Final Report On

**Number Plate Detection**

****

Submitted to

**Department of Computer Science and Engineering**

**Nepal Engineering College**

IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF B.E. IN COMPUTER

By

Aashish Roka (020-301)

Bhagwat Kumar Shah (020-312)

Reeya Mishra (020-332)

**Supervised by**

Asst.Prof. Deepak Bahadur Dhami

Date**:** 07/16/2024

Abstract

In this project we used YOLOv8n that was trained on custom dataset collected by us which consisted of 600 images of 2 classes which was augmented to extend our dataset to 1200 images and was split in the ratio of 70:20:10 for train, validation, and test respectively. For tracking the detected objects in the video, we used DeepSORT which tracks and outputs the bounding box for the object with respective track IDs. Then the detected and tracked license plate of the object is sent as input for segmentation program. The image of the license plate undergoes HSV color space conversion, color masking and perspective transformed in that order before it is preprocessed for profiling the different types of license plate in the dataset. The segmented characters are then fed to a CNN trained on our custom dataset of characters of license plate, for now which consists of only Embossed Number Plate.

Acknowledgement

We would like to express our sincere gratitude to all those who have contributed to the completion of this report on the number plate detection system. First and foremost, we extend our deepest appreciation to our supervisor Asst.Prof. Deepak Bahadur Dhami sir for his invaluable guidance, support, and encouragement throughout the duration of this project. Their expertise and insightful feedback have been instrumental in shaping our report. We are also thankful to the faculty members of Nepal Engineering College, whose teachings and mentorship have provided us with a solid foundation of knowledge and skills in the field of computer vision and machine learning. Furthermore, we extend our thanks to our colleagues and friends for their assistance, encouragement, and constructive criticism, which have helped refine our ideas and improve the quality of this report.

This report would not have been possible without the collective efforts of all those mentioned above. Thank you.

Table of Contents

[Abstract I](#_Toc172000870)

[Acknowledgement II](#_Toc172000871)

[List of Figures V](#_Toc172000872)

[List of Table VI](#_Toc172000873)

[1 Introduction 1](#_Toc172000874)

[1.1 Overview 1](#_Toc172000875)

[1.2 Problem statement 2](#_Toc172000876)

[1.3 Objectives 3](#_Toc172000877)

[1.4 Aim 3](#_Toc172000878)

[1.5 Motivation 3](#_Toc172000879)

[1.6 Scope and Application 4](#_Toc172000880)

[1.7 Feasibility Study 5](#_Toc172000881)

[1.7.1 Technology Requirements 5](#_Toc172000882)

[1.7.2 Operational Feasibility 5](#_Toc172000883)

[1.7.3 Economic Feasibility 5](#_Toc172000884)

[2 Literature Review 6](#_Toc172000885)

[2.1 YOLOv8 Architecture 8](#_Toc172000886)

[3 System Design 9](#_Toc172000887)

[3.1 Data Acquisition 9](#_Toc172000888)

[3.2 Data Preprocessing 10](#_Toc172000889)

[3.3 Number plate segmentation 10](#_Toc172000890)

[3.4 Data Augmentation 10](#_Toc172000891)

[3.5 OCR implementation 11](#_Toc172000892)

[3.6 Hardware and Software requirements 12](#_Toc172000893)

[3.6.1 Hardware 12](#_Toc172000894)

[3.6.2 Software 12](#_Toc172000895)

[4 Implementation and Discussion 13](#_Toc172000896)

[4.1 Methodology 13](#_Toc172000897)

[4.2 Implementation Steps 14](#_Toc172000898)

[4.2.1 Model Training 14](#_Toc172000899)

[4.2.2 Tracking 14](#_Toc172000900)

[4.2.3 Processing ROI 14](#_Toc172000901)

[4.2.4 Character Segmentation 15](#_Toc172000902)

[4.2.5 Character Pre-processing 16](#_Toc172000903)

[4.2.6 Data storage 16](#_Toc172000904)

[4.3 Output Obtained 18](#_Toc172000905)

[4.4 Time Schedule 18](#_Toc172000906)

[5 Analysis and Evaluation 19](#_Toc172000907)

[5.1 Data Analysis 19](#_Toc172000908)

[5.1.1 Graphs 19](#_Toc172000909)

[5.1.2 Confusion Matrix 21](#_Toc172000910)

[5.2 Comparison with the Objectives 22](#_Toc172000911)

[5.3 Discussion and findings 22](#_Toc172000912)

[6 Conclusion and future work 23](#_Toc172000913)

[6.1 Future Work 23](#_Toc172000914)

[7 Bibliography 24](#_Toc172000915)

List of Figures

[Figure 3‑1 Object Detection 10](#_Toc171990144)

[Figure 4‑1 Project Methodology 12](#_Toc171990145)

[Figure 4‑2 Pre-processing Detected License Plate ROI 14](#_Toc171990146)

[Figure 4‑3 Different phases of detecting character in LP 15](#_Toc171990147)

[Figure 4‑4 Image of character before and after applying padding 15](#_Toc171990148)

[Figure 5‑1 Graphs 17](#_Toc171990149)

[Figure 5‑2 Correlogram of Labels in Our Model 18](#_Toc171990150)

[Figure 5‑3 Confusion Matrix 19](#_Toc171990151)

List of Table

[Table 1: Gantt Chart 15](#_Toc166220167)

List of Abbreviation

AI Artificial Intelligence

OpenCV Open-source computer vision Library

R-CNN Region based Convolutional Neutral Network

CCTV Closed Circuit Television

CNNs Convolution Nural Network

OCR Optical Character Recognition

Chapter 1

# Introduction

## Overview

Computer vision, a field of artificial intelligence (AI), focuses on enabling computers to interpret and understand the visual world, mimicking the complexity of the human visual system to identify and analyze objects in images and videos similarly to human perception. Historically, computer vision's capabilities were limited by early algorithms and computational power, allowing only basic image processing and simple object detection. However, recent advancements in AI, particularly deep learning and neural networks, have transformed computer vision. Convolutional Neural Networks (CNNs) and other deep learning models have dramatically improved the accuracy and efficiency of image recognition and object detection tasks, enabling AI systems to outperform humans in these areas.

The number of vehicles on the road is steadily increasing, especially in emerging nations like ours. The likelihood of breaking traffic laws, resulting in unanticipated accidents, and initiating traffic crimes has increased due to the substantial use of vehicles. An intelligent traffic monitoring system is necessary to solve these issues. The intelligent system's ability to recognize vehicle license plates can be extremely useful in traffic control. This project uses a convolutional neural network, OCR, image annotation, and a large variety of Python tools to construct a system for identifying and recognizing vehicles number plates.

The automatic detection of number plate is a crucial project within the broader field of pattern recognition. Various disciplines such as architecture, cartography, electronics, and engineering heavily rely on domain-specific graphic notations to articulate their designs. The challenge lies in developing processes capable of automatically interpreting the provided images of vehicles or traffic, requiring systems that can recognize the corresponding segments of a Nepali number plate. The significance of number plate recognition extends beyond individual disciplines, offering potential applications in pattern recognition, automation of design processes, and even aiding in the interpretation of graphical content for individuals with visual impairments. The objective of this project is to develop a real-time number plate detection system using the YOLO (You Only Look Once) object detection algorithm and the OpenCV (Open-Source Computer Vision Library) framework. The system is designed to detect and recognize vehicle number plates from images or video streams, facilitating applications such as automated toll collection, traffic monitoring, and security surveillance.

## Problem statement

* **Identifying a Critical Need**: Recognizing the technological gap in our developing nation, we've prioritized a project aimed at automating the identification of number plates. This will streamline the process of retrieving vehicle data from images or video frames.
* **Acknowledging Resource Constraints**: We understand the challenge of obtaining a large and diverse dataset of number plates specific to Nepal. Given the complexity of patterns and variations in number plate designs, especially in segments, this poses a significant obstacle for small-scale projects like ours.
* **Utilizing Existing Resources**: To address this challenge, we plan to leverage preexisting models and the latest tools available. By harnessing these resources, we aim to enhance the accuracy and credibility of our solution, even with limited access to extensive training data.
* **Addressing a Nationwide Issue**: The absence of automatic license plate recognition technology across many high-traffic areas in our country has contributed to increased crime rates and violations of traffic regulations. Our project seeks to provide an affordable yet precise software solution to these regions.
* **Impact on Crime and Safety**: By implementing our software in areas with high traffic volume, we anticipate a significant reduction in crime rates related to traffic violations and accidents. The automation of license plate recognition will facilitate law enforcement efforts and enhance overall road safety.
* **Contributing to National Development**: Our project aligns with broader national goals of leveraging technology to address societal challenges. By deploying innovative solutions tailored to local needs, we aim to contribute to the progress and safety of our nation's transportation infrastructure.

## Objectives

The main objectives of this project are as follows:

* To recognize motorcycle, car, bus, truck, and Embossed license plates.
* To recognize the characters of detected license plates.
* To store the detected information in a specific format.

## Aim

The aim of the number plate detection project is to create a software capable of accurately and automatically identifying the information on the segments of number plate of Nepali vehicle from an image or a frame from a video and provide information regarding that vehicle to the user.

## Motivation

Number plate detecting application advances the fields of computer vision, machine learning, and artificial intelligence. It gives students a deep understanding of pattern recognition in image and how to translate those patterns into numerical representations. In addition, it gives them a terrific new tool that allows them to explore new ideas simply by giving them access to an image.

## Scope and Application

The field of number plate detection has a wide range of uses. By automating the toll collecting process i.e. by using license plate recognition to automate toll booths. It will aid in traffic management by lowering wait times and congestion. In regions with high traffic, it will also be useful for traffic monitoring. In addition, it will support law enforcement organizations in managing traffic infractions, tracing stolen automobiles, and identifying vehicles implicated in criminal activity.

Automatic Number Plate Recognition (ANPR) systems are sophisticated technologies designed to read vehicle registration plates using optical character recognition. These systems have found widespread use in various domains, significantly improving efficiency, security, and operational effectiveness.

In toll collection booths, ANPR systems automate the toll payment process by identifying vehicles through their license plates as they pass through. This automation not only speeds up toll collection by eliminating manual processing but also reduces congestion and waiting times at toll plazas. The accuracy of ANPR ensures that toll charges are correctly applied, reducing the potential for revenue loss due to human error or fraudulent activities.

Parking management in commercial settings such as hotels, malls, airports, and office buildings has also been revolutionized by ANPR systems. They automate the entry and exit of vehicles, replacing the need for physical tickets or access cards. This leads to quicker processing times and more accurate billing based on the duration of stay. Additionally, the ability to maintain detailed records of all vehicles entering and exiting enhances security, making it easier to track any incidents that might occur.

In the realm of commercial fleet management, ANPR systems assist businesses in optimizing their operations. They provide real-time tracking of fleet vehicles, which helps in route optimization, improving delivery times, and reducing fuel consumption. Maintenance schedules can be better managed by tracking vehicle usage, thus preventing unexpected breakdowns and extending the life of the vehicles. Continuous monitoring also aids in theft prevention and quick recovery of stolen vehicles.

## Feasibility Study

### Technology Requirements

* **Hardware:** High-resolution cameras, processing servers.
* **Software:** Image processing, OCR, tracking algorithms, data logging tools.

### Operational Feasibility

* **Integration:** Fits well with existing security and traffic systems.
* **Training**: Minimal training required for operators.
* **Challenges**: Ensuring consistent data quality.

### Economic Feasibility

* **Hardware and Installation**: The primary costs include purchasing cameras, servers, and other necessary hardware. Installation expenses cover setting up cameras at strategic locations such as toll booths, parking lots, and secure entry points.
* **Software Development and Licensing**: Developing or purchasing ANPR software involves significant costs. This includes licensing fees, customization, and integration with existing systems.
* **Data Storage and Management**: Storing and managing the large volumes of data generated by ANPR systems requires investment in robust data storage solutions and database management systems.

Chapter 2

# Literature Review

Automatic Number Plate Detection is a subject undergoing intense research for quite some time. In this method, the computer recognizes the string of characters from a vehicle's number plate. This system has a wide range of applications in the field of security, parking, charging for over speeding, identification of stolen cars, etc. The system is built under six modules as followed: Preprocessing, Threshold Optimization, Plate Localization, De-noising, Character Segmentation and Character Recognition [1].

Recent interests of ANPR systems include sophisticated machine learning techniques (like deep learning, neural net-works, SVMs) along with good plate localization and character segmentation algorithms. Localization of license plate refers to extracting the region in an image that contains the plate and some of the widely used techniques for localization include scale shape analysis, edge detection, mathematical morphology, connected component analysis, regional segmentation, and statistical classification [2] .

Nepali vehicles have license numbers encoded in the both rear and front side with two different sized rectangular plates. The front sized plates are usually in 4 :1 ratio and the back sized plates are in 4: 3 ratios. The rectangles with desired aspect ratio will only qualify for the further processing. The aspect ratio test won’t guarantee to output that one specific rectangle which has got the license plate but do make the job easier for further processes by eliminating the large numbers of the candidate rectangles. Profile tests are performed to accurately identify a license plate. Profile tests are carried out on candidate image regions that are filtered from aspect ratio test. For a profile test, we normally convert image regions into binary form and then calculate both row and column profiles. The row profile of the license plate rectangle will either have one or two peaks above the preset threshold corresponding to the 4: 1 or 4: 3 ratio plates respectively [3].

Beymer [4] proposed a model that deals with traffic and lighting conditions while using segmentation, classification, and tracking methodologies. The model focuses on vehicle segmentation and tracking and the computation of traffic parameters from tracking data. They have used a feature-based traffic tracking approach to track vehicles under congestion. The system tracks vehicle sub-features, which makes it less susceptible to partial occlusion. A network of C40 DSP (Digital Signal Processor) chips linked to a host PC has been used to construct a real-time version of the system. Wang proposed an approach to use an enhanced background-updating algorithm and feature-based tracking method. In their paper, they have used video-based traffic detection through an improved background-updating algorithm, then tracking the moving vehicles by a feature-based tracking method.

Zhu [5]proposed a vehicle detection and tracking volume statistics algorithm based on an improved single Gaussian model. The algorithm has three sections: moving target detection, shadows suppression, and traffic volume count. The single Gaussian model detects moving vehicles while shadow suppression is performed in the RGB feature space. Traffic volume was calculated in the virtual lanes. Deng proposed a model that can be used for object segmentation, classification, and tracking methodologies to know the real-time measurements in urban roads. The model used the background subtraction method to detect stationary or slow-motion objects. Experimental results showed a diminishing 20% degradation of infrastructure capacities.

Wen-Juan [6]proposed a model that can locate and track the vehicles in a video and calculate the traffic flow, queue length, queue waiting time, the average speed, and other vital parameters. The model used online learning mechanisms and optical flow to track objects. Experimental results showed the accuracy of daytime detection was 98%, and nighttime detection was 92%. The average speed detection of daytime was 95%, while nighttime was 90%. Kim compared different Artificial Neural Network (ANN) models for real-time vehicle type recognition. They compared deep learning-based object detection models R-CNN (Region Based Convolutional Neural Network), Fast R-CNN, Faster R-CNN, YOLO, and SSD (Single Shot Detector) in processing speed and accuracy for best performance. Faster R-CNN had less FPS (frames per second) with better accuracy, while SSD had less accuracy with better FPS. The YOLO model was a middle ground.

## YOLOv8 Architecture

YOLOv8, an advanced object detection model, builds on the strengths of previous YOLO versions with several architectural and performance improvements. The model starts with an input layer that preprocesses images through normalization and resizing. It features an enhanced backbone, typically an improved CSPDarknet, which uses convolutional layers, batch normalization, activation functions like Leaky ReLU or Mish, and residual connections to extract rich feature representations. The neck of the network aggregates multi-scale features using structures like the Feature Pyramid Network (FPN) and Path Aggregation Network (PAN), which enhance both top-down and bottom-up information flow. The head of YOLOv8 employs an anchor-free mechanism, predicting bounding boxes directly without predefined anchor boxes, and uses a grid-based approach for bounding box regression, class probability prediction, and objectless scoring. The output layer provides detected bounding boxes, class labels, and confidence scores, with Non-Maximum Suppression (NMS) applied to eliminate duplicates. Key innovations in YOLOv8 include its improved CSPDarknet backbone and the anchor-free prediction mechanism, which simplify the architecture and boost performance [7].

Chapter 3

# System Design

Python being a simple, versatile, scalable and a language with vast number of libraries and framework which makes it easy for the process of data analysis, machine learning, training and testing data, we will be programming our software mostly using python language utilizing its powerful libraries like Pytorch, NumPy, Ultralytics (YOLO), Pytesseract etc.

## Data Acquisition

Data had to be collected for training dataset and to collect data for creating a dataset for YOLOv8 to detect the Embossed Number Plate, we used our phones to capture video. We position our camera at various angles and height to capture different views of the numberplate. We ensured that each image contained at least one Embossed Number Plate which was visible and clear.

When using a camera to collect data, it is important to be aware of the artifacts that might be introduced to the images and data. For instance, the depth of field blur might cause certain parts of the image to be out of focus which might make it hard to see the characters of the license plate. The motion blur might result from the movement of the vehicle or camera and could also lead to image distortions. These artifacts can significantly affect the dataset’s quality and accuracy, making it challenging to detect the vehicles and license plates.

The images collected thus far need to be labeled properly in order to turn it into useful data to create a dataset for our training model. In order to label the images, we use an open-source free to use tool called “CVAT.ai”. It runs locally and supports multiple output formats like YOLO, XML, VGG, JSON, CSV.

## 3.2 Data Preprocessing

Regardless of all the measures we took while acquiring the data some unwanted artifacts are introduced in the image which hinders the accuracy of the model. To minimize its effects, the intensities of the image’s pixel were adjusted and resolution of every image was set to a standard scale i.e. (640\*640) which is an optimal size for achieving a good result.

## Number plate segmentation

Number plate segmentation is a computer vision task that involves isolating and extracting the region of an image containing the license plate. This process is essential for number plate detection for accurately identifying and isolating the license plate area from the rest of the image. We will be implementing segmentation using YOLO (You only look once) which is a popular object detection algorithm used in computer vision and deep learning.

The way in which YOLO will work is, it will divide the input image into a grid and make predictions for each grid cell. As we will contain bounding box in our images YOLO will predicts bounding boxes for objects present in the image. Each bounding box is associated with confidence scores and class probabilities.

## Data Augmentation

To expand the amount of data that is at disposal when training a model in machine learning, it’s common to apply a technique known as data augmentation. The basic idea behind this strategy involves deriving fresh sets of training samples by subjecting original datasets to diverse transformations. Such manipulations can include completely random rotations and flips or modifications involving crops with different brightness levels, contrast variations or even changes in saturation levels too. This method offers an opportunity for models’ exposure towards more complex dynamics present amongst input information thus enhancing their capacity for generalization during tests with unseen data inputs.

## OCR implementation

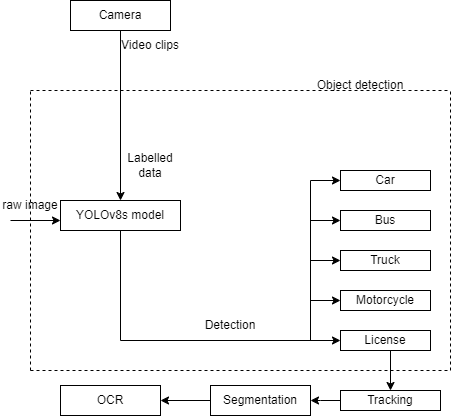
Optical Character Recognition is a technology that converts different types of documents, images captured by a digital camera into an editable and searchable data. After the image segmentation technique to locate and extract the region of interest (ROI) containing the license plate in an image or video frame we will apply preprocessing like resizing, normalization, and contrast adjustment to improve OCR performance. As in nepali number plate there will be lots of characters representing different meaning as shown in fig 3. We again need to use the segmentation process to isolate individual characters of the image of number plate. After that we will apply an OCR engine to recognize characters within the segmented regions. We will be using Tesseract OCR library for our project.

Figure 3‑1 Object Detection

Figure 3.1-1 shows different types of objects we are detecting and the complete work flow of our software. Firstly, car, bus, truck etc. in the image or the video frame will be detected from our model then the vehicle will get tracked and assigned with a unique id, then the tracked Number plate of that vehicle will be segmented and fed into the OCR.

## Hardware and Software requirements

### Hardware

* **Cameras**: Need high-resolution IP cameras with minimum 1080p resolution, 30fps as a good camera quality means the frames will be less noisy and with less motion blurring effect which will help in getting a good result. The camera should also be weather resistant as it could be installed in any place with harsh weather condition.
* **Processor**: CPU i.e. Intel Xeon or equivalent, minimum 8 cores or a GPU would be needed to analyze the images or the video frames.
* **Storage:** 1TB storage capacity SSD for fast read/write operations would be enough to store the video frames and results.

### Software

* **Operation System**: A light weight operation system of Linux might be helpful to run the server. For Camera OS a firmware provided by the camera manufacturer would be good.
* **License Plate Detection Algorithm**: Algorithm of YOLO was utilized for implementing our software.
* **Development Tools:** For programming our logic, we used python as a programming language as it is good for data related task.
* **Vehicle Tracking Software:** Deep Sort algorithm was used for multi-object tracking which was cost free (Open source).

Chapter 4

# Implementation and Discussion

Number plate detection is a system or a model intended to automatically identify a license plate on a vehicle and retrieve the number plate's data.

## Methodology

Our system will locate the number plate in a vehicle and by processing the video as shown in the following figure. YOLOv8s model trained in the COCO dataset detects motorcycle, car, bus, truck and other objects. The detected objects are tracked using DeepSORT and the object being tracked are then provided to our costume model implemented for especially detecting the Nepali Embossed Number Plate. For every license plate detected in a frame, we keep track of id, given by DeepSORT, and send it for recognition if it is continuously seen for consecutive frames. The license plate is fed to character segmentation module, and the segmented characters are sent for recognition. The output given by the character recognition model is then stored and displayed with the vehicle track id and the image of the vehicle.

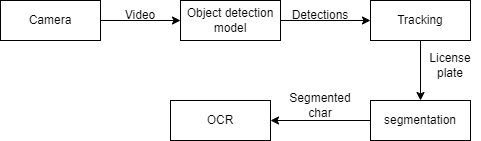


Figure 4‑1 Project Methodology

## Implementation Steps

For Implementation of the software, we followed following steps

### Model Training

We implemented a comprehensive data collection and processing strategy to develop our license plate detection model. Our data collection involved capturing images of vehicles using personal phones and sourcing additional images from the internet, resulting in a diverse dataset. These images were annotated with bounding boxes in the YOLO format, marking the location and dimensions of each vehicle's number plate.

The annotated dataset was divided into training and testing sets to accurately evaluate the model's performance on unseen data. We utilized the YOLO library by Ultralytics for training, configuring a YAML file to specify paths to the data directories. The model was trained for 100 epochs, which we determined to be optimal through experimentation.

Post-training, we evaluated the model's performance using data analysis graphs to identify strengths and areas for improvement. The best-performing model, demonstrating the highest accuracy and robustness, was saved as a .pt file named best.pt and stored in a directory named "trained".

### Tracking

For tracking the vehicle in the image or the video frame we utilize the pre-existing open-source object tracking library (SORT). The state of the objects being tracked is continuously calculated by the SORT algorithm using the KF algorithm. It uses Hungarian algorithm to accurately link the modeled objects that are being tracked, to new measurements acquired by an object detector.

### Processing ROI

Preprocessing the acquired image is the first stage in any image processing system. Pre-processing involves some actions on the image to improve the area of interest, and as a result, these procedures are wholly dependent on context. In our situation, the area of interest is the vehicle’s license plate, so we put the following processes into place to transform the image into one that can be processed further and improve the effectiveness of the segmentation algorithm. Preprocessing of ROI includes, region grow, HSV Color Space Conversion, color masking etc.

Gaussian Blur

Histogram\_filter

Original\_img



After\_clahe

Adaptive Threshold

Figure 4‑2 Pre-processing Detected License Plate ROI

### Character Segmentation

This is the crucial steps for ANPR system, this process involves segmenting the characters from the number plate individually. For this process we use contour function on our processed image and we construct a rectangular box using the draw-rectangle function for each contour, then the rectangular box was cropped and was sent for further processing.



Applying contour

Threshold image

Figure 4‑3 Different phases of detecting character in LP

### Character Pre-processing

This process involves adjusting the threshold value and other components of the segmented character to make it easier for the OCR engine to recognize the character accurately.

Pixels holding the shape/structure of a character in the boundary won’t contribute much, so we provide the segmented character with a padding layer which increase the accuracy in detecting the segmented character.



After

Before

Figure 4‑4 Image of character before and after applying padding

### Data storage

The data includes, Frame no, vehicle’s id, Numberplate Characters and accuracy for our software. These are automatically stored in CSV format and the stored file is further filtered and the data with highest accuracy will be stored in another file.

The above implementation process is described using the block diagram below:

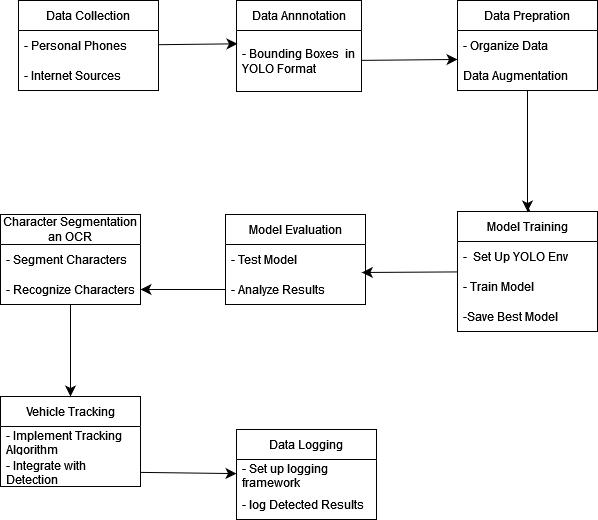


Figure 4‑5 Block diagram of Implementation ANPR software.

## Output Obtained

* Our system can detect the vehicle for a video frame.
* Our system can locate the number plate of the vehicle.
* The number plate is assigned with a unique track id.
* The information on the number plate will be stored in any required format,

## Time Schedule

Gantt-Chart table shows timeline of the project where each box in the rows shows 2 weeks of time. The project development phase lasted a total of 28 weeks starting from 2nd January 2024 to 12th July 2024 and the shaded boxes show the timeframe when the project was developed. The phase of the project development is presented in the first

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Weeks | **1-2** | **3-4** | **5-6** | **7-8** | **9-10** | **11-12** | **13-14** | **15-16** | **17-18** | **19-20** | **21-22** | **23-24** | **25-26** | **27-28** |
| Pre-analysis phase |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Project analysis |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Detailed study and analysis |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Learning ML |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Data collection |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementation of system |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Training and testing |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Documentation of project report |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

column and the respective row shows the timeframe when it was done.

Table 1 Gant Chart

[8]

**Chapter 5**

# Analysis and Evaluation

## Data Analysis

* Data Sources: We collected our data using phones from different location mainly including the areas with high traffic. We basically captured the frames containing at least one vehicle with embossed number plate from a video.
* Data Volume: There were total of 600 images which were further utilized through augmentation technique and we were able to extend our dataset to 1200 images.

### Graphs

Below is the graph obtained after the training model our model.

Figure 5‑1 Graphs

The graphs in figure 5-1 were automatically generated after the termination of the training process. In the above graph we can see different parameters showing the accuracy of our model. In above graph the x-axis represents the number of epochs and y-axis represents a value.

Figure 5‑2 Correlogram of Labels in Our Model

In above figure 5-2, it shows the correlogram which refers to a graphical representation or visualization that shows the correlation between different classes or labels assigned to objects detected by the YOLO model. It help in understanding which classes tend to appear together in the same scene or image.

### Confusion Matrix

Overall, the confusion matrix is a powerful tool for evaluating the performance of a classification model. By breaking down the model’s predictions into a detailed table, we can gain valuable insights into its strengths and weaknesses, and identify areas for improvement. The confusion matrix for our trained model and precision, recall graphs are as follows:

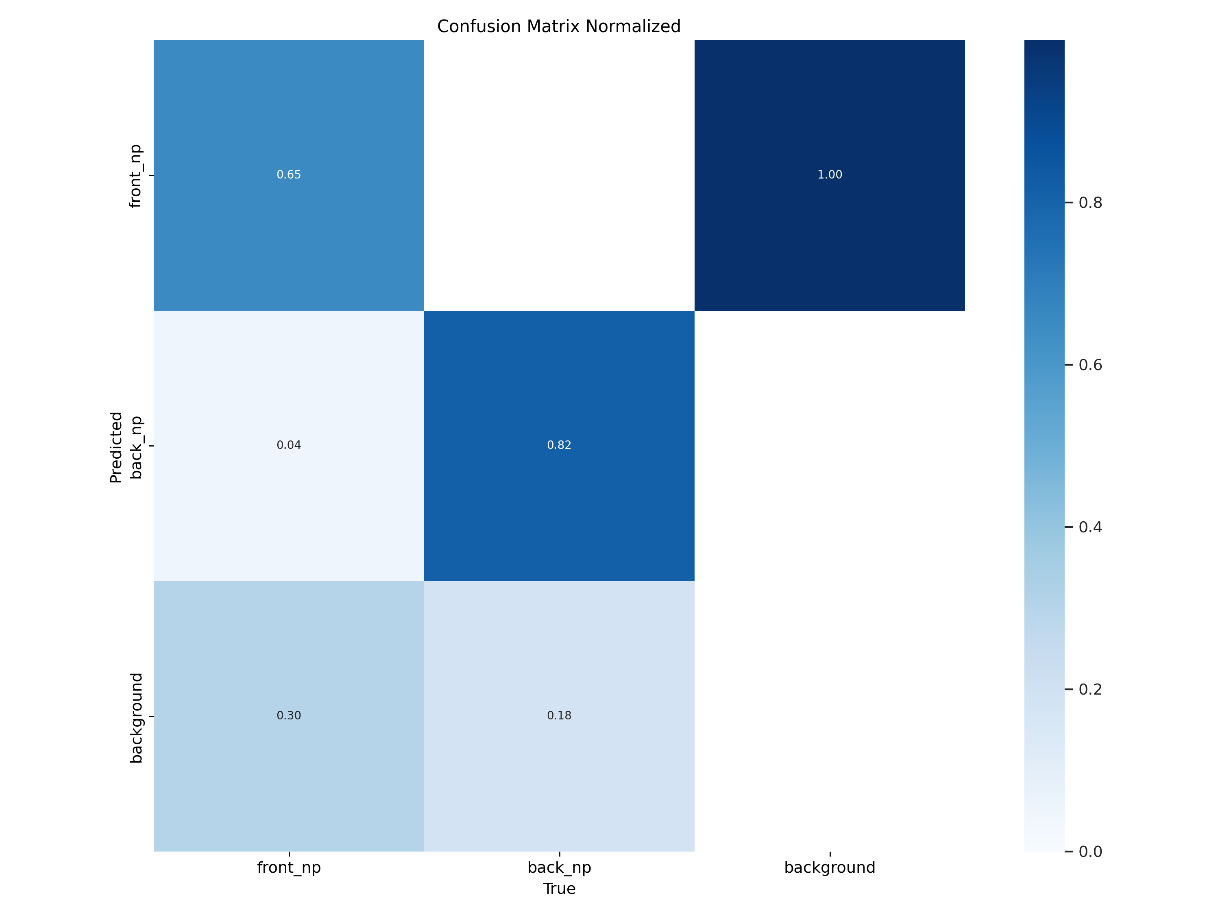


Figure 5‑3 Confusion Matrix

Our model is trained for two classes i.e. front number plate and back number plate of vehicles. Above normalized matrix shows the confidence score of each class of our model. It was generated by YOLO after our training was finished on 100 epochs.

## Comparison with the Objectives

The output of our software aligned well with the initially stated objective. We were able to build a software which detects the number plate of the vehicle and automatically reads and stores the content of the number plate in suitable format.

* Vehicle were successfully detected.
* Unique id was assigned to each number plate.
* Was able to detect vehicle for the video frame.
* Accuracy on detecting the number plate was above 85%.

## Discussion and findings

1. Accuracy of ANPR software: After training our model for above 100 epochs the mAP score and box loss were not seen improving, so we terminate our training operation with about 87% accuracy.
2. Integrating with Other Systems: As our software analyze the frames of video and a low-end camera can cause motion blurring and other issue while capturing a video so integrating our system in a high-end camera can have better result.
3. Data Management: The data captured from each frame is stored in a any format as per the user choice. For now, the data is stored in CSV format.
4. Character Segmentation: Segmenting the character: It was hard to segment the particular character from the number plate as the numberplate still contained some noise so for that we calculate the average range of aspect ratio and character area percentage by observing some output and used it to select particular portion of the number plate.
5. Threshold value: It was hard to determine a perfect threshold value for the input images as we had different types of images in different lighting condition so for that we adjust the parameter of adaptive thresholding i.e. (box size, C) by observing different outputs and implemented the best outcome.

**Chapter 6**

# Conclusion and future work

The number plate detection system developed in this project has successfully demonstrated the ability to detect vehicle number plates and store the recognized numbers accurately. By leveraging the YOLOv8 model for robust object detection and integrating it with an Optical Character Recognition (OCR) program, the system efficiently extracts and reads characters from number plates in real-time. This project not only showcases the feasibility of using state-of-the-art technologies for number plate detection but also sets a foundation for future enhancements and broader applications in traffic surveillance and road safety. The system's ability to store detected number plate information allows for efficient data management, facilitating tasks such as vehicle tracking and automated toll collection.

### 6.1 Future Work

To further enhance the system, several improvements and additional features can be considered:

* **Model Optimization:** Fine-tuning the YOLOv8 model for higher accuracy and faster inference times.
* **Scalability:** Expanding the system to handle larger volumes of data and support more extensive surveillance networks.
* **Advanced Features:** Integrating functionalities such as real-time alerts, automatic ticketing systems, and multi-language support for number plates.
* **Robustness:** Enhancing the system's performance under various environmental conditions and improving its resilience to partial occlusions and low-contrast scenarios.

# Bibliography

|  |  |
| --- | --- |
| [1] | S. Subedi, "Sanjaya's Blog," [Online]. Available: https://sanjayasubedi.com.np/deeplearning/nepali-license-plate-recognition-with-deep-learning/. [Accessed 20 01 2024]. |
| [2] | a. k. pant, p. k. gyawali and s. acharaya, "research gate," [Online]. Available: https://www.researchgate.net/figure/Automatic-Nepali-Number-Plate-Recognition-System\_fig3\_323999303. [Accessed 21 01 2024]. |
| [3] | A. K. P. r. K. G. Shailesh Acharya, "Automatic Nepali Number Plate Recognition with Support Vector Machines," International Conference on Software, Knowledge, Information Management and Applications (SKIMA), 2015. |
| [4] | P. M. B. C. a. J. M. D. Beymer, "A real-time computer vision system for measurint traffic parameters," pp. 495-501. |
| [5] | W. X., M. L-M. , Z. B. and D. K-L., "A video-based traffic violation detection system," *Proccedings 2013 Ubterbatuibak Conference on Mechatronic Sciences, Elctric Engineering and Computer(MEC),* pp. 1191-1194, 2013. |
| [6] | W.-j. X. and J.-f. L., "Application of vision sensing technology in uurban intelligent traffic control system," *4th International Conference on Computer and Technology Applications (ICCE-Asia),* pp. 1-4, 2020. |
| [7] | I. C., "Yolov8 architecture explained".*https://iq.opengenus.org/yolov5/.* |
| [8] | A. S. A. Y. A. R. V. J. Anchal Baliyan, "AUTOMATIC LICENCE PLATE DETECTION AND RECOGNITION," International Research Journal of Engineering and Technology (IRJET), 2020. |