124971, 2021, doi: 10.1109/ACCESS.2021.3109798.

[8] M. D. RADU, I. M. COSTEA and V. A. STAN, ”Automatic Traffic Sign Recognition Artificial Inteligence - Deep Learning Algorithm,” 2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), 2020, pp. 1-4, doi: 10.1109/ECAI50035.2020.9223186.

[9] . D. Horn and S. Houben, ”Fully Automated Traffic Sign Substitution in Real-World Images for Large-Scale Data Augmentation,” 2020 IEEE Intelligent Vehicles Symposium (IV), 2020, pp. 465-471, doi: 10.1109/IV47402.2020.9304547.

[10] H. Luo, Q. Kong and F. Wu, ”Traffic Sign Image Synthesis with Generative Adversarial Networks,” 2018 24th International Conference on Pattern Recognition (ICPR), 2018, pp. 2540-2545, doi: 10.1109/ICPR.2018.8545787.

[11] H. Ibrahem, A. Salem and H. S. Kang, ”Weakly Supervised Traffic Sign Detection in Real Time Using Single CNN Architecture for Multiple Purposes,” 2020 IEEE International Conference on Consumer Electronics (ICCE), 2020, pp. 1-4, doi: 10.1109/ICCE46568.2020.9042974.