A MAJOR PROJECT MID TERM REPORT ON

Parking Management System with Automatic Number Plate Recognition

Submitted by:

Aayushma Paudel [20070546]

Anisha Silwal [20070547]

Garima Paudel [20070551]

Nisha Pokharel [20070557]

Submitted in Partial Fulfillment of the Requirements for

Bachelor of Engineering in Computer Engineering

under **United Technical College**

Submitted to:

Department of Computer Engineering

Supervisor: Er. Sahit Baral

UNITED TECHNICAL COLLEGE

(Affiliated to Pokhara University)

Bharatpur-11, Bhojad, Chitwan

[Subject Code: CMP-490]

June, 2024

… # Abstract

This report focuses on developing a Parking Management System using the YOLOv8 Deep Learning Model for Automatic Number Plate Recognition (ANPR). The project aims to efficiently manage parking spaces by automating the identification and keeping track of vehicles through their number plates. The methodology involves dataset collection, annotation, preprocessing, and augmentation for training the YOLO algorithm. The system aims to provide real-time information to drivers about parking space availability and establish a robust record-keeping mechanism for monitoring vehicles. Additionally, the project includes the development of parking management software with database integration, testing, and deployment. The expected outcome of the project is an automated system utilizing YOLOv8 for real-time license plate detection and extraction. This system will recognize vehicle number plates, extract the numbers, and store them in a database. The end goal is to facilitate efficient management of parking by maintaining accurate records of entry and exit times along with corresponding license plate information and help drivers find parking spaces in the parking lot.

Keywords: *Parking Management System, ANPR, Deep Learning, Data Collection, Preprocessing, Augmentation, YOLOv8 algorithm*

Table of Contents

List of Figures

List of Tables

List of Equations

Acronyms and Abbreviation

AI	:	Artificial Intelligence
AHE	:	Adaptive Histogram Equalization
ALPR	:	Automatic License Recognition
ANPR	:	Automatic Number Plate Recognition
BBA	:	Boundary Box Analysis
CCA	:	Connected Component Analysis
CSS	:	Cascading Style Sheets
DWT	:	Discrete Wavelet Transform
JS	:	JavaScript
MSE	:	Mean Square Error
NMS	:	Non-Max Suppression
PSNR	:	Peak Signal to Noise Ratio
SVM	:	Support Vector Machine
YOLO	:	You Look Only Once

Chapter 1: Introduction

1.1. Background

As cities get crowded and space becomes limited, managing parking efficiently becomes important. One smart solution is using Automatic Number Plate Recognition (ANPR) in parking systems. This technology automates the process of entering and leaving parking areas, making it easier for everyone. It's like a high-tech way to ensure that parking spaces are used wisely in busy city areas where space is tight. This not only helps with traffic flow but also makes sure parking is secure, well recorded and well organized.

Automatic Number Plate Recognition (ANPR) system is a crucial component of smart cities as it includes number plate detection and Optical Character Recognition (OCR) technology to read detected vehicle number plates. The detection task can be done using several alternatives like Faster R-CNN, SSD (Single Shot Multibox Detector), RetinaNet etc. YOLO model has been selected for the project because unlike some algorithms requiring multiple passes, YOLO conducts detection in a single forward pass, making it computationally efficient and suitable for real-time applications like video analysis. ANPR enables traffic control and law enforcement through an automated, fast, reliable, and robust vehicle plate recognition system [1]. License plate readers employ Optical Character Recognition (OCR) technology to quickly and accurately interpret vehicle information. To automatically identify and monitor cars as they enter and depart, ANPR systems are often employed in parking garages, toll booths, and traffic control systems. An automated vehicle entry system using ANPR typically consists of a camera, a processor, and a database. The camera is mounted at the facility's entrance and captures images of the license plates of incoming vehicles. The license plate’s characters are read using OCR [2].

Identifying vehicles within a country relies on the recognition of a distinctive alphanumeric number displayed on their license plates, and localizing these plate regions in images presents a challenging task due to variations in color, texture, size, shape, and position of these regions [3].

Monitoring of vehicle traffic and management of parking lots in crowded and congested locations like shopping malls, business parks, residential complexes etc has emerged as a major area of research in Intelligent Transportation System (ITS). Often parking lot operators do not enter complete details or sometimes enter incorrect details into the system, especially during peak hours, which may later cause problems for vehicle owners while exiting the lot and is also a major security issue [4].

1.1.1. Nepalese License Plate

The Nepalese government has categorized license plates (LPs) into seven classes based on vehicle ownership, denoted by color combinations in the LP's background and foreground. LPs come in three structures: 1-row, 2-row, and 3-row, each with specific character arrangements. These plates include fixed elements like Province, province identifier (PN), plate status (PS), lot number (L), load type (LD), and vehicle identity (X). Various LP structures exist, such as 3-row Red (private) LP and zonal format LPs. Inconsistencies arise in non-standardized LPs, with variations in fonts, character sizes, and spacing [5].

Background (Plate)	Foreground (Characters)	ownership	Load			letters and Digits			
			Type	letters		Zonal Code		Digits	
				English	Devanagari	English	Devanagari	Arabic	Devanagari
			Heavy	KA	क	ME	मे	0	०
Red	White	Private	Middle	CHA TA	च त	KO	को	1	१
			Light	PA	प	S	स	2	२
			Heavy	PA	प	J	ज	3	३
Green	White	Tourist	Middle	YA	य	NA	ना	4	४
			Light	PA	प	BA	बा	5	५
			Heavy	GHA	घ	GA	ग	6	६
Yellow	Blue	Public/ National Institution	Middle	YNA	ञ	LU	लु	7	७
			Light	MA	म	DHA	ध	8	८
			Heavy	KHA	ख	BHE	भे	9	९
Black	White	Public	Middle	JA	ज	RA	रा	**DLP characters used in proposed system	
			Light	THA	थ	KA	क		
			Heavy	GA	ग	SE	से	**Characteristics of load type and zonal letters are identical within and between classes, 23 letters used	
White	Red	Government	Middle	JHA	झ	MA	म		
			Light	BA	ब	**New Style (3-row LP)**			
			Heavy	**		**English**	**Devanagari**		
Red Blue	White	Minister	Middle	JHA	झ				
			Light	**					
			Heavy	**		**Province**	प्रदेश		
Blue	White	Diplomat	Middle	C D	सी डी				
			Light	**					

Figure 1: License Plate System in Nepal [5]

1.2. Problem statement

The current state of parking management in Nepal faces several challenges that hinder efficient and organized operations. One of the primary issues is the reliance on manual record-keeping, which not only consumes valuable human resources but is also prone to errors and inefficiencies. The absence of a robust automated parking system utilizing Automatic Number Plate Recognition (ANPR) technology worsens the problem. The lack of automation not only results in a difficult process for both parking lot management and drivers but also contributes to the absence of a reliable record-keeping system for vehicles entering and leaving the premises.

Furthermore, there is a notable absence of a centralized platform for parking lot managements to access real-time information about vehicles within their facilities. Additionally, drivers lack a user-friendly platform to ascertain the availability of parking spaces in real-time, leading to frustration and inefficiencies in their search for suitable parking spots.

In the context of Nepal, where automation systems for parking management are yet to be implemented, the need for a comprehensive solution is urgent. Moreover, during our research, we identified a unique challenge specific to Nepal – the difficulty in accurately identifying and processing Nepali letters using existing ANPR systems. This adds an additional layer of complexity to the adoption of automated parking systems in the country. In light of these challenges, our proposed project aims to develop an advanced Parking Management System using ANPR that not only addresses the current limitations in record-keeping and automation but also incorporates solutions for the unique linguistic challenges posed by Nepali letters.

1.3. Research Questions

1. How can YOLOv8 be used to achieve real-time license plate detection for automatic vehicle recognition in a parking system?
2. How can the implementation of this system facilitate drivers in finding parking spaces?
3. How can we develop a system for proper recording of vehicles entering and exiting a parking lot?

1.4. Objectives

The project aims to develop an affordable, scalable, and accurate ANPR-based parking management system, contributing to improved security and streamlined operations. The objectives of the project are listed below:

1. To Implement a system (YOLOv8 Deep Learning Model) for real-time license plate detection that automatically recognizes vehicles entering and exiting the parking facility through ANPR.
2. To provide real-time information to drivers about the availability of parking spaces.
3. To establish a robust record-keeping mechanism that captures and maintains vehicle license plate information along with entry and exit time records for effective monitoring.

1.5. Application

The "Parking Management System using ANPR" project has various applications across different sectors. Some potential applications include:

1. Urban Parking Management: The system can be used for streamlining parking in busy urban areas to reduce congestion and optimize space utilization and for enhancing the efficiency of municipal parking systems for better city planning.
2. Business Parks and Commercial Areas: The proposed system can be used for managing employee and visitor parking in business parks and commercial

complexes and for improving the overall traffic flow and parking experience for customers and clients.

3. Event and Venue Management: The system can also be used for facilitating efficient parking solutions for events, concerts, and sports venues and ensuring a smooth flow of traffic and for minimizing parking-related challenges during large gatherings.
4. Commercial Parking Lots: The system can be used for optimizing parking operations in commercial parking lots, including pay-and-park facilities and for enhancing customer experience through quick and convenient parking solutions.
5. Security and Surveillance Integration: The system can also be used by integrating ANPR technology with security systems for enhanced surveillance and access control and improving overall security measures by identifying and monitoring vehicles entering specific areas.

These applications demonstrate the versatility and wide-reaching impact of the project, offering solutions to various parking challenges across different sectors and environments.

1.6. Scope and Limitations

The Parking Management System has a wide scope and presents a holistic approach to streamline vehicle monitoring in parking centers by leveraging the YOLOv8 Deep Learning Model for real-time license plate detection. The system's scope encompasses the development of an adaptable and user-friendly interface that facilitates effective navigation and provides real-time information on parking availability and vehicle movements. Key features include scalability to accommodate diverse parking spaces and traffic volumes, alongside the implementation of YOLOv8 to ensure accurate and prompt license plate detection, contributing to the creation of a comprehensive and efficient parking management solution. Furthermore, the system aims to establish a robust record-keeping mechanism that captures the license plate information, complemented by entry and exit time records for enhanced monitoring capabilities.

Despite its promising scope, the Parking Management System does have certain limitations. The accuracy of license plate detection may be compromised under extreme environmental conditions, such as severe weather or low lighting, which could impact the system's overall reliability. Additionally, there may be initial implementation costs associated with hardware, software, and training. Dependency on hardware reliability poses another limitation, as the system's performance depends on the proper functioning and maintenance of cameras and servers. Addressing data privacy concerns remains crucial, and despite advanced security measures, continuous adaptation to evolving regulations and user expectations is essential. Furthermore, challenges may arise in user training and adoption, requiring careful consideration and proactive measures to ensure a smooth transition for parking administrators and users. Integrating the system with existing infrastructure or legacy systems may also present obstacles, necessitating strategic planning and potential modifications for seamless integration. Understanding these limitations is integral to managing expectations and addressing challenges during the deployment and operation of the Parking Management System.

Chapter 2: Literature Review

The Literature Review portion of the proposal is based on the observations of many research and journals on similar titles.

According to the paper by Budi Setiyono et.al [6], different processes were performed to detect the number plate correctly such as character recognition was done to get text character data from license plate. YOLOv3 (You Look Only Once), and Darknet-53 were used as feature extractors. The data used were number plate images derived from the extraction and cropping of motorized vehicle videos taken using cellphones and cameras. Two models were tested: one without preprocessing and one with additional data preprocessing to enhance image quality. The results indicate that the model with preprocessing achieves higher accuracy, with 88% in number plate recognition and 98.2% in character recognition. In contrast, the model without preprocessing achieves a lower accuracy, with 80% in number plate recognition and 97.1% in character recognition.

In their work described in reference [7], M. M. Abdellatif et.al used 200 images to identify Egyptian car plates. The model successfully identified Arabic license plates with 93% accuracy. A prototype was implemented using ESP32 Cameras and Raspberry-Pi to test the system's performance.

According to a paper by Mohamed S. Farag et.al [8], image processing was used for smart parking entrance control. Car plate recognition involved preprocessing, license plate detection, character extraction, and recognition. Techniques such as image enhancement, noise reduction, color filtering, DWT (Discrete Wavelet Transform) for feature extraction, and SVM (Support Vector Machine) as a classifier contribute to achieving high detection (97.8%), segmentation (98%), and recognition (97%) rates, making it an effective method for smart parking.

A paper by Hanae Moussaoui et.al [9], introduces a novel Arabic and Latin license plate detection and recognition method. In the first stage of the proposed method and after gathering images to build a new dataset, YOLOv7 was used to detect and localize the license plate in the image. The dataset was labeled manually before feeding it to the detection system. Secondly, some of the machine learning algorithms were used to enhance the detected license plate. Thresholding and the kernel algorithms were used to remove the additional vertical lines in the plate. Afterward, Arabic OCR (Optical Character Recognition) and Easy OCR techniques were used to recognize Arabic and Latin letters on the license plate. The proposed detection algorithm achieved a precision of 97%, a recall of 98% and an F1 score of 98% in license plate detection. For image segmentation, while using Arabic OCR and Easy OCR to segment and extract characters from the detected license plate, an accuracy of 99 % was achieved.

The article in reference [10], focuses on object detection techniques, the YOLO v8 architecture, and prior studies on license plate detection. The authors highlight the unique challenges of recognizing Bangla license plates and discuss relevant literature in the field. They emphasize the significance of YOLO v8 as a state-of-the-art object detection framework known for real-time capabilities. The article addresses the methodology, including data collection, preprocessing, and model training, while discussing the challenges faced, such as data labeling and handling varying plate formats. The YOLO v8 model demonstrated outstanding performance in recognizing Bangla license plates, achieving a mean Average Precision of 98.4% on the test set and a precision of 98.1%. The analysis further revealed the model's robustness in handling diverse license plate variations, emphasizing its potential for real-world applications and ability to generalize well across challenging scenarios. Overall, the YOLO v8 architecture proves highly effective for Bangla license plate recognition.

In their paper presented in reference [11], V. Gnanaprakash, N. Kanthimathi, and N. Saranya propose an Automatic Number Plate Recognition (ANPR) system using deep learning, specifically utilizing the YOLO algorithm for vehicle and license plate detection. The research addresses the challenges of real-time processing of surveillance camera footage for efficient vehicle tracking. A four-step approach, involving video-to-image conversion, car detection, license plate localization, and character recognition is introduced. The deep learning model employs the Image AI library and is trained on a dataset of Tamil Nadu license plate images. The system achieves high accuracy, with 97% for car detection, 98% for number plate localization, and 90% for character recognition. The authors highlight the significance of ANPR in traffic management, parking systems, and security applications, emphasizing the need for real-time, accurate, and automated tracking of vehicles. The proposed system demonstrates advancements over existing methods, leveraging YOLO and Image AI for enhanced performance in dynamic environments. The literature review discusses prior studies on ANPR, including approaches using machine learning, deep learning, and OCR techniques, providing a comprehensive overview of the current state of the art.

In the paper by R. Laroca et *al.* [12], the proposed ALPR system tackles the complexities of detecting LPs in traffic images. Instead of directly working on image frames, it first locates vehicles and then identifies their License Plate (LPs) within the vehicle patches. This two-stage process helps overcome challenges such as confusion with non-LP textual elements. A novel aspect is the introduction of a layout classification stage after LP detection, using a unified network for robust recognition across different LP layouts. The system leverages YOLO-inspired models for vehicles and LP detection, while LP recognition employs CR-NET. The approach proves adaptable to new LP layouts through retraining, offering a promising solution for real-world applications with diverse LP configurations.

The literature by S.Kaur [13] on ANPR systems underscores the advancements in image acquisition and processing. Researchers tackle challenges presented by diverse image categories, including light, dark, low contrast, blurred, and noisy images, with a focus on weather-induced variations. Pre-processing techniques, such as iterative bilateral filters and Adaptive Histogram Equalization (AHE), enhance image quality by improving contrast and reducing noise. Morphological operations, image binarization, and Sobel edge detection contribute to refining candidate plate areas. The literature highlights challenges in character segmentation and recognition, addressed through methods like Connected Component Analysis (CCA) and Boundary Box Analysis (BBA). The proposed ANPR approach proves effective in handling diverse image conditions, exhibiting improvements in metrics like Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), and overall success rate when compared to existing methods.

In conclusion, several articles highlight the diverse applications and methodologies employed in automatic license plate recognition (ALPR) using various technologies. YOLO-based models, such as YOLOv3, YOLOv7, YOLO v8, and YOLO-inspired architectures, consistently emerge as powerful tools for efficient vehicle and license plate detection. These models exhibit real-time capabilities and robust performance across different scenarios, contributing to high accuracy rates in number plate recognition and character extraction. The importance of preprocessing techniques, machine learning algorithms, and deep learning approaches is huge in enhancing ALPR systems' overall effectiveness. The results underscore YOLO's versatility and effectiveness in handling challenges related to license plate detection, localization, and character recognition, making it a preferred choice for researchers and developers working on ALPR applications.

Chapter 3: Methodology

3.1. Automatic Number Plate Recognition (ANPR) System

Automatic Number Plate Recognition (ANPR) system is a system that reads and process images that consist of vehicle number plate as input and recognizes the number plate as output automatically [14]. Figure below shows the overview of the ANPR system.

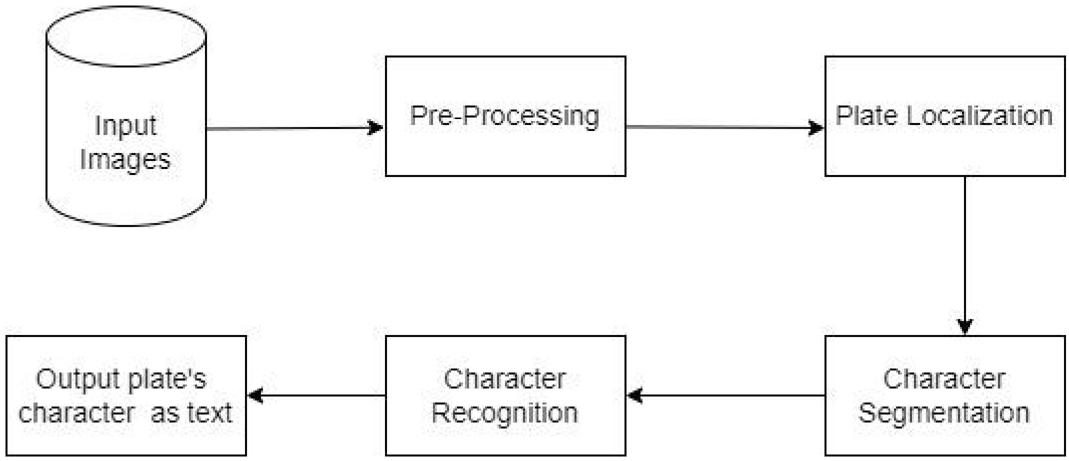

Figure 2: Automatic Number Plate Recognition System [15]

The license plate recognition process begins with the input images as the initial step. These images may come from various sources such as cameras or snapshots. The second step involves preparing the image for analysis or human perception through a pre-processing stage. This step ensures that the image is of good quality and suitable for computational processing or visual understanding. Following this, the plate localization stage comes into play, which focuses on identifying and pinpointing the location of the license plate within the image. Once the plate is located, the character segmentation stage takes over, separating each individual letter or number on the plate. This segmentation is crucial for making it easier to identify and analyze the characters. The final phase involves character recognition, where the system identifies and interprets the segmented characters, providing a comprehensive understanding of the license plate information. In essence, the entire process involves preparing the image, finding the license plate, breaking down its characters, and ultimately recognizing and understanding each character for various applications such as automated vehicle identification or security systems [15].

The overall steps involved in Automatic Number Plate Recognition (ANPR) are explained below:

3.1.1. Dataset Collection

In the first step i.e. Data Collection, large amounts of data was gathered by capturing images of vehicles, ensuring clear visibility of number plates. Total of 942 images were collected to form a dataset covering vehicles in different scenarios ensuring the dataset encompasses various lighting conditions, vehicle types, and plate variations to ensure the robustness of the model. The type of data that has been used is a primary source of data or custom dataset [10].

3.1.2. Dataset Preparation and Annotation

In this phase, all the collected 942 images were first cropped to ensure that the license plate is clearly visible. This was followed by the dataset annotation step where the images were annotated by marking bounding boxes around the license plates using Roboflow. Roboflow, as a platform providing tools and services for computer vision and image processing tasks, has been utilized to efficiently manage and preprocess the image datasets needed for the project. This annotation step formed the foundation for training the YOLOv8 model to accurately recognize and localize license plates [10].

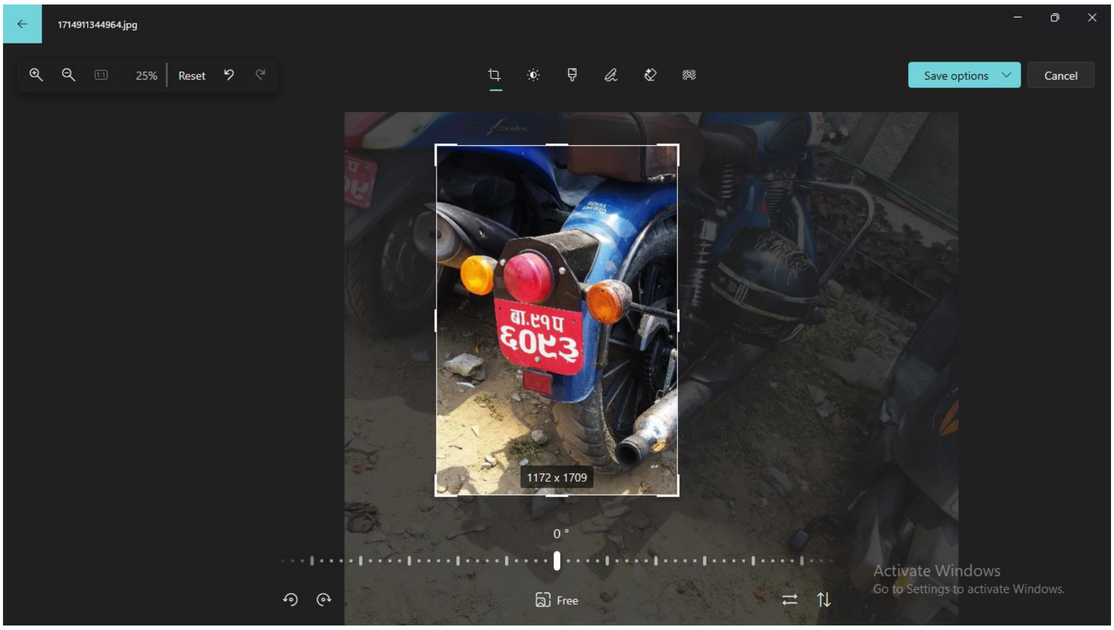

Figure 3: Image Cropping

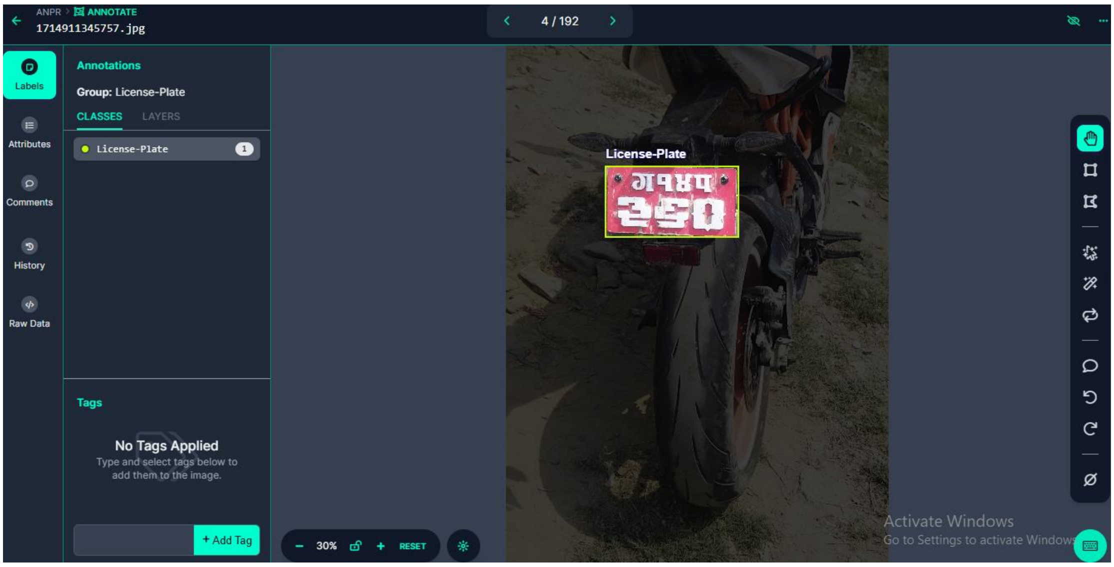

Figure 4: Image Annotating

3.1.3. Dataset Splitting

In the Dataset Splitting stage, the dataset was divided into training, validation and testing sets, comprising 70%, 20%, and 10% of the dataset, respectively. This step divided datasets as: 661 images in training set, 188 images in validation set, 94 images in test set.

3.1.4. Preprocessing and Augmentation

In this Preprocessing phase, the Auto-Orient option was enabled to ensure that all images have a consistent orientation. Also the images were resized i.e stretched to 640*640. Similarly, several augmentation tasks were also done. Each training example was augmented to produce three variations. The augmentations included random rotations between -15° and +15°, applying grayscale to 15% of the images, adjusting brightness between -20% and +20%, adding blur up to 0.7px, and introducing noise to up to 1.49% of the pixels. This step multiplied the number of images in the training dataset by 3 and the training set reached a total of 1983 images. This way, the dataset of total 2265 images was prepared after this step. This diverse augmentation strategy helped in creating a robust model by exposing it to various transformations of the training data [16].

3.1.5. Training Setup and Configuration

For training purposes, the need for high computation was noticed. So, NVIDIA drivers, CUDA toolkit, and cuDNN were installed. The environment variables were set. This was done with the help of support matrix for CUDA and cuDNN.

```
import torch

print("Is CUDA available:", torch.cuda.is_available())
print("CUDA device count:", torch.cuda.device_count())
print("Current CUDA device:", torch.cuda.current_device())
print("CUDA device name:", torch.cuda.get_device_name(torch.cuda.current_device()))
```

```
Is CUDA available: True
CUDA device count: 1
Current CUDA device: 0
CUDA device name: NVIDIA GeForce MX330
```

```
print(torch.__version__)
```

```
2.3.1+cu118
```

Figure 5: Training Setup and Configuration

3.1.6. Model Training

For model training, we used the YOLOv8 model, which is a pre-trained model on the COCO dataset. Different parameters were configured as per the requirements, such as batch size 4, workers 4, save period 5, image size 512, and number of epochs 25. The YOLOv8 model was trained using the annotated and preprocessed dataset obtained from the previous steps.

About YOLO Algorithm

YOLO algorithm aims to predict a class of an object and the bounding box that defines the object location on the input image. The main idea behind YOLO is to divide the input image into a grid and make predictions for each grid cell. Unlike traditional object detection methods, YOLO performs object detection in a single forward pass through the neural network.

It recognizes each bounding box using four numbers:

i. Center of the bounding box (b_x, b_y)

ii. Width of the box (b_w)

iii. Height of the box (b_h)

YOLO predicts the corresponding number for the predicted class '*c*' and the probability of the prediction (P_c). YOLO is designed to be fast and efficient, allowing for real-time object detection.

Several new versions of the same model have been proposed since the initial release of YOLO in 2015, each building on and improving its predecessor. Here's a timeline showcasing YOLO's development in recent years [17].

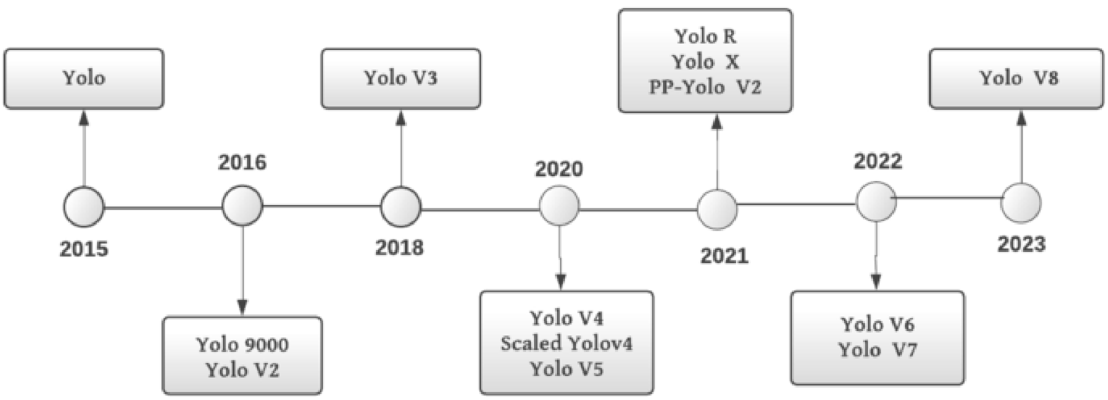

Figure 6: Evolution of YOLO Algorithm [17]

Yolo Architecture

The YOLO algorithm takes an image as input and then uses a simple deep convolutional neural network to detect objects in the image. The architecture of the CNN model that forms the backbone of YOLO is shown below.

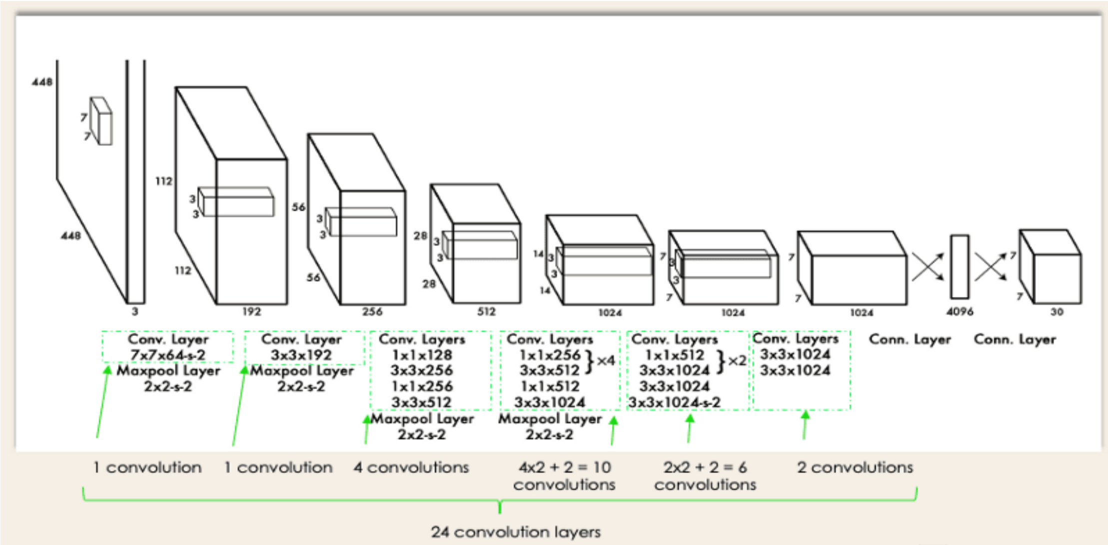

Figure 7: Architecture of YOLO Algorithm [18]

The first 20 convolution layers of the model are pre-trained using ImageNet by plugging in a temporary average pooling and fully connected layer. Then, this pre-trained model is converted to perform detection since previous research showcased that adding convolution and connected layers to a pre-trained network improves performance. YOLO's final fully connected layer predicts both class probabilities and bounding box coordinates [18].

YOLO divides an input image into an S × S grid. If an object's center falls within a grid cell, that cell is responsible for detecting the object. Each grid cell predicts 'B' bounding boxes and confidence scores, reflecting the model's confidence in the box containing an object and its accuracy. The bounding box attributes are predicted using a single regression module, forming a vector Y = [pc, bx, by, bh, bw, c1, c2], where pc is the probability of an object being in the grid, bx and by are the center coordinates, bh and bw are the height and width, and c1, c2 are class probabilities.

To address multiple bounding boxes for a single object, YOLO uses the Intersection over Union (IoU) metric and Non-Max Suppression (NMS) to retain only the most relevant boxes, reducing noise [19].

The formula of IoU is:

$$IoU = \frac{Area\ of\ the\ intersection\ between\ B1\ and\ B2}{Area\ of\ the\ union\ between\ B1\ and\ B2} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{Equation(1)}$$

where B_1 and B_2 are two bounding boxes.

About Yolov8

Released in May 2023, YOLOv8 is the latest iteration, offering significant improvements in accuracy and performance. It supports various vision tasks, such as object detection, segmentation, pose estimation, tracking, and classification. YOLOv8 features a modified CSPDarknet53 backbone with a C2f module, an anchor-free model with a decoupled head, and advanced loss functions (CIoU and DFL for bounding-box loss and binary cross-entropy for classification loss). Evaluated on the MS COCO dataset, YOLOv8x achieved an AP of 53.9% at 640 pixels and 280 FPS on an NVIDIA A100 and TensorRT. Figure shows the detailed architecture of YOLOv8. YOLOv8 uses a similar backbone [20].

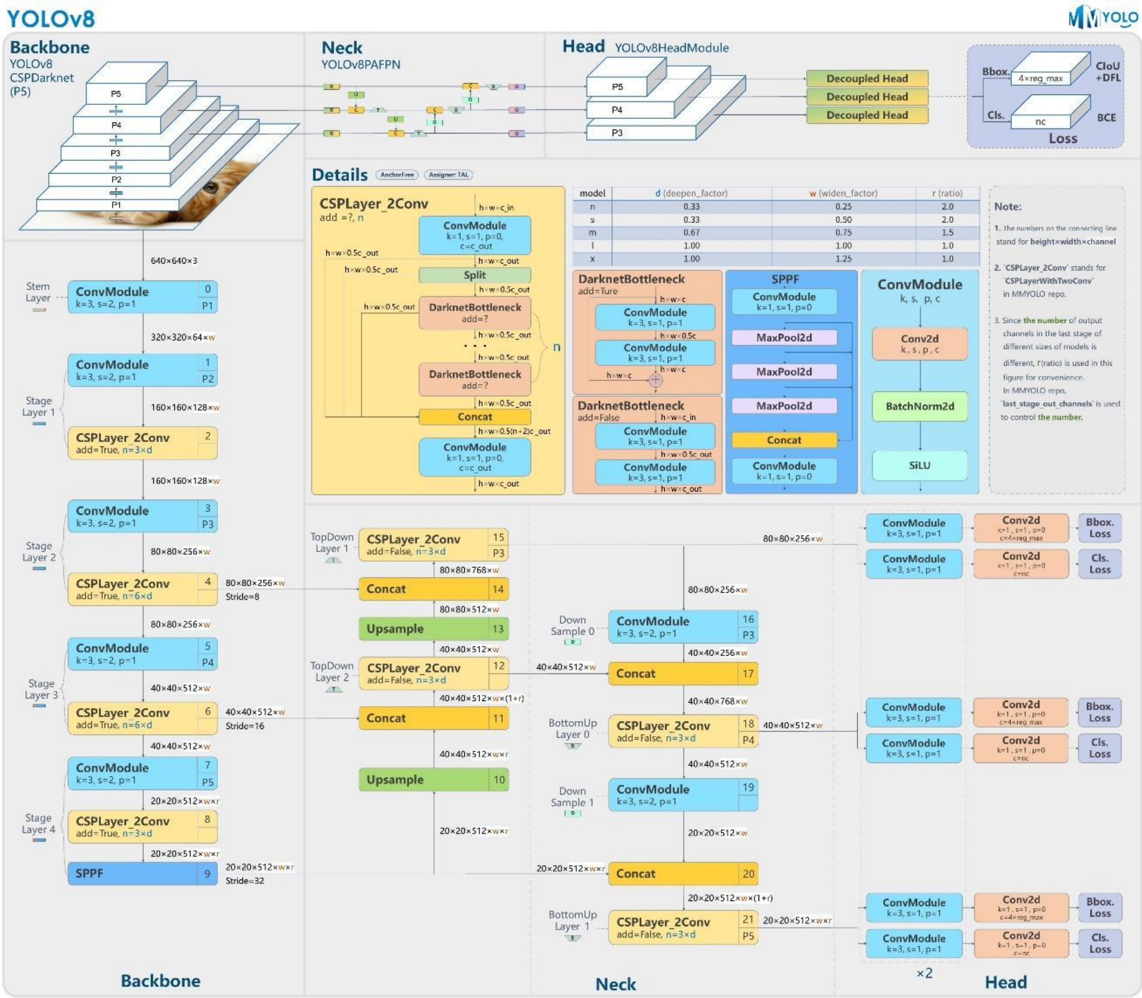

Figure 8: YOLOv8 Architecture [20]

The model needs to go through following steps to be ready for the detection purpose:

3.1.7. Model Evaluation

After training the YOLOv8 model for license plate recognition, various measures were employed to evaluate its performance comprehensively. These included training metrics and validation metrics to assess learning progress and generalization, respectively. Additionally, a confusion matrix was used to visualize the performance across different classes. Precision-Recall (PR) curves, Precision-Confidence (P) curves, and Recall-Confidence (R) curves were generated to understand the trade-offs between precision, recall, and confidence thresholds. Visualization of labeled data and pair plots provided insights into the data distribution and model predictions.

Model Performance Analysis:

The model's performance was evaluated over 25 epochs using a series of training and validation metrics. The analysis of these metrics provides insights into the model's learning process and its ability to generalize to unseen data.

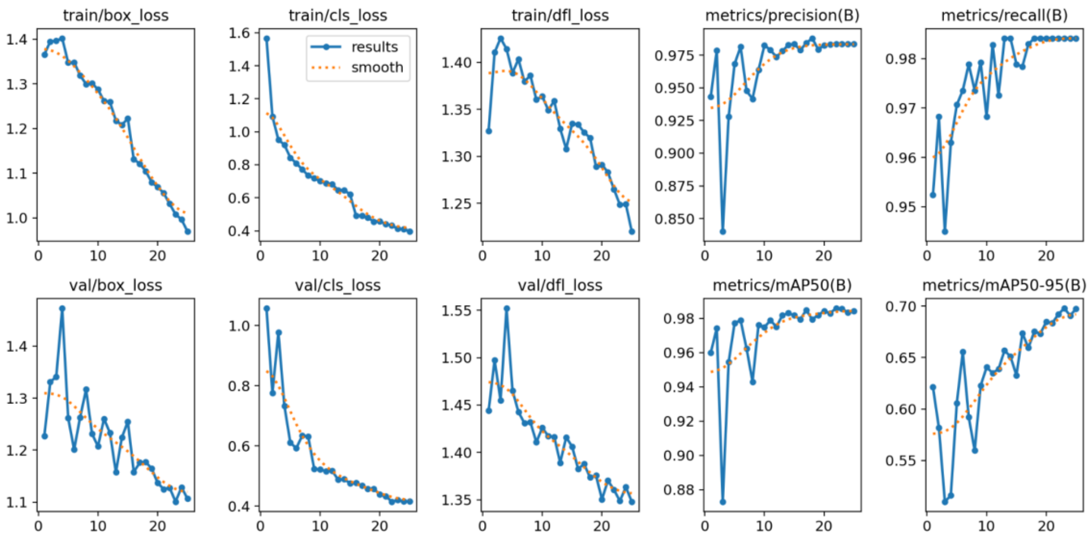

Figure 9: Model Performance Analysis

The key findings are summarized below:

Training Metrics:

1. Box Loss (train/box_loss): The box loss decreased steadily from 1.3657 in the first epoch to 0.96925 by the 25th epoch. This consistent decline indicates that the model is improving its ability to predict bounding boxes accurately.

2. Classification Loss (train/cls_loss):The classification loss saw a significant reduction from 1.567 initially to 0.3956 by the end of the training period. This substantial decrease suggests that the model is becoming more proficient at classifying objects within the bounding boxes.

3. Distributional Focal Loss (train/dfl_loss):The DFL (or similar metric) decreased from 1.327 in the first epoch to 1.22 in the final epoch. This steady decline implies that the model's confidence in its predictions is improving, and it is effectively focusing on more challenging examples.

4. Precision (metrics/precision(B)):Precision improved from 0.94329 at the beginning to 0.98347 by the 25th epoch. This increase indicates that the model's predictions are becoming more accurate, with fewer false positives.

5. Recall (metrics/recall(B)): Recall improved from 0.95238 initially to 0.98413 by the end of the training period. This improvement reflects the model's enhanced ability to detect true positives.

Validation Metrics:

1. Box Loss (val/box_loss): The validation box loss decreased from approximately 1.4 to 1.1, mirroring the trend observed in the training box loss. This indicates that the model generalizes well in predicting bounding boxes for unseen data.

2. Classification Loss (val/cls_loss): The validation classification loss dropped from about 1.0 to 0.4, consistent with the training classification loss. This similarity suggests that the model maintains good classification performance on the validation dataset.

3. Distributional Focal Loss (val/dfl_loss): The validation DFL decreased from around 1.55 to 1.35, following a trend similar to the training DFL. This consistency indicates stable model confidence when applied to validation data.

4. Mean Average Precision at 50% IoU (metrics/mAP50(B)): The mAP50 metric increased from 0.95998 to 0.98433, demonstrating the model's excellent performance in detecting objects with a high degree of accuracy.

5. Mean Average Precision at 50-95% IoU (metrics/mAP50-95(B)): The mAP50-95 metric improved from approximately 0.55 to 0.70, showing solid overall performance across varying levels of overlap between predicted and ground truth boxes.
6. Confusion Matrix Analysis: From the obtained confusion matrix, we can extract the following values:

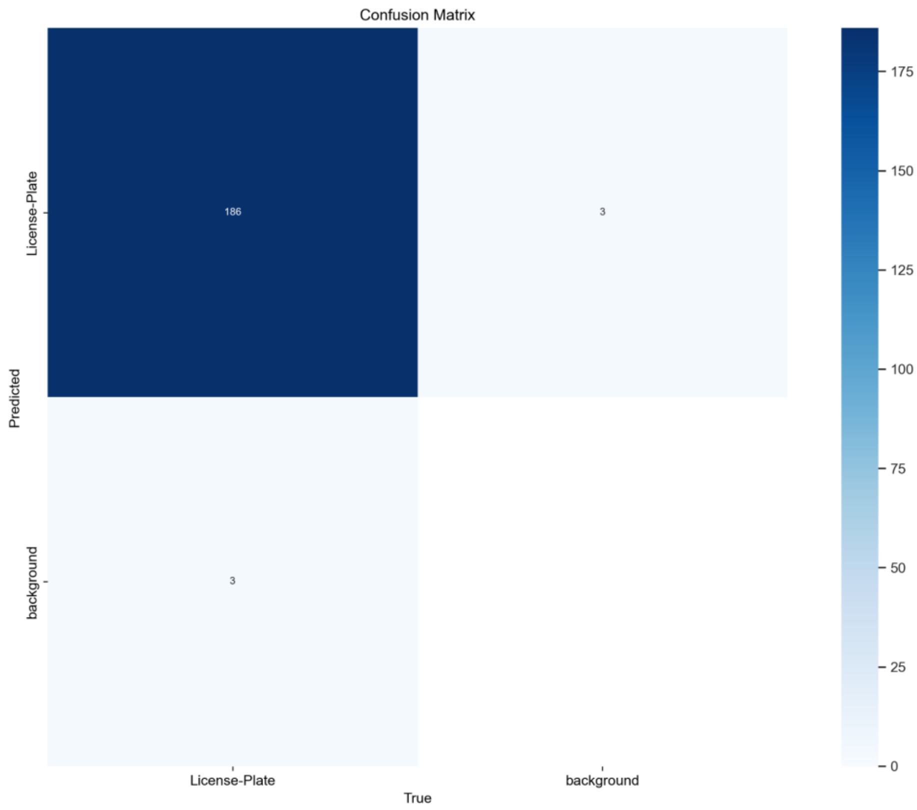

Figure 10: Confusion Matrix

True Positives (TP): 186 (Predicted License Plate correctly)

False Positives (FP): 3 (Predicted License Plate incorrectly)

False Negatives (FN): 3 (Missed License Plate)

True Negatives (TN): Not explicitly shown but can be inferred

7. Normalized Confusion Matrix Analysis: The normalized confusion matrix values are as follows:

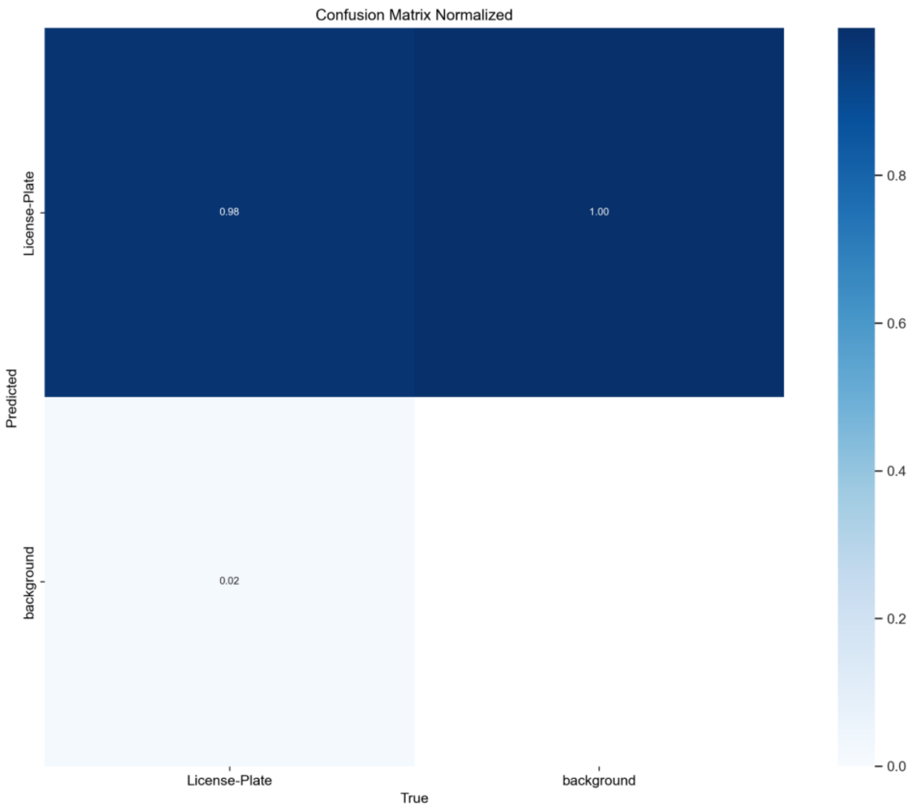

Figure 11: Confusion Matrix Normalized

License Plate (Predicted as License Plate): 0.98

License Plate (Predicted as Background): 0.02

Background (Predicted as License Plate): 0.00

Background (Predicted as Background): 1.00

The model demonstrates a robust learning process and effective performance improvement across all evaluated metrics.

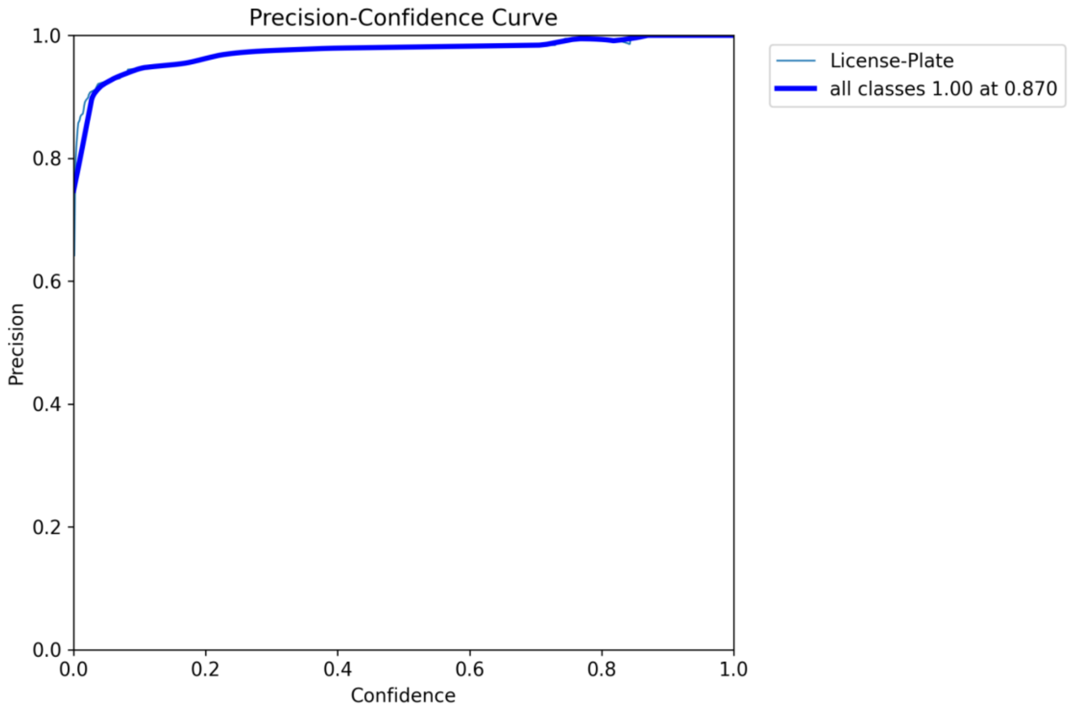

Figure 12: Precision-Confidence Curve

The Precision-Confidence Curve indicated that the YOLOv8 model achieved high precision at higher confidence thresholds, particularly at 0.870 where it reached perfect precision (no false positives), making it highly reliable for license plate detection with a trade-off against recall.

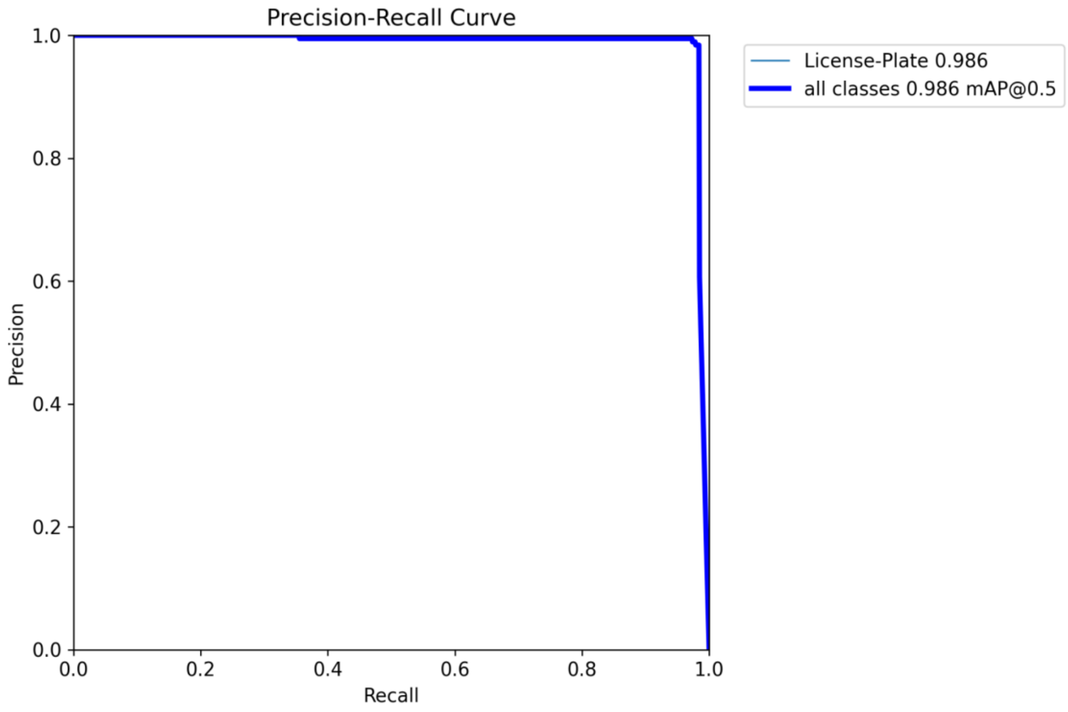

Figure 13: Precision-Recall Curve

The Precision-Recall Curve showed a high precision of 0.986 across nearly the entire range of recall, indicating excellent model performance with minimal trade-offs between precision and recall.

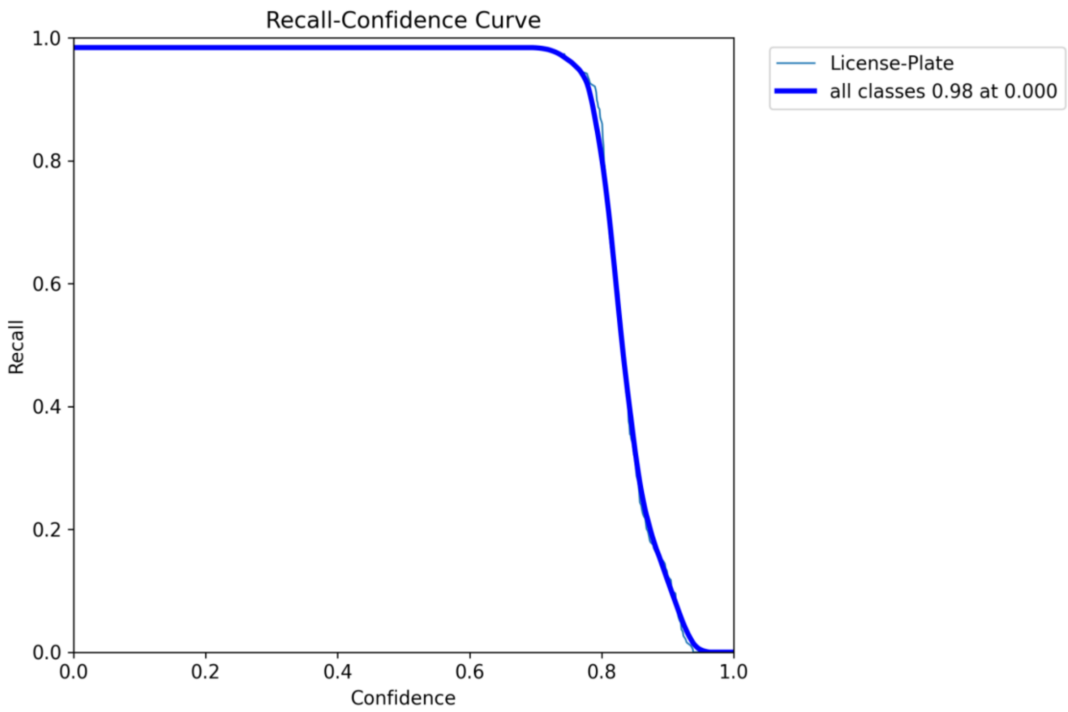

Figure 14: Recall-Confidence Curve

The Recall-Confidence Curve indicated that high recall was achieved at low confidence thresholds, but recall sharply decreased at higher thresholds, necessitating a balance between recall and precision based on application requirements.

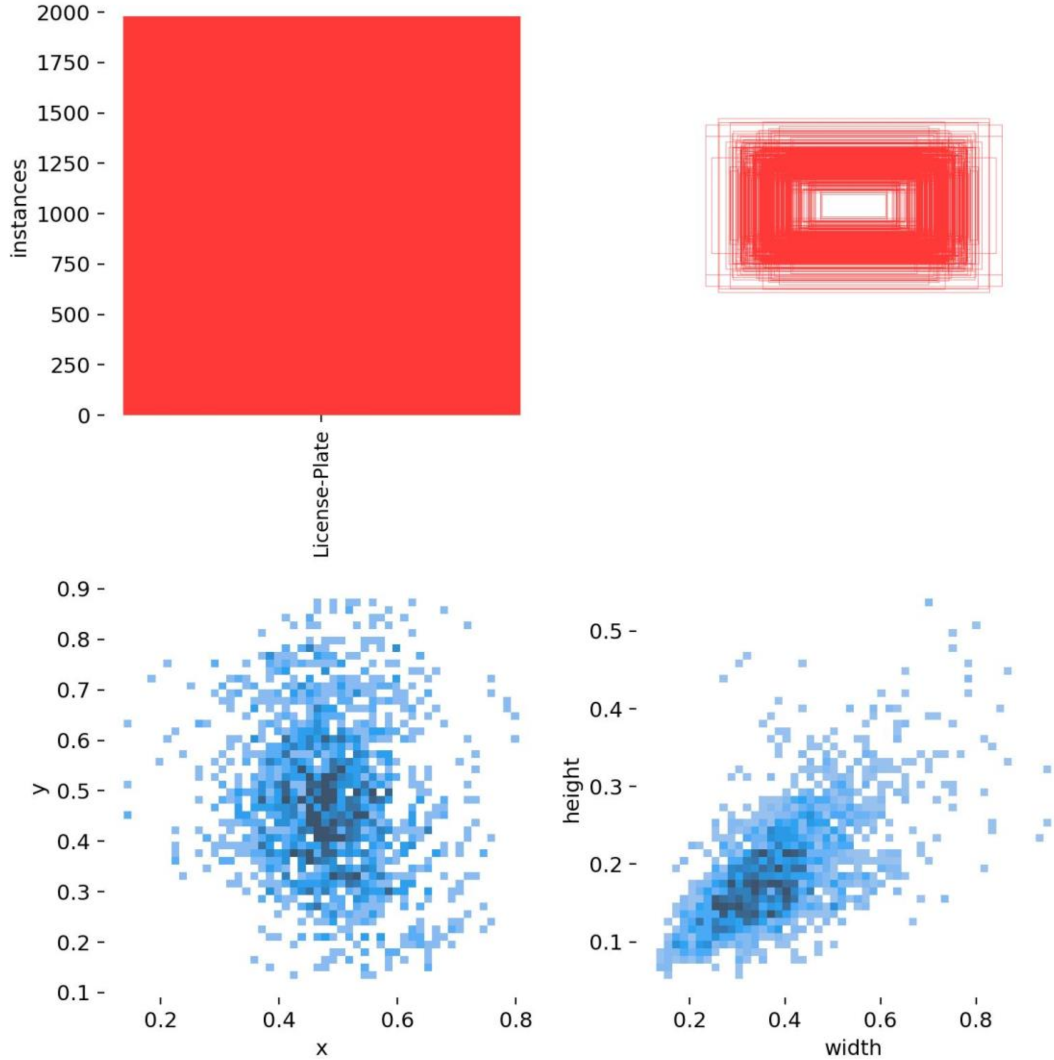

Figure 15: Visualization of labeled data

The figure contains 4 parts:

Top-Left (Red Square Plot)

Instances vs. Class: The top-left plot shows the number of instances for each class in the dataset. In this case, it indicates that there are around 2000 instances of the "License-Plate" class.

Top-Right (Red Rectangle Plot)

Bounding Box Distribution: The top-right plot shows the distribution of bounding boxes for the license plates. Each red rectangle represents a bounding box around a detected license plate in the dataset. The concentration and overlap of rectangles indicate the common size and position of the license plates in the images. This helps in understanding the general location and size of the license plates that the model needs to learn.

Bottom-Left (Blue Scatter Plot)

Center of Bounding Boxes (x, y): The bottom-left plot shows the distribution of the centers of bounding boxes (x, y coordinates). The x-axis represents the horizontal position (normalized) of the bounding box centers, and the y-axis represents the vertical position. The concentration of points in the middle suggests that most license plates are located towards the center of the images, which is typical for many detection datasets.

Bottom-Right (Blue Scatter Plot)

Width and Height of Bounding Boxes: The bottom-right plot shows the distribution of the width and height of the bounding boxes (both normalized). The x-axis represents the width of the bounding boxes, and the y-axis represents their height. The concentration of points indicates the common sizes of the license plates in your dataset. For example, if most points are concentrated around a certain width and height, it suggests that license plates have a relatively consistent size across the dataset.

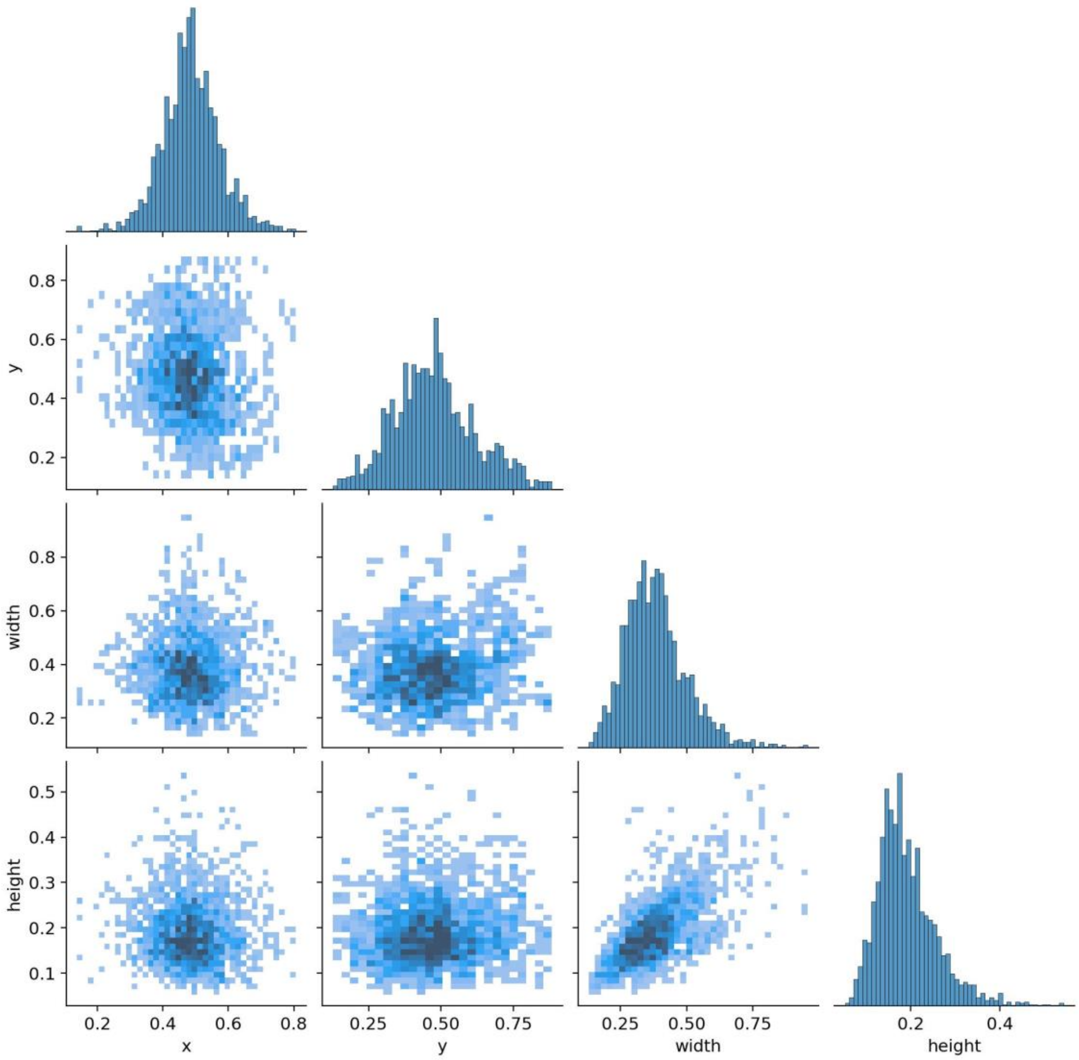

Figure 16: Pair Plot

This image is a pair plot, which is often used to visualize the relationships between multiple variables (x, y, height, width).

3.1.8. Post-Processing

In the post-processing Implement post-processing techniques will be done to refine the model's predictions. This may include filtering out false positives or smoothing the bounding box predictions for more accurate and stable results [16].

3.1.9. Optimization and Hyperparameter Tuning

This step involves fine-tuning the model and its hyperparameters based on the evaluation results. ANPR system for performance, accuracy, and efficiency. At the end of this step, a fine-tuned, more accurate and efficient YOLO model will be obtained which will be used for number plate detection [14].

3.1.10. Optical Character Recognition (OCR)

OCR (Optical Character Recognition) will be a crucial component for identifying the characters in the number plate in this project. Once the ANPR system detects the region of interest containing the license plate using object detection, OCR will be applied to recognize and extract the alphanumeric characters from the detected plate [21].

3.2. Parking Management Software Development

The web-based platform will be developed by using frontend and backend technologies which are described below:

3.2.1. Html

HTML, which stands for Hypertext Markup Language, is the standard markup language used to create and design documents on the World Wide Web. It will form the backbone of web content by providing a structured way to describe the elements on a web page, such as text, images, links, forms, and multimedia [22].

3.2.2. CSS

CSS, which stands for Cascading Style Sheets, is a style sheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG or XHTML). So, CSS will allow us to control the layout, formatting, and appearance of web pages [23].

3.2.3. React JS

React JS, commonly referred to as React, is an open-source JavaScript library for building user interfaces. Developed and maintained by Facebook, react is widely used for creating dynamic and interactive web applications with a focus on efficient updates to the user interface. React will allow us to build reusable UI components and manage the state of an application in a more organized and predictable way [24].

3.2.4. Node JS

Node.js is an open-source, server-side JavaScript runtime environment built on the V8 JavaScript engine developed by Google for use in Chrome. It allows developers to use JavaScript to write server-side code, enabling the execution of JavaScript code on the server. Node.js is designed to be lightweight, efficient, and scalable, making it well-suited for building fast and scalable network applications [25].

In this project, a web-based application for parking space detection and license plate recognition has been developed using deep learning and OCR.

The frontend of this application has been built using HTML, CSS and JavaScript. This report outlines how each of these technologies has been utilized to create a responsive, interactive and visually appealing user interface.

HTML is the backbone of any web application. It structures the content and provides the basic framework of the website.

CSS is used to style the HTML elements, making the web page visually appealing and responsive.

JavaScript is used to add interactivity to the web page. It handles events, such as form submissions and dynamically updates the content.

3.3. Application Interface and User Experience

3.3.1. Home Page

The central hub of the application where users can upload images for parking space detection, view the results and view parking charges.

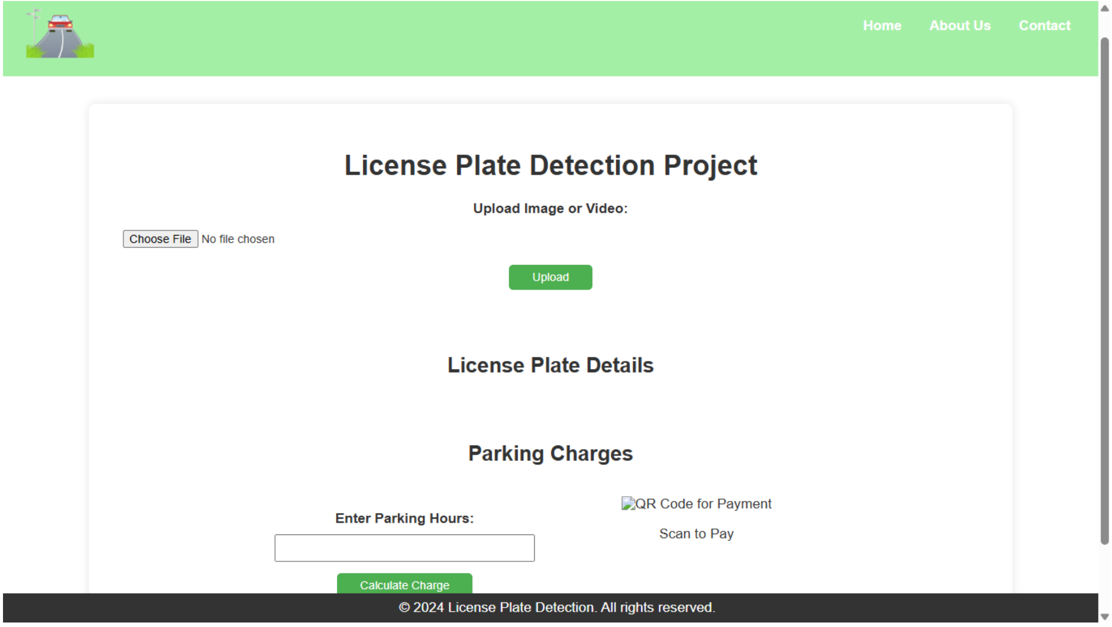

Figure 17: Home Page of User Interface

3.3.2. About Us and Contact Us Pages:

About Us and Contact Us Pages have been built where user can view information about the team or organization behind the application, outlining the purpose, mission, contact details and background of the project.

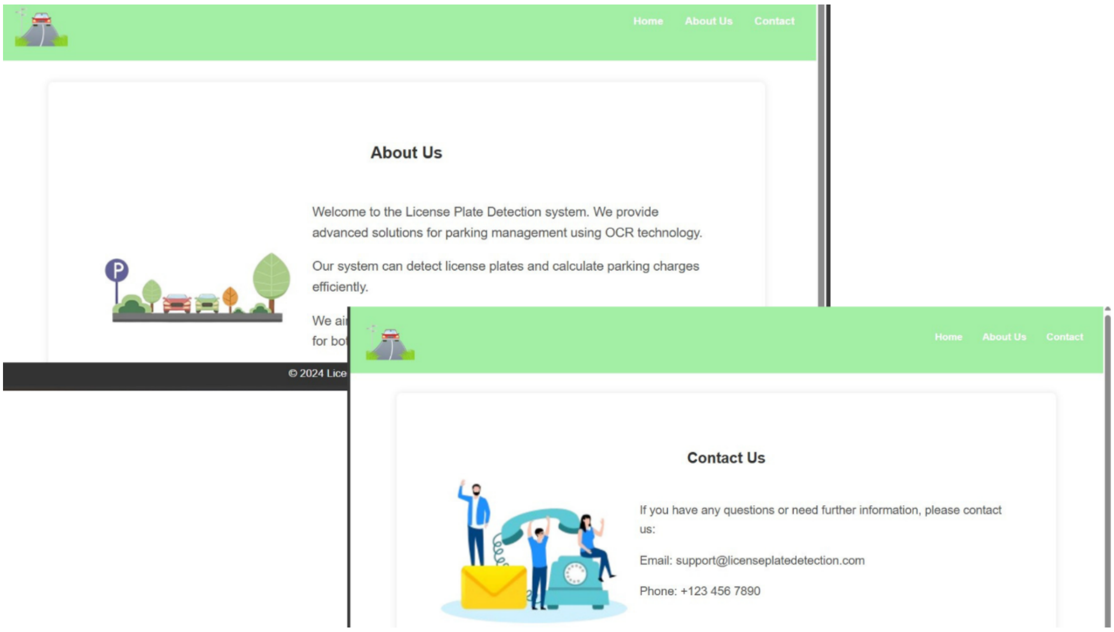

Figure 18: Contact Us and About Us pages

3.4. Database Integration

A database will be integrated to store information about recognized license plates, parking availability, and other relevant data.

3.5. Testing and Deployment

The entire system will be thoroughly tested to ensure seamless integration and functionality. Once testing is successful, the Smart Parking Management System with Automatic Number Plate Recognition will be ready for real-world use.

Chapter 4: Task Done and Task Remaining

4.1. Task Done

1. Collected a large number of clear pictures (942) of vehicle license plates in various situations.
2. The images were cropped and annotated by marking bounding boxes around the license plates using Roboflow.
3. The 942 images were splitted into 661 images for training, 188 images for validation, and 94 images for testing i.e 70% , 20% and 10% of total dataset, respectively.
4. Images preprocessed to Auto-orient, resize to 640*640, and augmented with rotations, grayscale, brightness, blur and noise. Training set tripled to 1983 images from various transformations.
5. The YOLOv8 model was trained using the annotated and preprocessed dataset obtained from the previous steps.

4.2. Task Remaining

1. The model architecture to be fine-tuned if needed, considering the specifics of license plate recognition in parking scenarios.
2. Post-processing techniques to refine the model's predictions and fine-tuning the model and its hyperparameters based on the evaluation results.
3. The task for Optical Character Recognition is still remaining which will be used to recognize and extract the alphanumeric characters from the detected plate.
4. Building a web-based platform using the following frontend and backend technologies is still remaining.
5. Task of Integration of the trained model to the web based platform is still remaining.

Chapter 5: Result and discussion

The YOLOv8 model for license plate recognition demonstrated significant improvements and robust performance through comprehensive evaluation and analysis. This section details the key findings and their implications, structured to cover dataset preparation, model training, and performance metrics.

The initial stages of data collection and preparation involved capturing 942 images of vehicles in various lighting conditions and scenarios, ensuring a diverse and representative dataset. The dataset was meticulously prepared, with images cropped for clear visibility of license plates and annotated using Roboflow to create bounding boxes around the plates.

Upon splitting the dataset into training (70%), validation (20%), and testing (10%) sets, a series of preprocessing and augmentation steps were applied. Images were auto-oriented and resized to 640x640 pixels, and augmentations like rotations, grayscale application, brightness adjustment, blur, and noise addition were employed. This augmentation increased the training dataset to 1983 images, enhancing the model's robustness by exposing it to various transformations.

During model training, the YOLOv8 model showed substantial progress. Training metrics indicated a consistent decrease in box loss from 1.3657 to 0.96925, classification loss from 1.567 to 0.3956, and Distributional Focal Loss (DFL) from 1.327 to 1.22 over 25 epochs. Precision and recall improved significantly, from 0.94329 to 0.98347 and 0.95238 to 0.98413, respectively. These metrics suggest that the model became more proficient at accurately predicting bounding boxes and classifying objects within them.

Validation metrics mirrored the positive trends seen in training metrics, with validation box loss decreasing from 1.4 to 1.1, classification loss from 1.0 to 0.4, and DFL from 1.55 to 1.35. The mean Average Precision at 50% IoU (mAP50) increased from 0.95998 to 0.98433, and at 50-95% IoU (mAP50-95) from approximately 0.55 to 0.70, demonstrating excellent model performance in detecting objects accurately across different overlap levels.

The confusion matrix analysis showed high true positive rates (TP: 186) and minimal false positives (FP: 3) and false negatives (FN: 3), indicating the model's reliability in license plate detection. The normalized confusion matrix further highlighted the model's precision, with 98% of license plates correctly predicted and no significant misclassifications.

Visualizations of labeled data provided insights into the dataset and model predictions. Plots showed the distribution of bounding boxes, centers of bounding boxes, and width and height of bounding boxes, confirming that most license plates were centrally located and of consistent size across the dataset.

Overall, the YOLOv8 model demonstrated strong performance and reliable predictions for license plate recognition. The comprehensive evaluation of training and validation metrics, combined with the detailed analysis of confusion matrices and visualizations, confirmed the model's robustness and effectiveness in real-world applications.

Chapter 6: Epilogue

6.1. Time Estimation

Before getting started with any project, we must prepare a working schedule consisting of several topics that we would be working on throughout the project development phase. For the same reason, the following is the Gantt chart representing our work schedule in a total span of 3 months, i.e., 10 weeks (about 2 and a half months) ranging from the phase after proposal defense to final report submission and defense:

Table 1: Gantt Chart of the project

S.No.	Activities	Poush		Magh		Chaitra	Baisakh				Jestha
		3rd	4th	1st	2nd	4th	1st	2nd	3rd	4th	1st
1.	Literature review										
2.	Preparation of proposal										
3.	Proposal Defense										
4.	Coding initiation										
5.	Mid-Term Presentation										
6.	Coding Continuation										
7.	Report preparation										
8.	Final defense of project report										

- Poush 3rd – Paush 4th Week: At the very beginning of our project, we took some time to decide the topic of our project. After deciding what topic, we are going to work on, we then started reviewing the various literature (project reports, articles, books, journals etc.) related to our topic.

- Magh 1st Week: After reviewing various literature reviews, we started writing the proposal for our project.

- Magh 2nd Week: After submitting our project proposal, we will defend it on the second week of Magh.

- Chaitra 4th – Baisakh 2nd Week: We will be focusing on building the YOLO model, user interface and integrating model, choosing a web development framework, and designing the interface. We will do a mid -term presentation of our project on the 2nd week of Baisakh.

- Baisakh 3rd – Baisakh 4th Week: We will be focusing on developing the backend code, testing the web application, and finalizing it before launch.

- Jestha 1st Week: Finally, at the end of the 10th week, we will complete our final project report and submit it to the respective department.

6.2. Budget Estimation

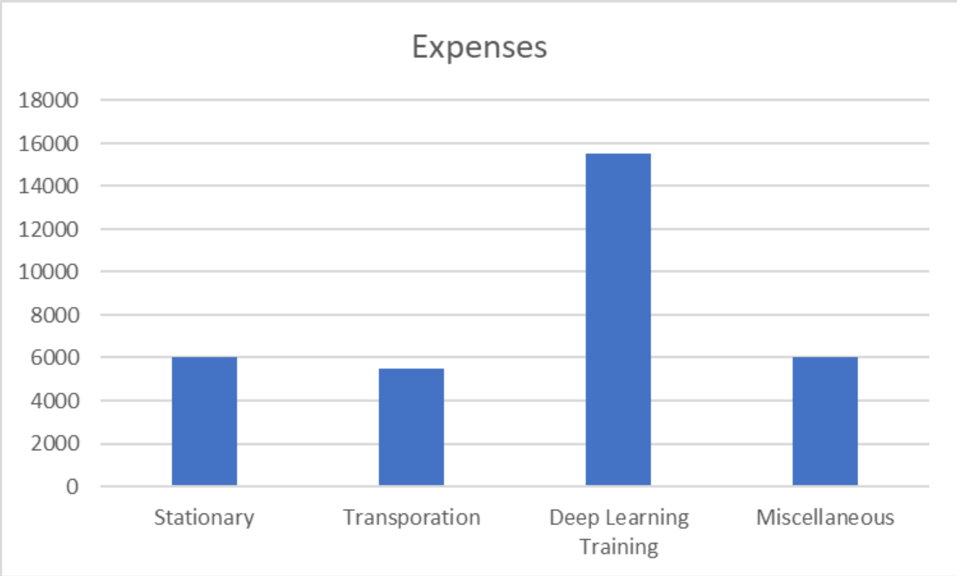

Figure 19: Budget Estimation

Chapter 7: Expected outcome

The proposed Parking Management System with Automatic Number Plate Recognition (ANPR) aims to revolutionize and optimize the parking experience. The main goal is to make parking easier and more organized. The outcome we expect is a smart system that can quickly and accurately recognize license plates when vehicles enter or leave a parking area. We'll use advanced technology (Deep Learning) to make this recognition process fast and reliable. The system will also have an easy-to-use interface for both the people managing the parking and the users. This will help with tasks like monitoring the parking area in real-time and handling payments automatically. Our project aims to reduce the need for manual work, make parking operations smoother, prevent unauthorized parking, and ultimately make the whole parking experience better for everyone.

References

[1] R. Tater, P. Nagrath, J. Mishra, V. H. de Albuquerque, and J. W. Menezes, “Smart parking system using Yolov3 Deep Learning Model,” *Proceedings of Data Analytics and Management*, pp. 387–398, 2023. doi:10.1007/978-981-99-6550-2_30

[2] S. Gopikrishnan, A. K. Madduru, K. Karamsetty, and D. R. Ravuri, “Smart parking system with automated vehicle log using Haar Cascade Classifier ANPR,” *IFIP Advances in Information and Communication Technology*, pp. 266–286, 2023. doi:10.1007/978-3-031-38296-3_21

[3] A. Menon and B. Omman, “Detection and recognition of multiple license plate from still images,” *2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*, 2018. doi:10.1109/iccsdet.2018.8821138

[4] N. Darapaneni *et al.*, “Computer Vision based license plate detection for Automated Vehicle Parking Management System,” *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2020. doi:10.1109/uemcon51285.2020.9298091

[5] P. R. Dawadi, B. K. Bal, and M. Pokharel, “Devanagari license plate detection, classification and recognition,” *Proceedings of the First International Conference on Advances in Computer Vision and Artificial Intelligence Technologies (ACVAIT 2022)*, pp. 304–318, 2023. doi:10.2991/978-94-6463-196-8_24

[6] B. Setiyono, D. A. Amini, and D. R. Sulistyaningrum, “Number Plate recognition on vehicle using YOLO - Darknet,” *Journal of Physics: Conference Series*, vol. 1821, no. 1, p. 012049, 2021. doi:10.1088/1742-6596/1821/1/012049

[7] M. M. Abdellatif, N. H. Elshabasy, A. E. Elashmawy, and M. AbdelRaheem, “A low cost IOT-based Arabic license plate recognition model for Smart Parking Systems,” *Ain Shams Engineering Journal*, vol. 14, no. 6, p. 102178, 2023. doi:10.1016/j.asej.2023.102178

[8] M. S. Farag, M. M. Mohie El Din, and H. A. Elshenbary, “Parking entrance control using license plate detection and recognition,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 15, no. 1, p. 476, 2019. doi:10.11591/ijeecs.v15.i1.pp476-483

[9] H. Moussaoui, N. E. AKKAD, and M. BENSLIMANE, *Arabic and Latin license plate detection and recognition based on Yolov7 and image processing methods*, 2023. doi:10.21203/rs.3.rs-3195386/v1

[10] S. H. Fahim, A. A. Shafi Rasel, A. R. Sarker, T. Chowdhury, and E. L. Rimban, *Utilizing yolo V8 for automated license plate detection of vehicles in Bangladesh: An approach for vehicle identification*, 2024. doi:10.33774/coe-2024-gxw4j

[11] V. Gnanaprakash, N. Kanthimathi, and N. Saranya, "Automatic number plate recognition using Deep Learning," *IOP Conference Series: Materials Science and Engineering*, vol. 1084, no. 1, p. 012027, 2021. doi:10.1088/1757-899x/1084/1/012027

[12] R. Laroca *et al.*, "An efficient and layout-Independent Automatic License Plate Recognition System based on the Yolo Detector," *IET Intelligent Transport Systems*, vol. 15, no. 4, pp. 483–503, 2021. doi:10.1049/itr2.12030

[13] S. Kaur, "An automatic number plate recognition system under image processing," *International Journal of Intelligent Systems and Applications*, vol. 8, no. 3, pp. 14–25, 2016. doi:10.5815/ijisa.2016.03.02

[14] N. A. Jalil, A. S. Basari, S. Salam, N. K. Ibrahim, and M. A. Norasikin, "The utilization of template matching method for license plate recognition: A case study in Malaysia," *Lecture Notes in Electrical Engineering*, pp. 1081–1090, 2014. doi:10.1007/978-3-319-07674-4_100

[15] W. W. Keong and V. Iranmanesh, "Malaysian automatic number plate recognition system using Pearson correlation," *2016 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2016. doi:10.1109/iscaie.2016.7575034

[16] D. Neupane *et al.*, "Shine: A Deep Learning-based accessible parking management system," *Expert Systems with Applications*, vol. 238, p. 122205, 2024. doi:10.1016/j.eswa.2023.122205

[17] W. by: E. Zvornicanin, "What is Yolo Algorithm?," Baeldung on Computer Science, https://www.baeldung.com/cs/yolo-algorithm (accessed Jan. 22, 2024).

[18] "Yolo algorithm for object detection explained [+examples]," YOLO Algorithm for Object Detection Explained [+Examples], https://www.v7labs.com/blog/yolo-object-detection (accessed Jan. 22, 2024).

[19] Z. Keita, "Yolo Object Detection explained: A beginner's guide," DataCamp, https://www.datacamp.com/blog/yolo-object-detection-explained (accessed Jan. 22, 2024).

[20] M. Hussain, "Yolo-V1 to Yolo-V8, the rise of Yolo and its complementary nature toward digital manufacturing and industrial defect detection," *Machines*, vol. 11, no. 7, p. 677, 2023. doi:10.3390/machines11070677

[21] M. Simon, B. Shibwabo, and K. Mutua, "An automatic number plate recognition system for car park management," *International Journal of Computer Applications*, vol. 175, no. 7, pp. 36–42, 2017. doi:10.5120/ijca2017915608

[22] "What is HTML (hypertext markup language)?," Computer Hope, https://www.computerhope.com/jargon/h/html.htm (accessed Jan. 19, 2024).

[23] CSS tutorial, https://www.w3schools.com/css/ (accessed Jan. 19, 2024).

[24] C. Deshpande, “What is react? [easily explained],” Simplilearn.com, https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs#:~:text=What%20is%20React%20JS%20used,is%20the%20JavaScript%20virtual%20DOM. (accessed Jan. 19, 2024).

[25] “Node.js - about node.js®,” Node.js -, https://nodejs.org/en/about (accessed Jan. 19, 2024).