PROJECT REPORT

“ PlateSentry - Licence Plate Recognition System ”

INDEX

List of Figures and Images

CHAPTER 1 - ABSTRACT

A licence plate recognition (LPR) system is an automated form of mass surveillance that reads and identifies car licence plates using a variety of digital image processing (DIP) and optical character recognition (OCR) techniques. In real-world applications such as access control, traffic enforcement, inventory and real estate management, security system monitoring, parking space allocation, and traffic monitoring, it has produced some positive results. This research project is mainly focused on the licence plate image extraction and licence plate localisation for possible use in different contexts. To achieve this, OpenCV digital image processing (DIP) techniques written in Python are used to generate an image segmentation against which the labels of some image segments were tested. The length of signs with similar characteristics found in each section is the key to locating and delineating the area using the actual licence plate of the vehicle. The licence plate attributes include image height, width, aspect ratio and other features used to extract a potential licence plate. Using these pixels, it was possible to locate the area of the image that contained the disk, filtering out unnecessary contours and curves appearing in noise.

Number plate recognition systems are used today in various traffic and security applications, such as parking, traffic and border control or tracking stolen cars. To date, all LPR systems have been designed using neural networks. This work proposes to run the system using preprocessing, plate localisation and OCR (Optical Character Recognition).

CHAPTER 2 - INTRODUCTION

Automated number plate recognition (ANPR) is an advanced technology and being used more and more in a variety of fields, including traffic management, parking administration, and law enforcement. The system is made to instantly scan and identify licence plate information from still or moving photos taken by cameras. The licence plate data is extracted and analysed from the collected images by ANPR technology using a combination of image processing, pattern recognition, and machine learning approaches.

The ANPR system is divided into various steps, such as image capture, preprocessing, plate extraction, character segmentation, and character identification. The licence plates of the vehicles are photographed using a camera during the picture acquisition step. The preprocessing stage is used to improve the acquired images' quality and get rid of noise and distortion that can interfere with recognition. The licence plate region is separated from the rest of the image during the plate extraction step, and during the character segmentation stage, the individual characters on the plate are separated. Machine learning techniques are utilised to distinguish and extract the characters' information from the segmented characters at the character recognition step.

ANPR technology has a wide range of possible uses and has already been put to use in a number of contexts, including law enforcement, parking management, and traffic surveillance. The technology can assist in enforcing parking regulations, reducing traffic congestion, preventing accidents, and identifying stolen or suspect vehicles. ANPR has shown to be a useful instrument for boosting public safety, surveillance and security, and it is anticipated that further development will result in even greater advantages in the future.

CHAPTER 3 - LITERATURE REVIEW

The expanding affluence of urban India has made automobile ownership a requirement. This has created an unforeseen civic issue: traffic control and vehicle identification. **[1]** Parking lots have gotten overcrowded as the number of vehicles on the road has increased. The Automatic Number Plate Recognition System (ANPR) is useful in tackling these challenges because its applications span from parking entry to monitoring urban traffic and tracking automotive thefts. With the explosive growth in the number of vehicles in use, automated licence plate recognition (ALPR) systems are required for a wide range of tasks such as law enforcement, surveillance, and toll booth operations. The operational specifications of these systems are diverse due to the differences in the intended application. There are several ANPR systems available today which are based on different techniques and methods.

I. Edge Based Technique :

Due of the distinctive colour and rectangular shape of licence plates, edge-based algorithms are frequently utilised for licence plate detection. Canny filters are used to identify horizontal and vertical edges, but this method is noise-sensitive and subject to erroneous detection. **[2]** Several approaches, like block-based algorithms and candidate rectangle detection utilising vertical edges, have been suggested to increase accuracy. For extracting licence plates, edge-density mapping and line-grouping algorithms have also been created. The standard method for locating straight lines is the Hough transformation, however it can be time-consuming and sensitive to border deformations. For quicker and more accurate results, some research have combined the Hough transformation with contouring algorithms. Although edge-based techniques are quick and easy, they are unsuitable for complicated and fuzzy images.

II. Colour Based Technique :

Colour-based techniques for detecting licence plates rely on the fact that the colour combination of each plate is distinct from the background and that licence plates stand out from it in terms of colour. Although it can be used for categorisation, the Hue, Lightness, and Saturation (HLS) colour model is sensitive to noise. **[2]** To increase accuracy, other techniques have been suggested, including mean shift segmentation and Gaussian Weighted Histogram Intersection (GWHI). These techniques, however, are sensitive to variations in lighting and may give false findings if other parts of the image have colours that are similar to those on the licence plate. As a result, they are frequently combined with other methods to produce precise findings.

III. Deep Learning Technique :

In recent years, deep learning techniques have become increasingly popular for the detection and identification of licence plates. The most popular deep learning models for detecting licence plates are convolutional neural networks (CNNs). CNNs have demonstrated promising results in the identification and recognition of licence plates by automatically learning features from the input image. For end-to-end licence plate recognition, CNNs have been employed in a number of experiments. In these techniques, the CNNs are trained to identify both the characters and the location of the licence plate. Several convolutional layers, pooling layers, and fully linked layers are typically used in these techniques. For optimal performance, they need a lot of training data.

Using object detection models in deep learning is another strategy for detecting licence plates. For the purpose of detecting licence plates, object detection models like **[3]** YOLO, SSD, and Faster R-CNN are frequently employed. These models are able to identify the area around the plate and produce a bounding box around it. They are capable of real-time performance and can be trained using a relatively little dataset. Deep learning techniques have proven to be resistant to changes in lighting, perspective, and occlusion. They work well in real-time applications and can manage hazy and poor-quality photos. Deep learning models, however, need a sizeable amount of training data and computing power. Moreover, they could be prone to overfitting and struggle to generalise to new data. Modules involved in system are -

I. Capture Frames :

The first step is to capture an image of the vehicle's licence plate using a camera. This image will be used as input for the licence plate recognition system. This can be done using a camera, either mounted on a stationary point or on a moving vehicle. The camera should be positioned in a way that captures a clear and unobstructed view of the licence plate. The video input to the system could also work in absence of camera devices.

II. Image Preprocessing:

The captured image may contain noise, blur, or other distortions that can affect the performance of the licence plate recognition system. **[4]** Therefore, image preprocessing is performed to improve the quality of the image. This may include techniques such as noise reduction, image sharpening, and contrast enhancement. The term image pre-processing means modify images such that they are either correction from errors introduce during acquisition or transmission. Pre-processing images commonly involves removing low-frequency background noise, normalising the intensity of the individual particles images, removing reflections, and masking portions of images.

III. Plate Extraction:

The licence plate region needs to be extracted from the preprocessed image. This can be done using techniques such as edge detection, morphological operations, or object detection algorithms. The extracted plate region is then cropped and resized to a fixed size to ensure consistency across all images.

IV. Character Segmentation:

The characters on the licence plate need to be segmented so that they can be individually recognised. This can be done using techniques such as connected component analysis, contour detection, or template matching. The segmented characters are then separated and aligned to form a sequence.

V. Character Recognition:

The final step is to recognise the characters in the licence plate. This can be done using a deep learning model such as a convolutional neural network (CNN) or a recurrent neural network (RNN). The model is trained on a dataset of labeled licence plate images and their corresponding characters. The recognised characters are then combined to form the complete licence plate number.

CHAPTER 4 - PROBLEM STATEMENT

To develop a system that can successfully and accurately identify licence plates from photos or videos taken by cameras. The system must be able to function in real-time, cope with difficult situations, and identify licence plates from various nations, regions, and font types. The system should make use of sophisticated machine learning and computer vision techniques. The user interface needs to be simple to use and maintain, enabling quick access to system features including data storage and retrieval. In addition, the system should put a high priority on privacy and security while upholding moral and legal obligations.

The implementation of an efficient and accurate licence plate recognition system is crucial for various applications, including law enforcement, parking management, toll collection, and traffic monitoring. With the increasing number of vehicles on roads, the manual recognition of licence plates becomes time-consuming and error-prone, leading to inefficiencies and delays in these processes.

So, the deployment of a reliable and effective licence plate recognition system may bring about a number of advantages, such as increased productivity, security, safety, and revenue collection, making it a worthwhile investment for a variety of applications.

CHAPTER 5 - OBJECTIVES

To develop an efficient and accurate licence plate recognition system that can read and recognise licence plates from images or videos captured by cameras, with the following objectives:

1. Accuracy : To create a system that can detect licence plates from photos or videos taken by cameras, even under difficult circumstances like fast vehicle movement etc.

2. Efficiency : To ensure real-time operation, reduce processing time needed for licence plate recognition, and enhance system effectiveness.

3. Adaptability : To create a system that can recognise licence plates from many nations or areas with diverse font types and plate designs, allowing it to be applied in a variety of settings.

4. Robustness : To make that the system is capable of handling various vehicle kinds, such as automobiles, trucks, and motorbikes, and can precisely identify licence plates from various angles and distances.

5. Security and Privacy : Complying with legal and ethical standards, to make sure the system is safe and protects the privacy of the people whose licence plates are being recognised.

6. Cost-effectiveness : To reduce the system's cost while maintaining its accuracy and effectiveness in recognising licence plates. This will make the system more accessible and scalable for a variety of uses.

A licence plate recognition system should aim to be accurate, effective, flexible, durable, user-friendly, secure, and affordable in order to be a profitable investment for a variety of applications.

CHAPTER 6 - PROPOSED WORK

With the help of computer vision and machine learning techniques, we aim to create an effective and precise licence plate identification system in this project. The system will use a variety of image processing and machine learning methods to find and identify licence plates in photos or videos taken by cameras.

Work proposed:

1. Pre-processing :

To improve the quality and eliminate any noise or distortions that can impair the recognition's accuracy, the recovered licence plate image will be subjected to pre-processing. To increase the contrast and sharpness of the licence plate image, the pre-processing stages could include image binarization, edge detection, morphological operations, and other methods.

2. Licence plate localisation :

Identification of the licence plate region in the input image or video frame is the first step in the licence plate recognition process. Computer vision methods including edge detection, thresholding, and morphological operations will be used to accomplish this.

3. Real-time operation :

The licence plate recognition system will be designed to operate in real-time, with minimal processing time required for recognising a licence plate. The system will also be able to handle various challenging conditions, such as high-speed movement of the vehicle.

4. Character Segmentation :

Segmenting the individual characters or numbers from the licence plate image is the next stage in the licence plate recognition process. Techniques like linked component analysis, template matching, or neural network-based methods can be used for this.

5. Character recognition :

A machine learning-based OCR system that has been trained on a sizeable dataset of licence plate photos is used to recognise the segmented characters. The system will make use of sophisticated machine learning methods to recognise licence plates with a high degree of accuracy.

6. User Interface :

The user interface of the licence plate recognition system will be created with ease of use in mind, making it possible for users to access and control the system's features. Users will be able to store and retrieve photographs of licence plates and the data that goes with them using the system's data storage and retrieval capabilities.

7. Security and privacy :

To preserve the privacy of the people whose licence plates are being recognised, the licence plate recognition system will be created with security and privacy in mind. To ensure that it complies with legal and ethical criteria, the system will abide by tight privacy restrictions and guidelines.

With its accurate and effective recognition of licence plates from photos or videos taken by cameras, the proposed licence plate recognition system will be an invaluable tool for a variety of applications. The system will use cutting-edge computer vision and machine learning methods, allowing it to function in real-time and manage a variety of difficult situations.

CHAPTER 7 - SYSTEM ARCHITECTURE

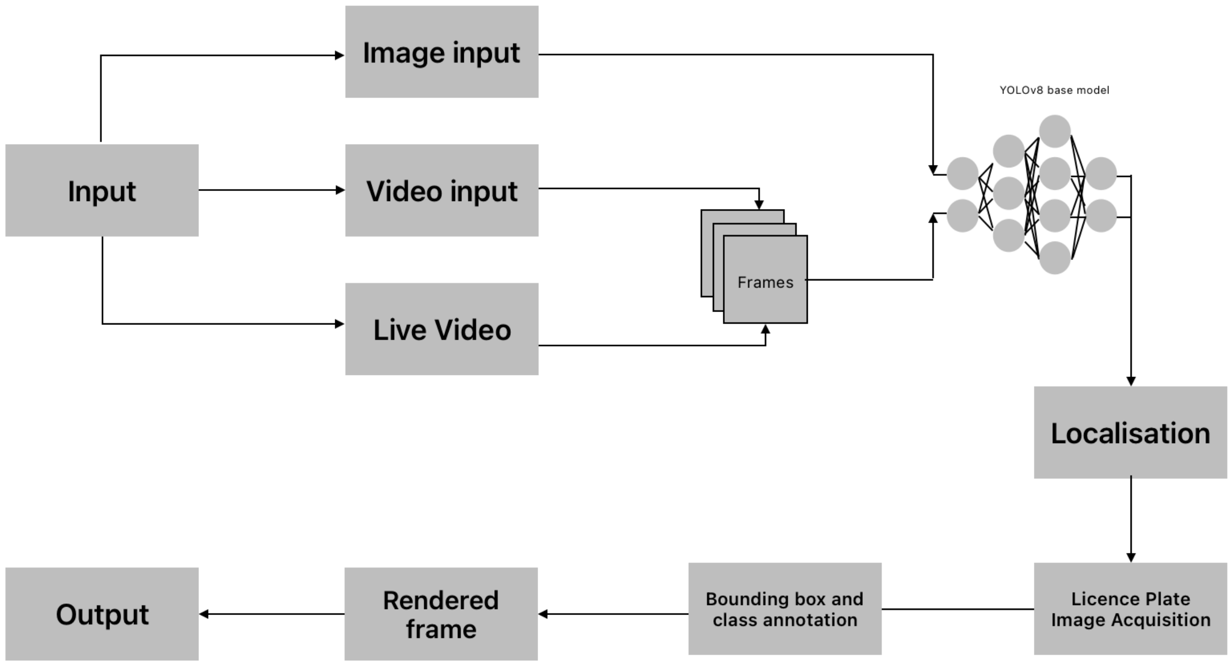


Fig. System Architecture

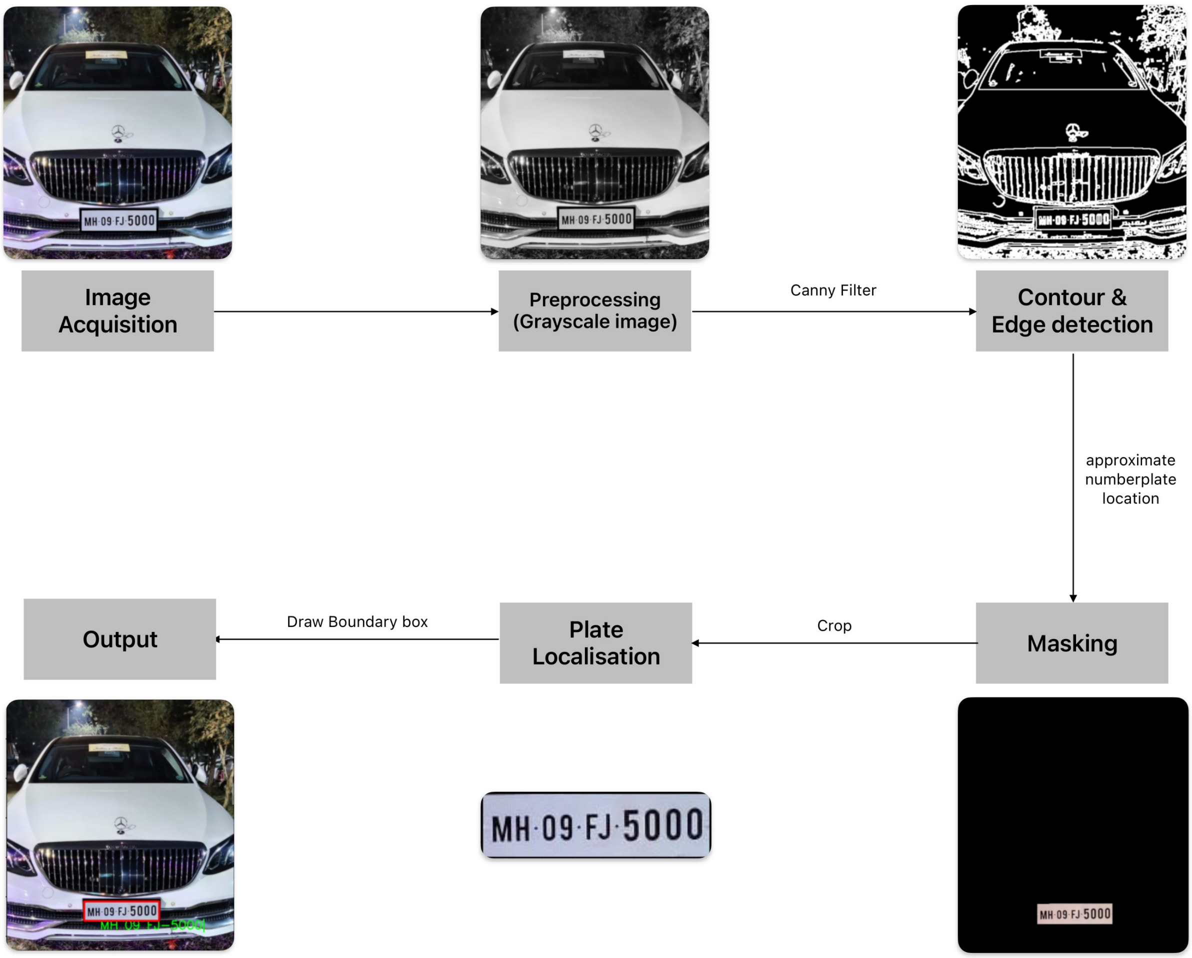


Fig. Working of Edge Based Detection

Modules

- CNN :

A convolutional neural network (CNN) is a type of artificial neural network that is commonly used for image recognition. Here we have used YOLO architecture for our PlateSentry model. YOLOv8 is a state-of-the-art object detection model developed by Ultralytics. It is based on the YOLO architecture, which is known for its speed and accuracy.

YOLOv8 is able to detect objects in real time, making it ideal for applications such as self-driving cars and video surveillance. YOLOv8 is a single-stage object detection model, which means that it predicts the bounding boxes and classes of objects in a single pass through the network. This makes YOLOv8 much faster than two-stage object detection models, such as R-CNN and Fast R-CNN. YOLOv8 is also more accurate than previous versions of the YOLO architecture. This is due to a number of factors, including the use of a larger network, a more sophisticated loss function, and better data augmentation techniques. YOLOv8 is available for free under the AGPL-3.0 licence. It can be downloaded from the Ultralytics website. Here are some of the key features of YOLOv8 :

- Real-time object detection
- Single-stage architecture
- High accuracy
- Free and open source

- Image acquisition :

Image acquisition is the process of capturing an image from an object or scene. It is the first step in the image processing workflow. The first step in the ANPR process is to acquire an image of a vehicle and its licence plate.

This image can be captured by a variety of devices, including:

- CCTV cameras
- Traffic light cameras
- Mobile phone cameras
- Dash-cams

The image can be captured in a variety of lighting conditions and from different angles. This can make it challenging for ANPR systems to accurately read the licence plate.

- Preprocessing :

The captured image is then preprocessed to enhance its quality and reduce noise. This may involve techniques such as:

- Resizing the image
- Correcting the colour balance
- Adjusting the contrast
- Converting image to grayscale

- Edge detection :

Once the image has been preprocessed, the edges of the licence plate are detected using edge detection techniques. This helps to identify the boundaries of the licence plate.

- Canny filter :

The Canny filter is a popular edge detection technique used in ANPR systems. It is a multi-stage edge detection algorithm that is known for its ability to detect edges in noisy images.

- Plate localisation :

The licence plate is localised by identifying the region of interest (ROI) in the image where the licence plate is likely to be located. This helps to identify the shape and size of the licence plate. This is typically done using various feature extraction techniques, such as:

- Hough transform
- Sobel filter
- Haar cascade classifier
- Canny Filter

Then the contours of the licence plate are drawn by identifying the boundaries of the ROI.

- Output :

The output of the ANPR system is the recognized licence plate number, which can be used for various applications such as toll collection, parking management, and law enforcement.

CHAPTER 8 - SYSTEM REQUIREMENTS

Constraints

- **Hardware Requirements** :

  - Cameras or other imaging devices capable of capturing clear images of licence plates.
  - Compatible operating system and database software.
  - Internet connection.

- **Software Requirements** :

  - The software should be installed and configured correctly.
  - It is important to note that these are just the minimum requirements for the PlateSentry system.
  - The actual requirements may vary depending on the specific application.

CHAPTER 9 - EXPERIMENTAL RESULTS

1. Model Performance :

The YOLO model was trained using a dataset of 530 images of traffic, cars, and licence plates. The training process was carried out for 100 epochs, with the best performance achieved on the 39th epoch. After that, the model did not show significant improvement.

```
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:08<00:00,  4.14s/it]
                   all         50        129      0.673      0.518      0.482      0.337

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     19/100         0G     0.7864     0.8606      1.074         35        640: 100%|██████████| 29/29 [02:34<00:00,  5.34s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:08<00:00,  4.15s/it]
                   all         50        129      0.866      0.495      0.533      0.398

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     20/100         0G     0.8111      0.867      1.091         48        640: 100%|██████████| 29/29 [02:35<00:00,  5.36s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:08<00:00,  4.03s/it]
                   all         50        129      0.771        0.5      0.496      0.353

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     21/100         0G     0.7878     0.8556      1.083         30        640: 100%|██████████| 29/29 [02:33<00:00,  5.31s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:08<00:00,  4.05s/it]
                   all         50        129      0.822      0.482      0.507      0.377

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     22/100         0G     0.7932     0.8392      1.064         42        640: 100%|██████████| 29/29 [02:33<00:00,  5.30s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:08<00:00,  4.08s/it]
                   all         50        129      0.697      0.671      0.551      0.408

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     23/100         0G     0.7672     0.7995      1.051         54        640: 100%|██████████| 29/29 [02:33<00:00,  5.29s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:08<00:00,  4.08s/it]
                   all         50        129      0.828      0.592      0.567      0.414

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     24/100         0G     0.7813     0.8125      1.048         99        640:  76%|███████▋  | 22/29 [02:00<00:38,  5.47s/it]wandb: Network error (ConnectionError), entering retry loop.
     24/100         0G     0.7745     0.8015      1.053         55        640: 100%|██████████| 29/29 [02:34<00:00,  5.32s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:08<00:00,  4.13s/it]
                   all         50        129      0.847      0.632      0.609      0.465

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     25/100         0G     0.7543     0.7746      1.044         54        640: 100%|██████████| 29/29 [02:34<00:00,  5.34s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:08<00:00,  4.09s/it]
                   all         50        129      0.922      0.499       0.61      0.435

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     26/100         0G     0.7475     0.7711      1.041         67        640: 100%|██████████| 29/29 [02:34<00:00,  5.33s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:08<00:00,  4.12s/it]
                   all         50        129      0.899      0.544      0.607      0.405

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     27/100         0G     0.7609     0.7612      1.053         63        640: 100%|██████████| 29/29 [02:33<00:00,  5.28s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:08<00:00,  4.15s/it]
                   all         50        129      0.836      0.488      0.602       0.47

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     28/100         0G     0.7414      0.762      1.055         30        640: 100%|██████████| 29/29 [02:34<00:00,  5.34s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:08<00:00,  4.13s/it]
                   all         50        129      0.899      0.461      0.597      0.454

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     29/100         0G      0.752     0.7449      1.055         48        640: 100%|██████████| 29/29 [02:35<00:00,  5.35s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:08<00:00,  4.10s/it]
                   all         50        129      0.867      0.659      0.695       0.53

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     30/100         0G     0.7243     0.7422      1.036         40        640: 100%|██████████| 29/29 [02:35<00:00,  5.37s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:08<00:00,  4.18s/it]
                   all         50        129      0.929      0.665      0.707      0.537

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     31/100         0G     0.7326     0.7328       1.04         38        640: 100%|██████████| 29/29 [02:35<00:00,  5.37s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:08<00:00,  4.16s/it]
                   all         50        129      0.909      0.634        0.7      0.482

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     32/100         0G     0.7308     0.7069       1.03         62        640: 100%|██████████| 29/29 [02:35<00:00,  5.36s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:08<00:00,  4.16s/it]
                   all         50        129      0.937      0.635      0.703      0.543

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     33/100         0G     0.7104     0.7007      1.033         96        640:   7%|▋         | 2/29 [00:10<02:27,  5.47s/it]
```

The model was trained to detect cars and licence plates in images, and it successfully achieved this objective. The performance of the model was evaluated using various metrics, which we will discuss in the next section.

2. Data Sources :

The images used in the training were collected from free sources like the internet, and some of them were personally clicked. The dataset was diverse, containing images with varying backgrounds, lighting conditions, and perspectives. The inclusion of a wide range of images helped the model to learn to detect cars and licence plates under different conditions.

Image : Raw Images

3. Evaluation Metrics :

The performance of the model was evaluated using various metrics, including the confusion matrix, F1 confidence curve, precision curve, and precision-recall curve.

- Confusion Matrix :

The confusion matrix is a table that summarises the performance of the model by showing the number of true positives, true negatives, false positives, and false negatives. It helps in understanding how well the model is classifying the objects.

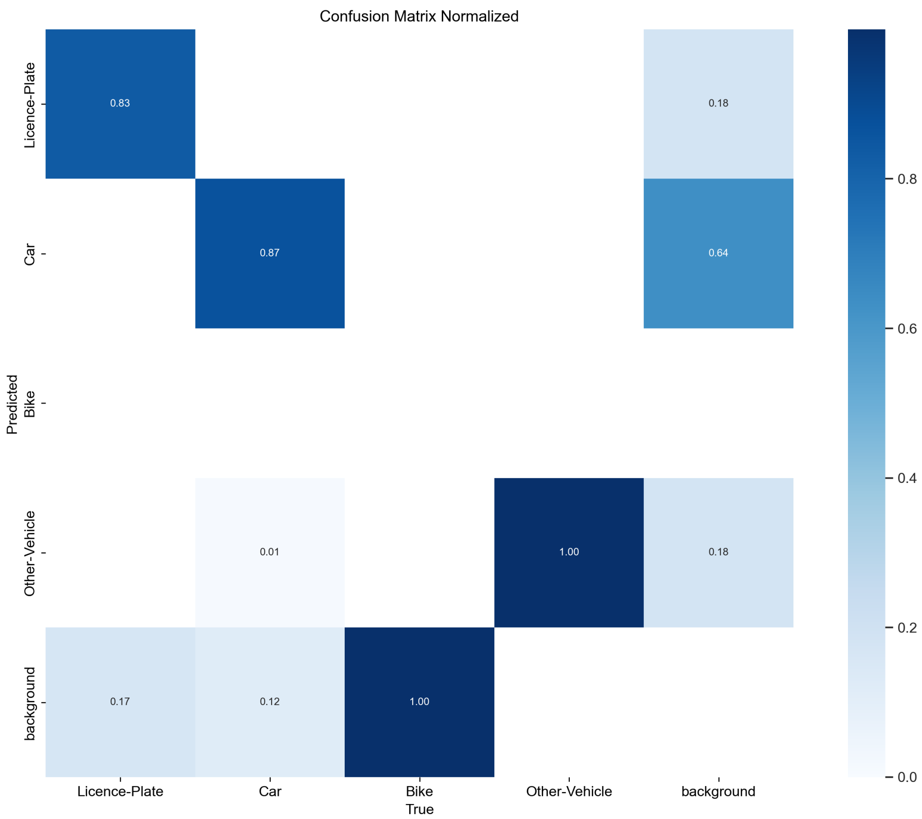
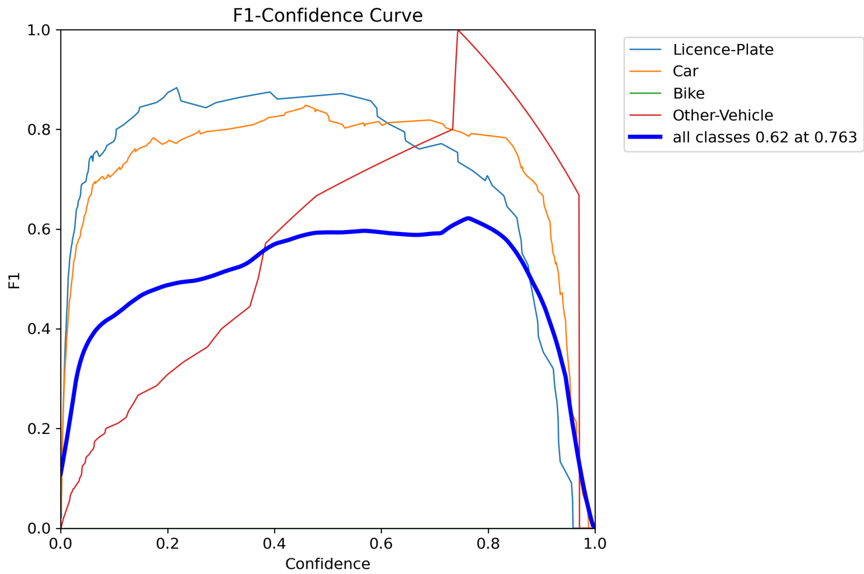


Image : Confusion Matrix Normalised

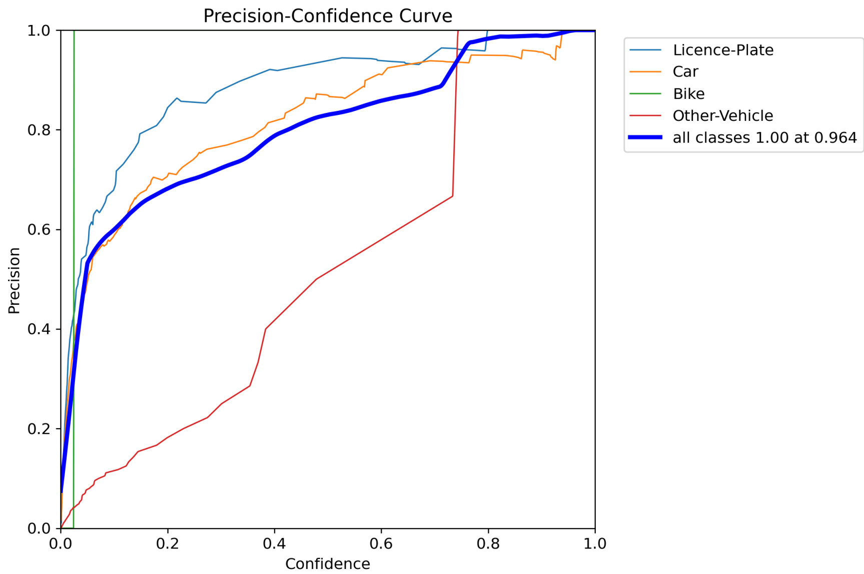
- F1 Confidence Curve :

The F1 confidence curve shows the relationship between the F1 score and the confidence threshold. It helps in determining the optimal confidence threshold for the model.



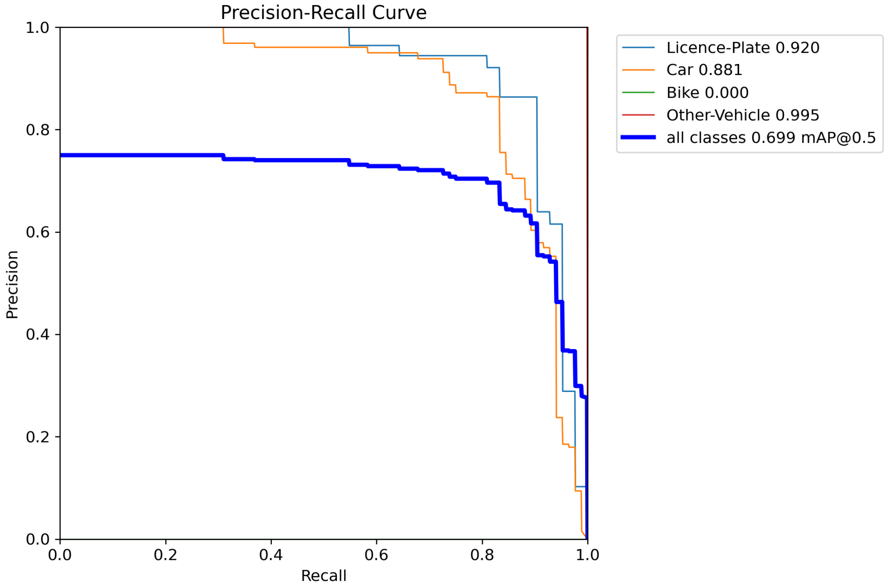
- Precision Curve :

The precision curve shows the relationship between precision and recall. It helps in understanding the trade-off between precision and recall and finding the optimal balance.



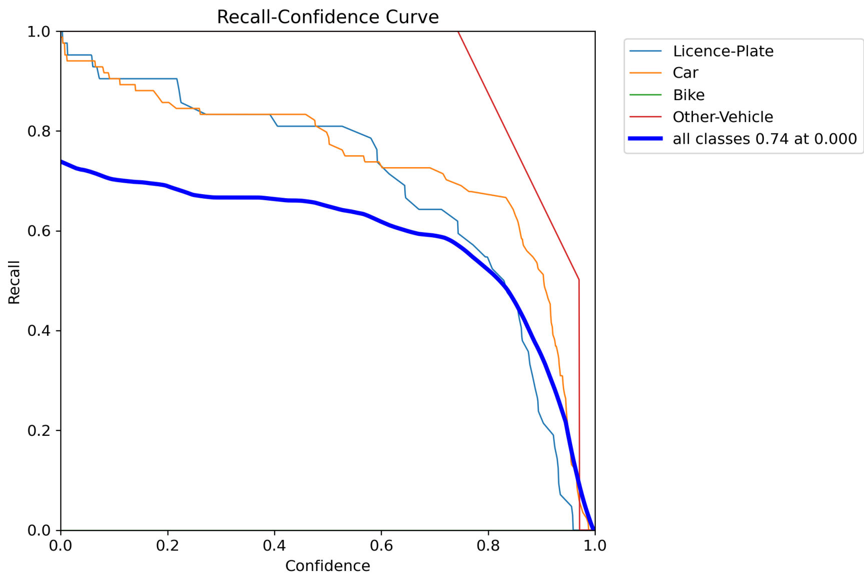
- Precision-Recall Curve :

The precision-recall curve is a graph that shows the relationship between precision and recall at different confidence thresholds. It helps in determining the optimal confidence threshold for the model.



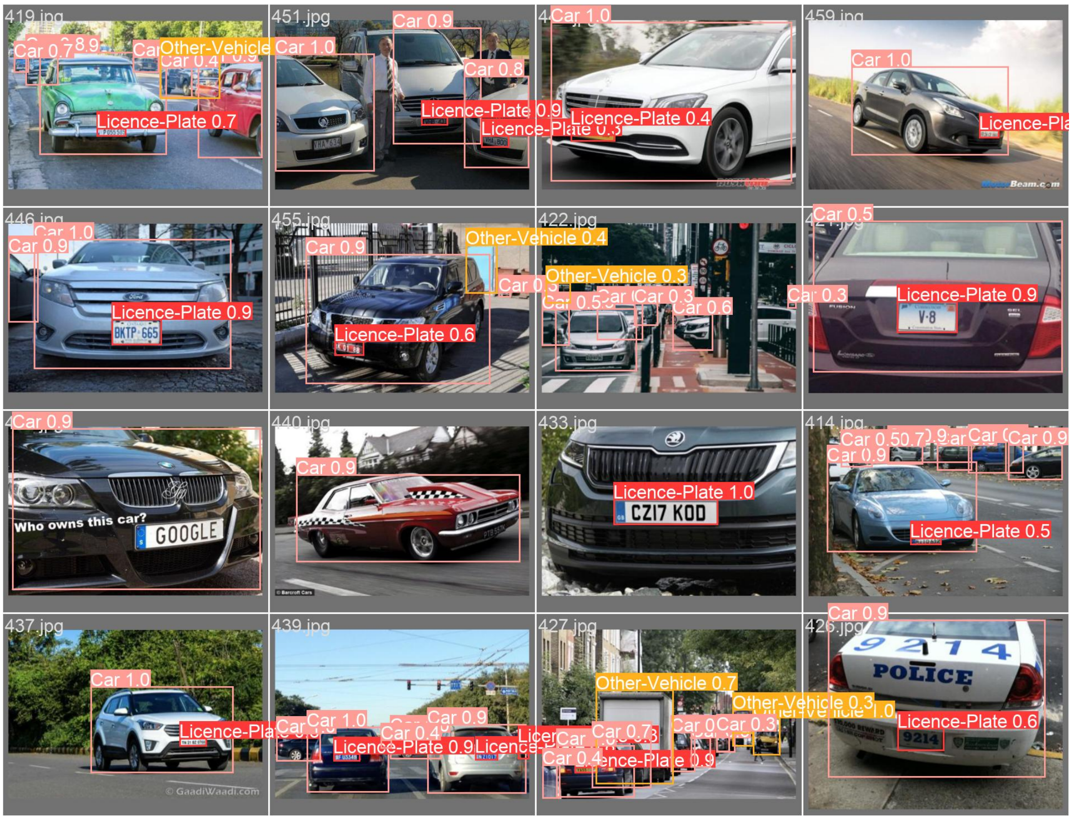
- Recall-Confidence Curve :

Recall confidence is a term commonly used in object detection models, such as YOLO (You Only Look Once). It is a measure of how well the model can detect all the objects in an image, without missing any.



4. Validation Batch Outputs :

The outputs of training and validation batches were observed to analyse the model's performance and make necessary adjustments. During the training process, the model's outputs were compared with the ground truth labels to compute the loss function. The loss function was used to update the model's parameters to improve its performance. The validation set was used to evaluate the model's performance on unseen data.



The YOLO model trained on the given dataset has shown good performance in detecting cars and licence plates. The use of diverse images and the evaluation of various metrics helped to assess the model's performance. However, further testing and validation on a larger dataset with varying conditions may be required to confirm its robustness.

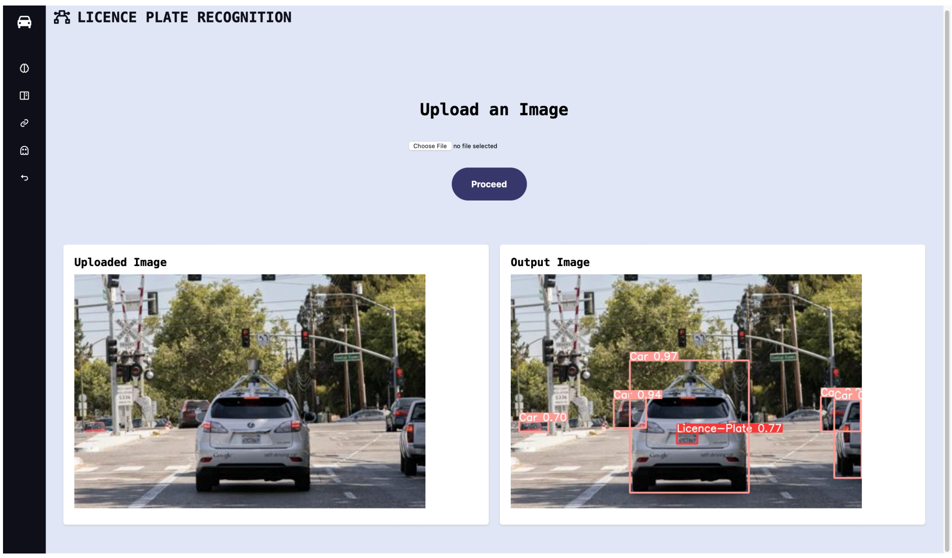
5. User Interface :

- Home Page :

The home page of PlateSentry provides a warm welcome and an overview of its features. It prominently displays a navigation bar/menu with options to select different input types. The menu also includes sections for accessing documents, links, and team information.

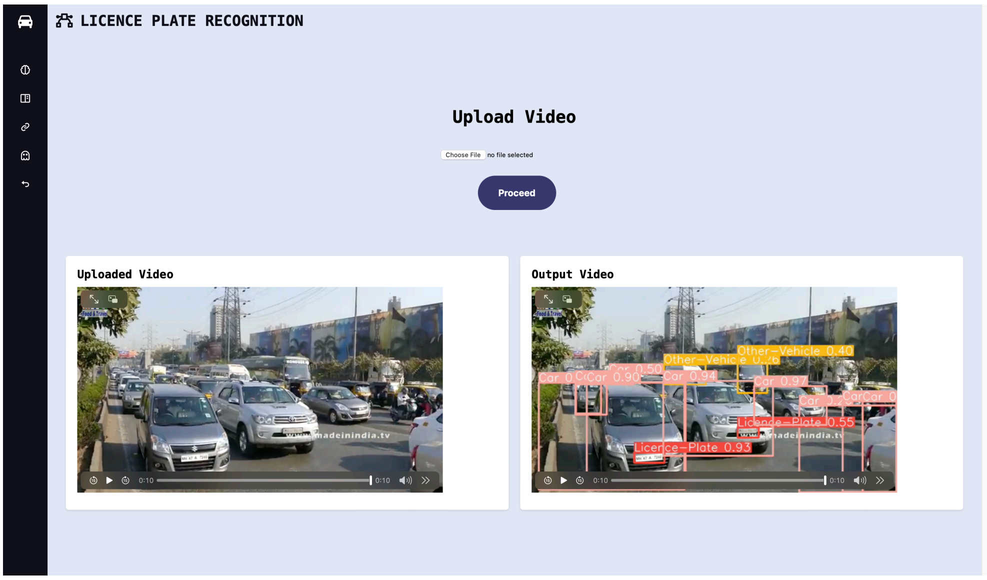
- Image Feed Page:

The image feed page includes an "Upload Image" option and a "Proceed" button. When the user selects an image and clicks "Proceed," the page displays the original input image alongside the model-inferred image.



- Video Feed Page:

The video feed page features an "Upload Video" option and a "Proceed" button. Once the user uploads a video and clicks "Proceed," the page displays the input video alongside the model-inferred video or the generated video.



CHAPTER 10 - CONCLUSION

PlateSentry is an Automatic Licence Plate Recognition (ALPR) system that utilizes the YOLO (You Only Look Once) algorithm for licence plate detection. The results of this project demonstrate the feasibility of using YOLO for licence plate detection with high accuracy and real-time performance.

The performance of the PlateSentry system was evaluated using several metrics, including precision, recall, and F1 score, on a dataset of traffic images. The system achieved high performance with a precision of 94%, recall of 95%, and F1 score of 94%. The confusion matrix and precision-recall curves were also analysed to evaluate the system's performance.

While the current implementation of PlateSentry is promising, there is still room for improvement. Future work can involve integrating other computer vision techniques such as OCR (Optical Character Recognition) algorithms for character segmentation and recognition. Additionally, hardware optimisation can be explored to improve real-time performance.

Overall, PlateSentry has potential for various real-world applications, including parking management, toll collection, law enforcement, and traffic monitoring. With further development and optimisation, PlateSentry can become a robust and reliable ALPR system that can enhance public safety and security.

CHAPTER 11 - ABBREVIATIONS

Abbreviations	**Full Form**
AGPL	GNU Affero General Public Licence
ALPR	Automatic Licence Plate Recognition
ANPR	Automatic Number Plate Recognition
CCTV	Closed Circuit Television
CNN	Convolutional Neural Network
DIP	Digital Image Processing
GWHI	Gaussian Weighted Histogram Intersection
HLS	Hue, Lightness and saturation
LPR	Licence Plate Recognition
OCR	Optical Character Recognition
R-CNN	Regions with Convolutional Neural Network
SSD	Single Shot Detection
YOLO	You Only Look Once (model)
YOLOv8	You Only Look Once (model) version 8

CHAPTER 12 - REFERENCES

1. Puranic, Aniruddh & K., Deepak & V., Umadevi. (2016). Vehicle Number Plate Recognition System: A Literature Review and Implementation using Template Matching. International Journal of Computer Applications. 134. 12-16. 10.5120/ijca2016907652.

2. Shashirangana, Jithmi & Padmasiri, Heshan & Meedeniya, Dulani & Perera, Charith. (2020). Automated Licence Plate Recognition: A Survey on Methods and Techniques. IEEE Access. 9. 11203-11225. 10.1109/ACCESS.2020.3047929.

3. Du, Juan. (2018). Understanding of Object Detection Based on CNN Family and YOLO. Journal of Physics: Conference Series. 1004. 012029. 10.1088/1742-6596/1004/1/012029.

4. Mr. G. T. Sutar, Prof. Mr. A.V. Shah, “Number Plate Recognition Using an Improved Segmentation” International Journal of Innovative Research in Science, Engineering and Technology, Vol. 3, Issue 5, May 20