Minor Project Final Report on

"Vehicle Number Plate Recognition "

in partial fulfillment for the award of the degree of

Bachelor of Computer Engineering

Submitted By

Abnish Karmacharya [20070274]

Bibek Manandhar [20070282]

Rikesh Ghimire [20070305]

Sulav Shrestha [20070315]

Submitted to

Department of Computer and IT Engineering

Everest Engineering College

Lalitpur-2, Sanepa

October, 2023

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who have contributed to the successful completion of this project on "Vehicle Number Plate Recognition using OCR". Their support and assistance were invaluable throughout the entire process.

First and foremost, We extend our deepest appreciation to minor project management commeette for their guidance, expertise, and unwavering support. Their insightful feedback and valuable suggestions played important role in shaping this project.

We would like to acknowledge our mentor Mr. Saroj Thapa for introducing OCR and Machine Learning and providing some previous papers regarding the topic.

Further, more we would like to thank our whole faculty members and supervisor who has helped us a lot for this project. Their involvement has make our work easier and complete on time.

Finally, we want to express our gratitude to all those who participated in the data collection process and provided access to the necessary resources for this project.

This project would not have been possible without the collective effort and support of all those mentioned above. Thank you for everyone's effort and hardwork.

ABSTRACT

Vehicle Number Plate Recognition is a critical component in modern traffic management and surveillance systems. This technology enables automated extraction of alphanumeric characters from vehicle number plates, allowing for efficient identification and monitoring of vehicles in various applications such as toll collection, parking management, law enforcement, and traffic analysis.

The proposed system comprises three main stages: pre-processing, object detection and recognition. In the pre-processing phase, input images are subjected to techniques such as noise reduction, Grey-scale conversion and binarizarion to improve the quality of the image.

The heart of the system lies in the object detection which employs a deep learning-based model trained on a diverse dataset of number plate images. The model is designed to recognize alphanumeric characters accurately, even under challenging conditions like low light, varying angles, and occlusions. Furthermore, the system incorporates post-processing techniques to refine the recognized characters and improve overall accuracy.

To evaluate the system's performance, extensive experiments were conducted on a benchmark dataset, comprising a wide range of license plate images from different regions and environments. The results demonstrate high accuracy rates, with an average recognition accuracy of over 95%. The system also exhibits robustness against various challenges commonly encountered in real-world scenarios.

Additionally, the proposed Vehicle Number Plate Recognition system is implemented on a distributed platform, allowing for real-time processing of high-resolution images from multiple cameras simultaneously. This ensures the system's scalability and applicability in large-scale traffic management infrastructures.

In conclusion, the Vehicle Number Plate Recognition system presented in this paper showcases a robust and accurate solution for automated number plate identification. Its effectiveness in diverse real-world scenarios positions it as a valuable tool for traffic management authorities, law enforcement agencies, and other stakeholders involved in vehicle monitoring and surveillance applications. The system's scalability and adaptability make it a promising candidate for integration into smart city initiatives aimed at enhancing traffic efficiency and safety.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1: INTRODUCTION

1.1 Background

Vehicle number plate recognition is a cutting-edge technology that has revolutionized the way we monitor and manage traffic and security on roads and in parking facilities. It involves the use of advanced computing and machine learning algorithms to automatically capture, read, and select number plate area of vehicles in real-time.

By utilizing advanced image processing methods, this system reduces dependency on manual intervention, leading to improved operational effectiveness and significant time savings.

This technology has been used in various sectors including law enforcement, transportation management, and commercial applications, due to its ability to improve safety, security, and efficiency.

In this era of rapid urbanization and increased vehicle ownership, vehicle number plate recognition plays an important role in enhancing the overall quality of life by providing smoother traffic operations, reducing crowd , and improving security

1.2 Statement of Problem

In modern traffic management and security systems, the accurate and efficient recognition of vehicle number plates plays a critical role. The existing manual methods of recording license plate numbers are time-consuming, error-prone, and does not fullfill demands of today's fast-paced environments.

The system must accurately recognize number plates under various environmental conditions.

Real time operation of the system may not provide the accuracy that the end user is searching for.

1.3 Objective

The objectives of Vehicle Number Plate Recognition system are used to enhance automation, improve efficiency, and provide accurate identification of vehicles.

The key objectives of implementing Vehicle Number Plate Recognition system is Automate Data Collection, Increase Accuracy, Improve Security and Safety, Enhance Traffic Management, etc.

1.4 Scope

The scope of Vehicle Number Plate Recognition system is extensive and continues to grow as advancements in technology improve its capabilities. The Vehicle Number Plate Recognition system can used for traffic management, parking management, enhance security system and data analysis and managemant.

1.5 Applications

Vehicle Number Plate Recognition has a wide range of applications across various industries and sectors. Basically this system can be used for the identification of stolen vehicle, crime investigations, traffic management such as automtic toll collection and traffic violation detection, parking management system such as automatic parking system and time limited parking system, rental and car sharing services, data analysis and reporting, smart cities and urban planning and this system helps in many sectors.

1.6 Hardware and Software Requirement

Hardware Required

1. Camera (any camera that can click picture)
2. Computer with decent RAM and GPU
3. Display Monitor

Software Required

1. Windows OS more than XP, 7 and 8
2. Python installed
3. Image Viewer
4. Web Browser

Chapter 2: LITERATURE REVIEW

2.1 Related Works Description 1

This paper represents that-In first step, camera captures the video of vehicles number plate. To read this video MATLAB software is used. The video used for operations has timing of 10 to 15 seconds. The 10 second video contains 240 frames/images. In second step, video gets converted into frames at frame rate 24 fps. In third step, frames are converted into Images which is very important step. Then Opening and closing operations are done. To extract vehicle number plate, Image processing like segmentation, recognition, localisation has been done. First canny edge detection algorithm detects the edges of image. Then morphological operator are used. And in this way number plate gets detected[1].

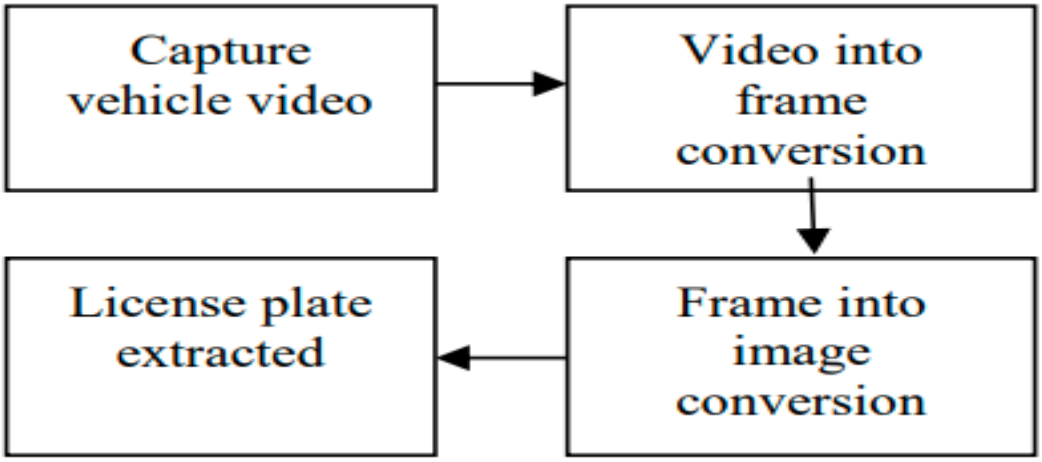


Fig 2.1: Block diagram of proposed approach

2.2 Related Works Description 2

The comprehensive survey provides a detailed overview of various techniques and methodologies used in VNPR systems, including image preprocessing, plate localization, character segmentation, and recognition methods.[2]

2.3 Related Works Description 3

In this report it explains the problem of having different colour number plate and different formate of number plate like the plate that is seen commonly in Nepal and India. Such as first two letters are state code then district code followed by four unique code for that particular vehicle number plate. . If the color of number plate

is identical to background then there can be errors while detecting and locating the number plate. If this error occurs the it can be resolve as, first input colored image is taken from camera. Next the colored image is converted into grey scale conversion. To remove noise from image various filtering methods are used, in this median filtering is used which removes salt and pepper noise. The contrast of image can be cleared using histogram equalization. After that localization of plate is done using Sobel Edge detector where regions and edges of image is done. Then the numbers of plate is segmented using segmentation method[3].

2.4 Related Works Description 4

In this paper a system captures image of vehicle and identity is verified using Raspberry pi processor that there is no change in the original image and test image. If any unregistered vehicle is detected then system gives alerts to the computer using buzzer alarm sytem. Input to the system is taken as image of vehicle and the output is detected number plate .This paper consists of following processes: Colored image to gray scale conversion, vertical edge detection using Sobel detector , converts non-linear image to linear image, opening-closing and dilation, vertical projection and thresholding, location of number plate, filtering the number plate and image enhancement, linearization and character segmentation for obtaining separate numbers. Linux operating system is used. Linux is an open source operating system. Here one can change codes and add programs[4].

2.5 Related Works Description 5

Number plate recognition system is used for monitoring and managing the entrance gates of colleges in private and public organizations to identify vehicle license plate numbers at the parking gate. This system also works on stolen vehicles on roads. In this system camera is placed near entrance and exit gates. The images taken from camera are processed in computer and the vehicles information is stored in the system database for longer period. Also this system can control the automatic opening and closing of parking gate giving permission to only authorized vehicles. Likewise, more information regarding vehicles can be extracted from the system. This system uses waterproof and dustproof camera which provides high quality images for processing. .This cameras are used under any weather conditions with powerful infrared radiations for detecting the vehicles under darkness. It can also be able to detect the vehicles under any temperature range so that a clear image is obtained. This system uses a software which can be available to users with low cost. The software proposed in the system is used to define both white list for authorized vehicles and black list for unauthorized vehicles which will not be allowed to enter the gate.This system by implementing proposed system can provide higher security and safety for the vehicles. The system can be connected to gates for automatic opening and closing of gates. If further modifications like the system is connected with alarms lights or smart boards, the system can display any message like hello, welcome,etc[5].

2.6 Related Works Description 6

In 2022 International conference on advanced creative networks and intelligent System (ICACNIS), in their research they develop an autoencoder for denoising text images and analyze the OCR performance in converting the denoised image into text. [6] The contribution done by this researcher are:

- An autoencoder architecture for denoising is used to improve OCR performance in text conversion.
- A test on the effect of the number of datasets on the autoencoder's performance in performing image denoising.
- A test on the effect of the number of datasets in the autoencoder model on the OCR performance of text conversion.

They have four stages for this research, first they study the OCR character with noisy images. Then secondly, they developed an autoencoder model to denoise the noisy image. Then, they evaluate the performance of the autoencoder model. Lastly, they evaluate the OCR performance on the denoised image created by autoencoder. [6].

In 2006, Tesseract was considered one of the most accurate open-source OCR engines then available. The Tesseract engine was originally developed as proprietary software at Hewlett Packard labs in Bristol, England and Greeley, Colorado between 1985 and 1994, with some more changes made in 1996 to port to Windows, and some migration from C to C++ in 1998. A lot of the code was written in C, and then some more was written in C++. Since then all the code has been converted to at least compile with a C++ compiler. Very little work was done in the following decade. It was then released as open source in 2005 by Hewlett Packard and the University of Nevada, Las Vegas . Tesseract development has been sponsored by Google since 2006. Tesseract was in the top three OCR engines in terms of character accuracy in 1995. It is available for Linux, Windows and Mac OS X. However, due to limited resources it is only rigorously tested by developers under Windows and Ubuntu.[7]

Literature Gap

So far, we read a lot of articles and research paper on OCR and ANPR, We have found that there is still lack of research in case of accuracy. Our main goal with this project is to able to detect character from the image and predict the character with high accuracy using CNN, Deep Learning AI and data set available to us. We found that there may not be dataset and we might have to create a new data set to detect in embossed and the normal number plates.

Chapter 3: METHODOLOGY

3.1 Introduction

The Vehicle Number Plate Recognition system is an important component of modern traffic management and surveillance systems. It plays an important role in automating tasks like toll collection, parking management, and law enforcement.

It is seen that anyone can enter into a private premises which lacks security and may cause havoc and destroy private property. Even when security is installed and authorities might face problems such as identifying the vehicle, parking space takeover.

Authorities find it very hectic on a busy day to log the vehicle numbers manually in a parking lot. So, in order to make the entire process autonomous, we can install this system so as to automatically detect the vehicle which is to be entered into college premises and take a picture of it and store the number in the database and check if it is allowed to enter.

Automatic Number Plate Recognition systems are important systems in transportation management and surveillance. They are capable of identifying vehicles by extracting the number plate and reading the plate identity which is unique identification code given to each vehicle. Such systems can be used for automatic traffic control, vehicle tracking and monitoring, security and many more.

Developing number plate recognition system requires integration of computer visionalgorithms with imaging hardware. Computer vision algorithms include image processing techniques for number plate localization, plate orientating and sizing, normalization andcharacter segmentation. Beside these, it includes pattern recognition techniques for optical character recognition. For better identification accuracy, machine learning techniques areused to learn from input data. There are many difficulties that such system may face; suchas, poor resolution, poor illumination conditions, blurry inputs, plate occlusion, different font size and variety of plate structures.

This technology reduces the unnecessary hectic manual work required on any busy day, saves the labor cost and is far more efficient than humans. This project is so versatile that it can be used as an entire application once converted to a software or can be used as a part of any big project.

3.2 Block Diagram

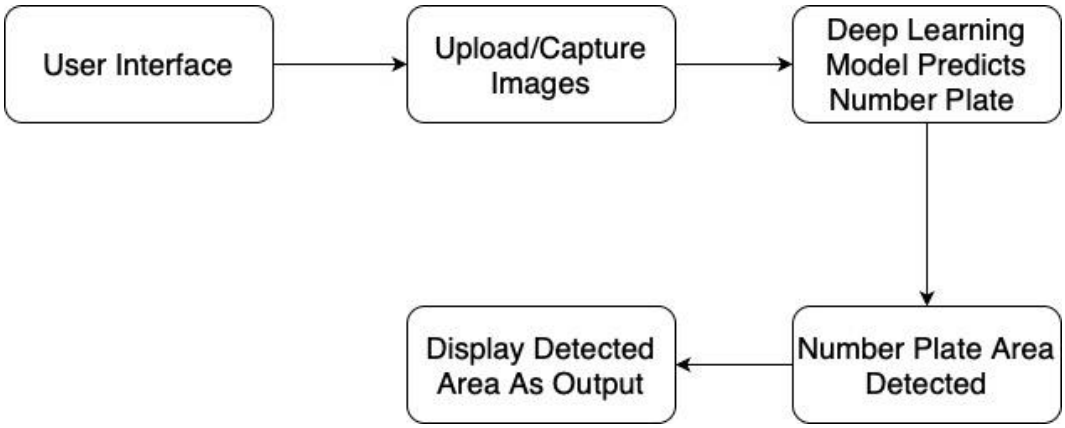


Fig 3.2.1: System Design

3.2.1 Description of System Design

In this system user first interact with system using user interface where he/she get option either to upload image image. The image further goes for pre-processing and deep learning model predict a area for number plate, after the prediction of number then this system marks number plate area. Finally, the user gets detected number plate boundary box as output.

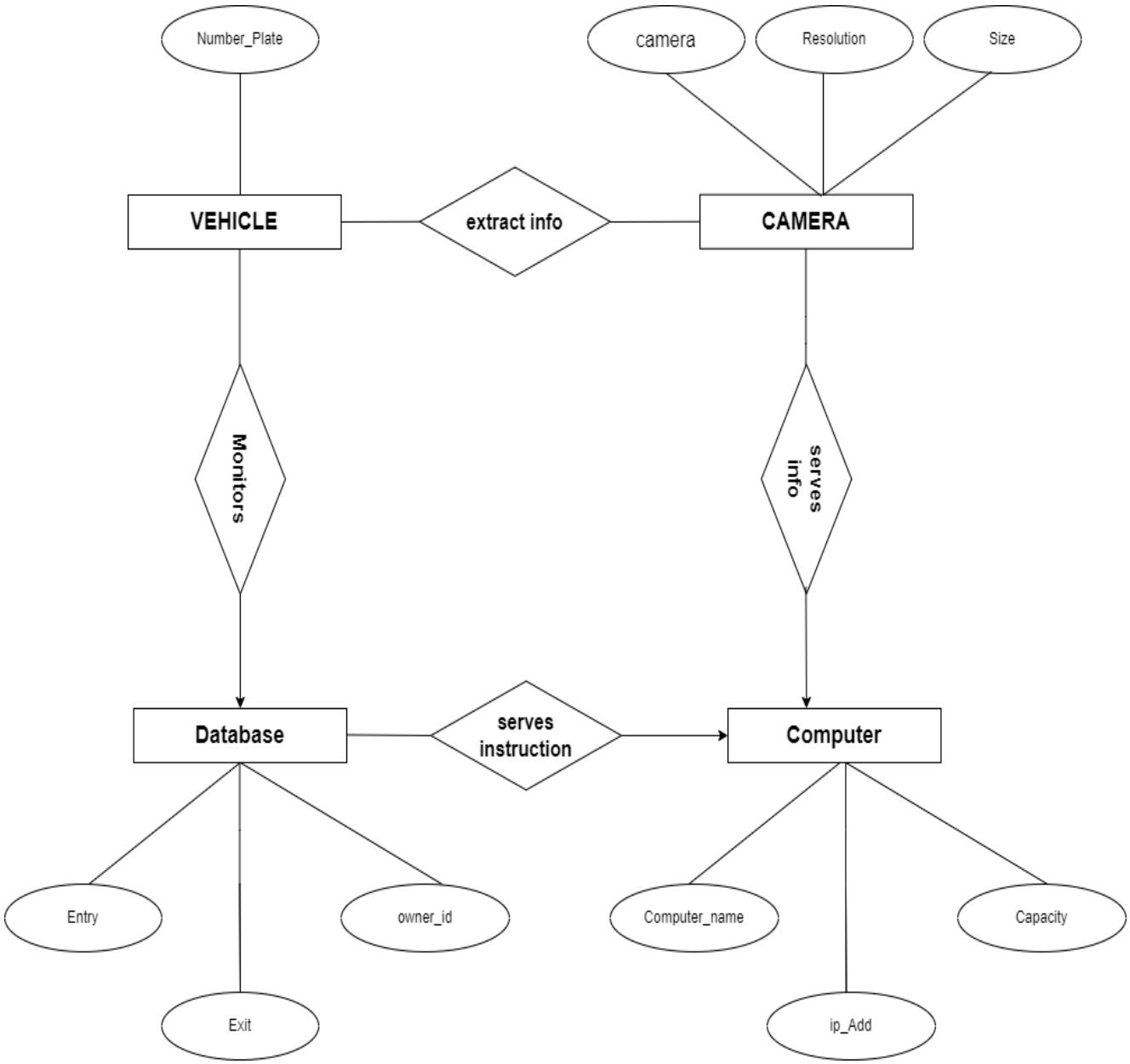


Fig 3.2.2: ER Diagram

3.2.2 Description of ER Diagram

An entity–relationship model describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between entities. In this system there are four components with their respective attributes and are connected with each other with some realtion.

3.3 Methodology

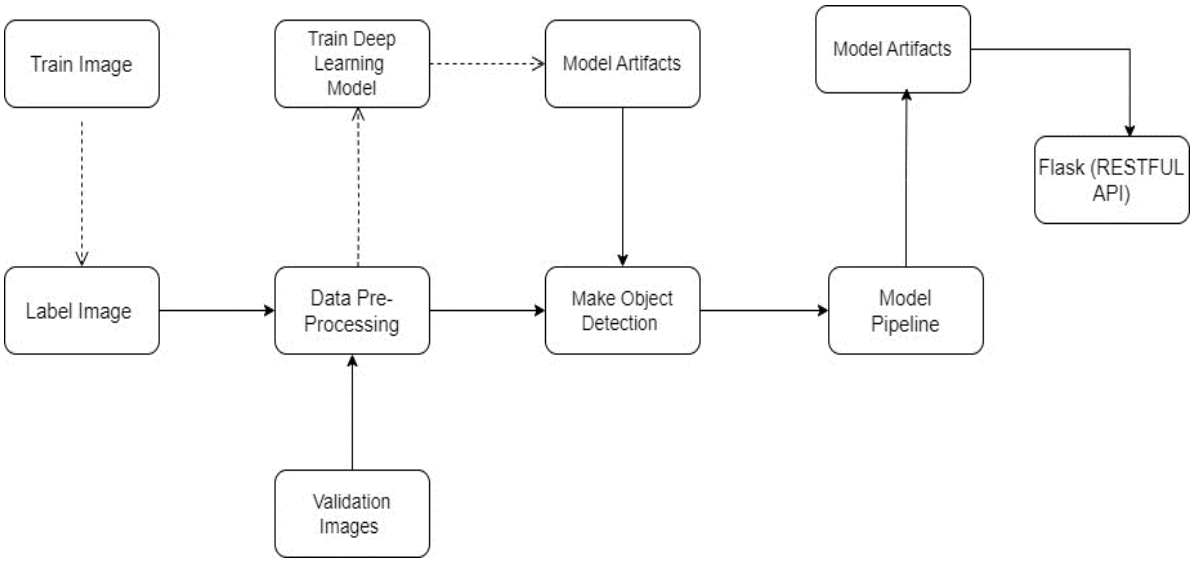


Fig 3.3: Methodology

3.3.1 Description of Methodology

3.3.1.1 Labelling Images

At first the uploaded image is trained which contains objects that the model needs to detect. The label image indicates the bounding box for the objects in the corresponding trained images and the data is obtained in the form of XML files which is later preprocessed.

3.3.1.1 Data preprocessing

For Data preprocessing we use libraries like train_test_split, load_img and img_to_array from Sklearn and tensorflow.keras.preprocessing.

Using the target values from the XML i.e. xmin, xmax, ymin, ymax. Saving these data in the form of array, we can get height, width and depth for the image. Where depth is R,G,B which is always 3 for images.

from arr.shape we can get h,w,d of image. Now load_img requires path and target size which is important to us. Target size is used for object detection model and we have different models like Inception, Mobile and all the different models are there and all the pretend models which was developed for the fixed sizes. Here we usesize of (224,224) which is stable and most widely used size. Then we convert it to array using img_to_array such that we can convert to our target size.

3.3.1.2 Normalization to labels

We have the lables i.e xmin, xmax, ymin, ymax = labels[0], now to normalize our data and model X should be normalized with width and Y should be normalized with height so

nxmin, nxmax = xmin/width , xmax/ width

nymin, nymax = xmin/height , xmax/height

Then we get normalized labels as label_norm = (nxmin,nxmax, nymin, nymax) which is for output and for normalized input we divide the load_image_arr array by 255.0. finally then the data can be used to train and test split for data to be trained to our Deep Learning CNN model.

3.3.2 Convolutional Neural Network for Number Plate Recognition

3.3.2.1. Definition

The license plate character recognition is the key to the whole process. It is the only authentication for vehicle identity. In the actual environment, under the influence of weather, angle, light and background, the collected characters may be deformed or polluted which poses great challenges to the recognition work.

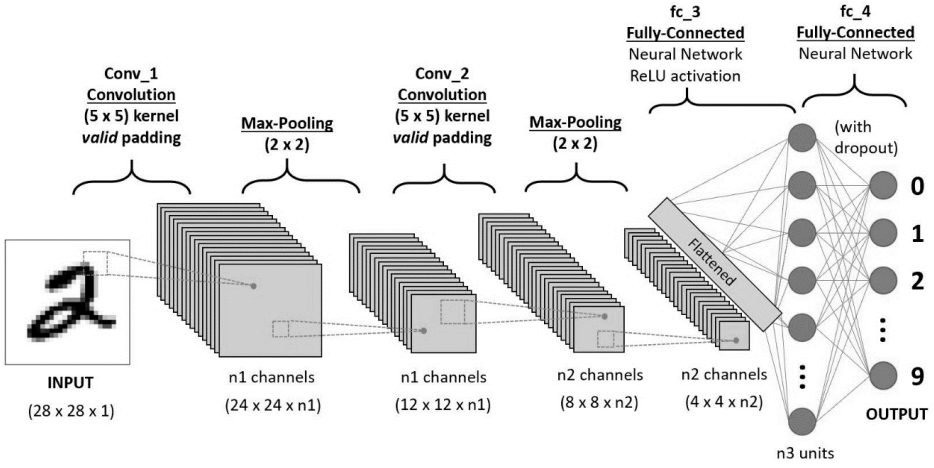


Fig 3.3.2 Convolutional Neural Network

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases to various aspects/objects in the image and be able to differentiate one from the other. The preprocessing required in a ConvNet is much lower as compared to other

classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics. The name “convolutional neural network” indicates that the network employs a mathematical operation called convolution. Convolutional networks are a specialized type of neural networks that use convolution in place of general matrix multiplicat ion in at least one of their layers.

In CNN, we obtain image features mainly through the convolution operation, that is, assigning a weight to each pixel in the image. Obviously, it’s a kind of linear operation. In reality, however, our samples may not be necessarily linearly separable. Therefore, to enhance the expressive ability of the network, we have to add nonlinear factors to the linear model by introducing the activation function.

The sigmoid function is the most commonly-used activation function, which simulates thebilateral inhibitory property of neurons. Nevertheless, we have to derive residuals in the back propagation. Since the maximum of the sigmoid function derivative is less than 1, the gradient dispersion may occur with the increase of network layers which eventually makes the network difficult to train.

3.3.2.2 Architecture of CNN

CNN architecture is inspired by the organization and functionality of the visual cortex and designed to mimic the connectivity pattern of neurons within the human brain. The neurons within a CNN are split into a three-dimensional structure, with each set of neuronsanalyzing a small region or feature of the image. In other words, each group of neurons specializes in identifying one part of the image. CNNs use the predictions from the layers to produce a final output that presents a vector of probability scores to represent the likelihood that a specific feature belongs to a certain class. There are two main parts to a CNN architecture

• A convolution tool that separates and identifies the various features of the image for analysis in a process called as Feature Extraction

• A fully connected layer that utilizes the output from the convolution process and predicts the class of the image based on the features extracted in previous stages.

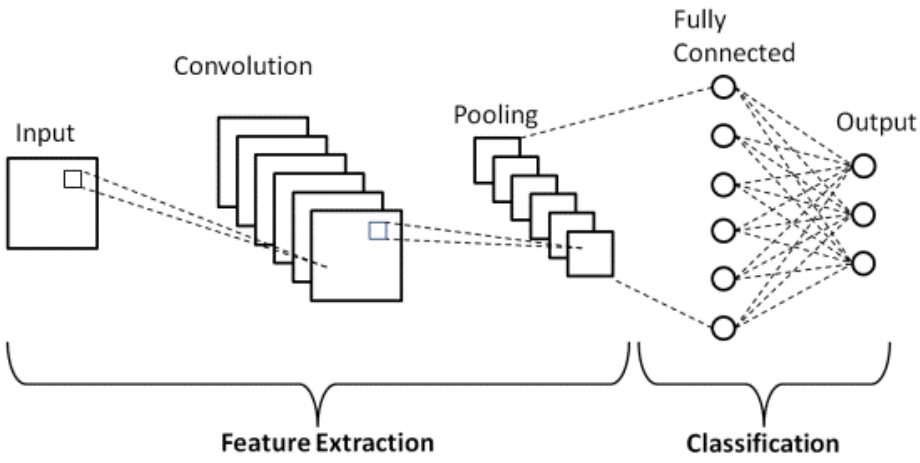


Figure 3.3.2.2 Architecture of Convolutional Neural Network

Convolution Layers

There are three types of layers that make up the CNN which are the convolutional layers, pooling layers, and fully-connected (FC) layers. When these layers are stacked, a CNN architecture will be formed. In addition to these three layers, there are two more important parameters which are the dropout layer and the activation function.

i. Convolutional Layer

This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size MxM. By sliding the filter over the input image, the dot product is taken between the filter and the parts of the input image with respect to the size of the filter (MxM).

The output is termed as the Feature map which gives us information about the image such as the corners and edges. Later, this feature map is fed to other layers to learn several other features of the input image.

ii. Pooling Layer

In most cases, a Convolutional Layer is followed by a Pooling Layer. Pooling layers reduce the dimensions of the data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. The primary aim of this layer is to decrease the size of the convolved feature map to reduce the computational costs. This is performed by decreasing the connections between layers and independently operates on each feature map. Depending upon method used, there are several types of Pooling operations.

In Max Pooling, the largest element is taken from feature map. Average Pooling calculates the average of the elements in a predefined sized Image section. The total sum of the elements in the predefined section is computed in Sum Pooling. The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer.

iii. Fully Connected Layer

The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. These layers are usually placed before the output layer and form the last few layers of a CNN Architecture.

In this, the input image from the previous layers are flattened and fed to the FC layer. The flattened vector then undergoes few more FC layers where the mathematical functions operations usually take place. In this stage, the classification process begins to take place.

iv. Dropout

Usually, when all the features are connected to the FC layer, it can cause overfitting in the training dataset. Overfitting occurs when a particular model works so well on the training data causing a negative impact in the model's performance when used on a new data.

To overcome this problem, a dropout layer is utilised wherein a few neurons are

dropped from the neural network during training process resulting in reduced size of the model. On passing a dropout of 0.3, 30% of the nodes are dropped out randomly from the neural network.

v. Activation Functions

Finally, one of the most important parameters of the CNN model is the activation function. They are used to learn and approximate any kind of continuous and complex relationship between variables of the network. In simple words, it decides which information of the model should fire in the forward direction and which ones should not at the end of the network.

It adds non-linearity to the network. There are several commonly used activation functions such as the ReLU, Softmax, tanH and the Sigmoid functions. Each of these functions have a specific usage. For a binary classification CNN model, sigmoid and softmax functions are preferred an for a multi-class classification, generally softmax is used.

3.3.3. Tools Used

Python

Python is an object-oriented, high-level programming language with dynamic semantics. It was created by Guido van Rossum and first released in 1991. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and

the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed.

Tensorflow

TensorFlow is an open-source library for numerical computation and large-scale machine learning. TensorFlow bundles together a slew of machine learning and deep learning (aka neural networking) models and algorithms and makes them useful by way of a common metaphor. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. Tensorflow is a symbolic math library based on dataflow and differentiable programming.

Keras

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research. It is a highly-productive interface for solving machine learning problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing and shipping machine learning solutions with high iteration velocity. The library follows best practices for reducing cognitive load: it offers consistent and simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear and actionable error messages. It also has an extensive documentation and developer guide.

Numpy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. It is a fundamental package for scientific computing in Python.

Sklearn

Scikit-learn features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific

libraries NumPy and SciPy. It facilitates best practices for testing and documenting estimators.

3.4 Flow Diagram

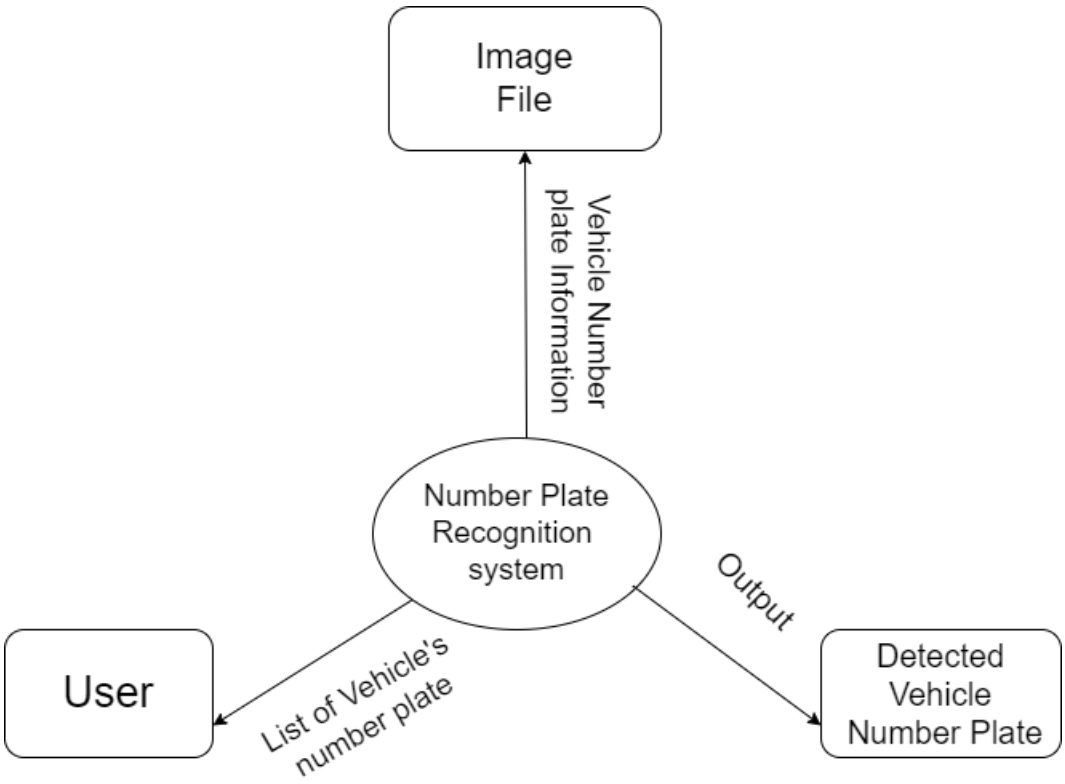


Fig 3.4.1: DFD Level 0

3.4.1 Description of DFD Level 0

DFD level 0 is the basic step where the user give input to system as image file and the image file is processed and vehicle number plate area is detected and displayed.

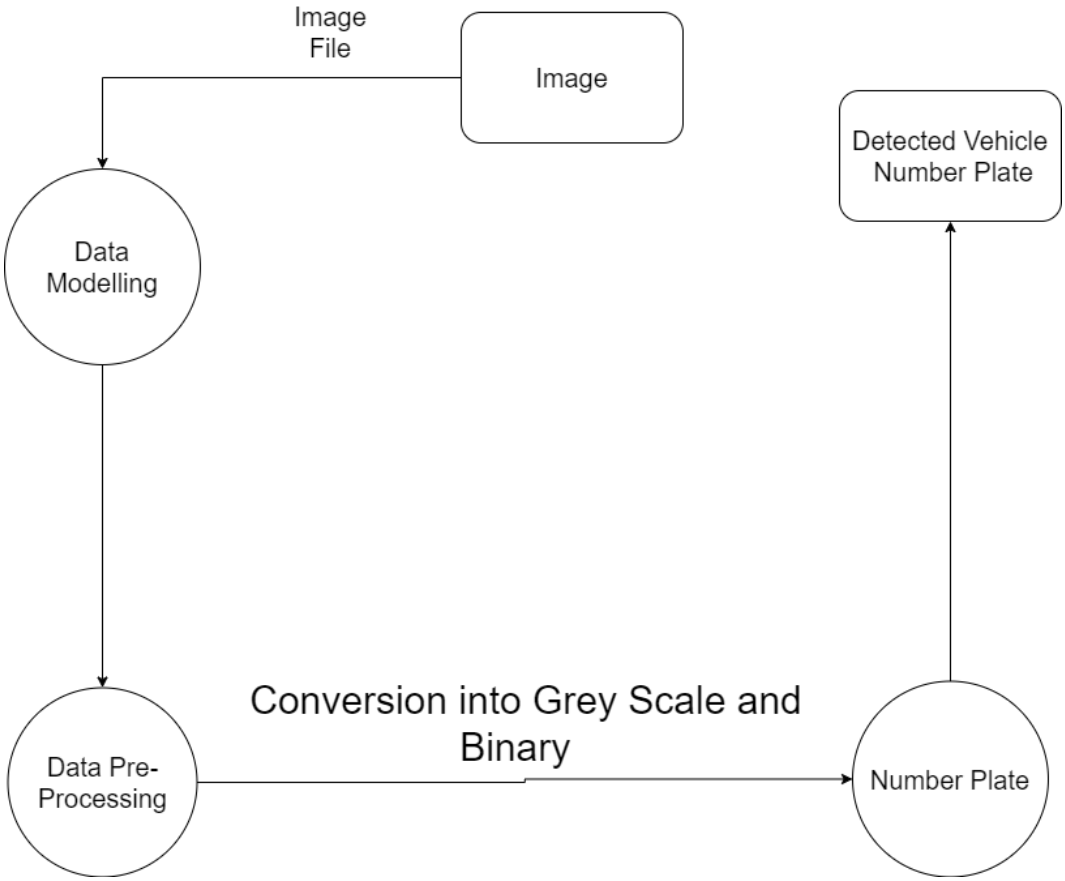


Fig 3.4.2 DFD level 1

3.4.2 Description of DFD Level 1

It is expanded form of DFD level 0, in this step the image is send as input and then it is modelled after the modulation procress the image is pre-procressed which convert grey- scale into binary and number plate area is selected and display it.

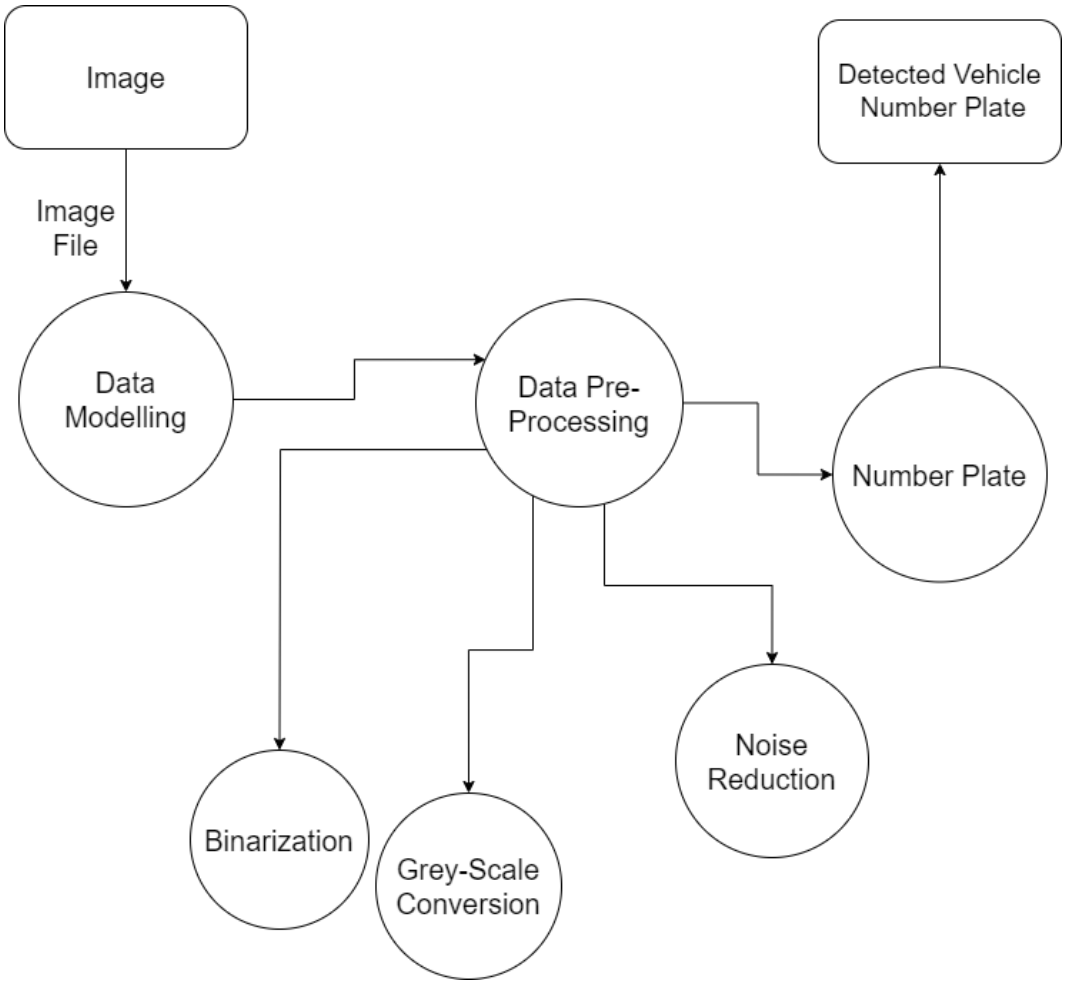


Fig 3.4.3: DFD Level 2

3.4.3 Description of DFD Level 2

After DFD level 0 and 1 there is DFD level 2 it explains steps in further brief in this step the image is send as input and then it is modelled after the modulation procress the image is pre-procressed in pre- procressing step there includes the procress like grey-scale conversion, binarization, noise reduction,after all this steps the number plate area is detected and displayed to end user.

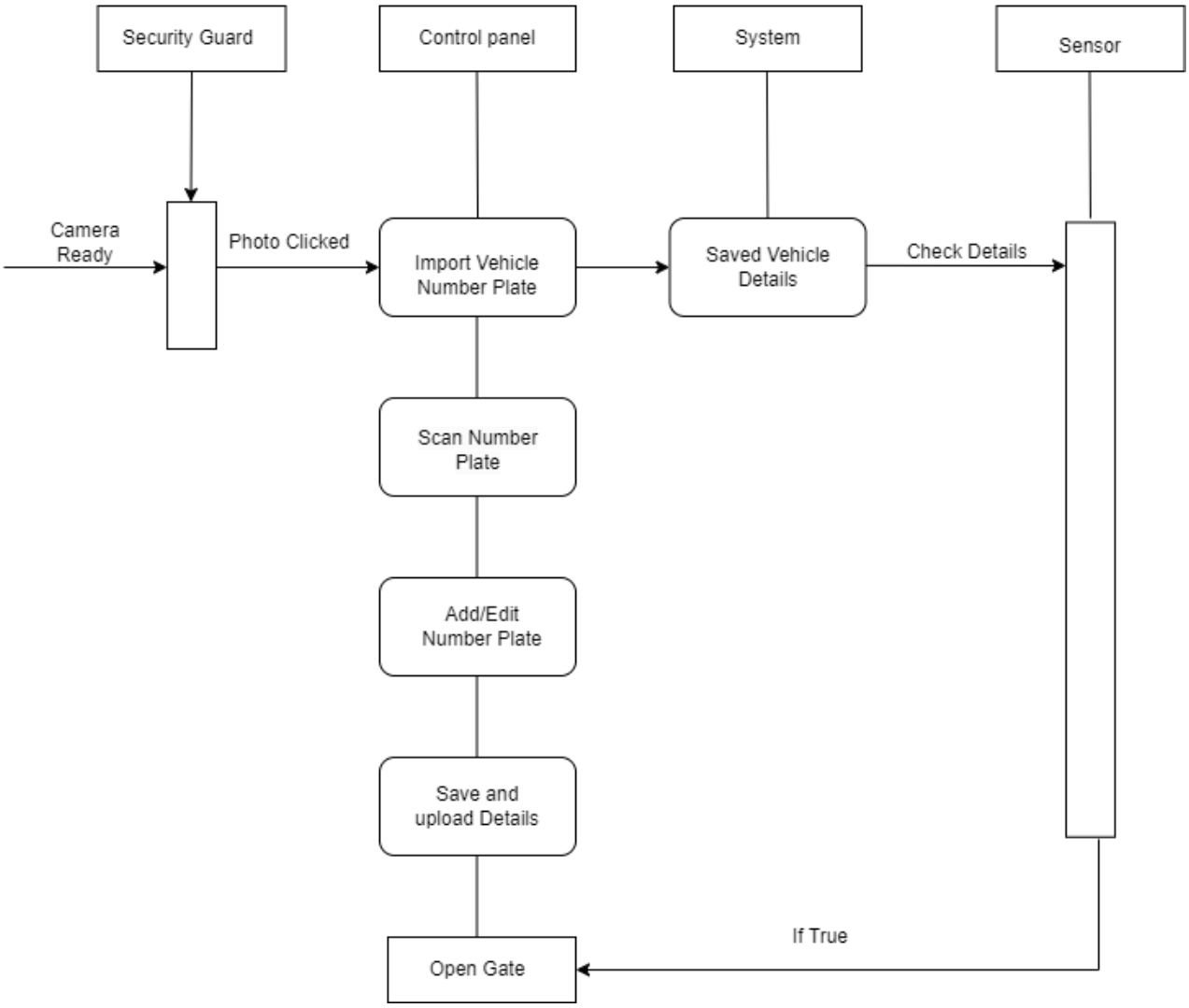


Fig 3.4.4: Sequence Diagram

3.4.4 Description of Sequence Diagram

A sequence diagram or system sequence diagram shows process interactions arranged in time sequence in the field of software engineering. It depicts the processes and objects involved and the sequence of messages exchanged between the processes and objects needed to carry out the functionality.

Chapter 4: Result and Analysis

4.1 Results and Analysis

The completion of our project resulted in the development of a web application that was able to track the area of license plate of vehicles and make a boundary box around it and show it to user.

Using the web application, the end user is able to access it from any device that supports a web browser. Then, the user can upload an image to identify the number plate. Once the number plate has been located, it creates a boundary box, crops it out and shows the cropped image of the detected license plate.

The project can be broadly divided tinto two parts. First, for the process of license plate detection and secondly, for flask app integration for better UI and UX.

For the process of number plate detection we used pre-defined libaries of sklearn and tensorflow which carried out the major part of preprocessing the data and normalization was done. Next we have used Convolutional Neural Network model and trained our data using train and test split. Then we feed and created our Deep Learning model which is able to detect the number plate area on the images. As the data fed into the model is relatively low than the high end systems it can detect the number plate in some conditons and sometimes it fails to detect. Nevertheless tried on 20 images, it could detect on 18 of them which is 90% accuracy which is okay but that 10% is quite hard to reach from the point of our supplied data.

Lastly, we integrated this Deep Learning Object detection model into our Flask app which runs in development mode on a local server and uses POST and GET methods to receive the uploaded file and store it locally. The user output is a cropped number plate which was predicted by our trained model

Chapter 5: Conclusion and Future Enhancement

5.1 Conclusion

In conclusion, vehicle number plate recognition systems have emerged as powerful tools with a wide range of applications across various sectors, from law enforcement and traffic management to security and commercial operations. These systems have show case their capability to improve safety, efficiency, and security on our roads and in parking facilities.

It offers the capability to efficiently and accurately identify and analyze number plates in real-time, thereby enhancing safety, security, and operational efficiency.

The vehicle number plate recognition system's based on a combination of advanced camera systems, image processing techniques, and artificial intelligence has demonstrated its potential to revolutionize urban environments. By automating the process of number plate recognition, the system minimizes reliance on manual intervention, leading to enhanced operational efficiency and considerable time savings.

Moreover, the vehicle number plate recognition system's adaptability to various environments, including low-light conditions, various environmental factors and various number plate formats, showcases its versatility.

As the field of automated surveillance continues to evolve, the vehicle number plate recognition system stands as a testament to the power of technology in creating safer, more efficient urban environments.With ongoing research and technological advancements, it holds the promise of even greater achievements in the future, further solidifying its role in the advancement of smart cities and intelligent transportation systems.

5.2 Future Enhancement

As this Vehicle Number Plate Recognition system is powerful in its own but it can be further enhance to improve its quality and accuracy. For forther enhancing we can perform Optical Caracter Recognition (OCR) that seperate character from number plate. Obtaining digital numberplate data from OCR enables it to be recorded in a database or a spreadsheet for better data management and future possibilites.

The real time operation system is less accurate to we can integrate AI with the system to increse the system accuracy.Advanced Pre-processing techniques can be used to reduce noise, image denoising and contrast enhancement methods to improve the quality of captured images.

By incorporating these advancements, the Vehicle Number Plate Recognition system can achieve higher accuracy, adaptability, and efficiency, further solidifying its role in revolutionizing urban surveillance, traffic management, and law enforcement. Continuous research and development in these areas will pave the way for even more sophisticated and effective Vehicle Number Plate Recognition systems in the future.

REFERENCES

[1] Harpreet Kaur, M. B. (Nov-2012). VEHICLE LICENSE PLATE DETECTION FROM EDGE DETECTION AND MORPHOLOGICAL OPERATORS. International Journal of Engineering Research & Technology(IJERT) ISSN:2278-0181 Vol.1 Issue 9

[2] A. M. Patil, S. H. Patil, R. R. Manza (2016) A SURVEY ON AUTOMATIC NUMBER PLATE RECOGNITION SYSTEM.

[3] Bailmare, M. H. (2013). A REVIEW PAPER ON VEHICLE NUMBER PLATE RECOGNITION USING IMPROVED CHARACTER SEGMENTATION METHOD. International journal of Scientific and Research Publication ,Volume 3

[4] Radhika S. Dangare, P. G. (2016). AUTOMATIC NUMBER PLATE RECOGNITION for use in DIFFERENT COUNTRIES using RASPBERRY PI (REAL TIME APPROACH). International Journal of Science and Research (IJSR)

[5] Saghaei, H. (2016). PROPOSAL FOR AUTOMATIC LICENSE AND NUMBER PLATE RECOGNITION SYSYTEM for VEHICLE IDENTIFICATION. International Conference on New Research Achievements in Electrical and Computer Engineering. Shahrekord,

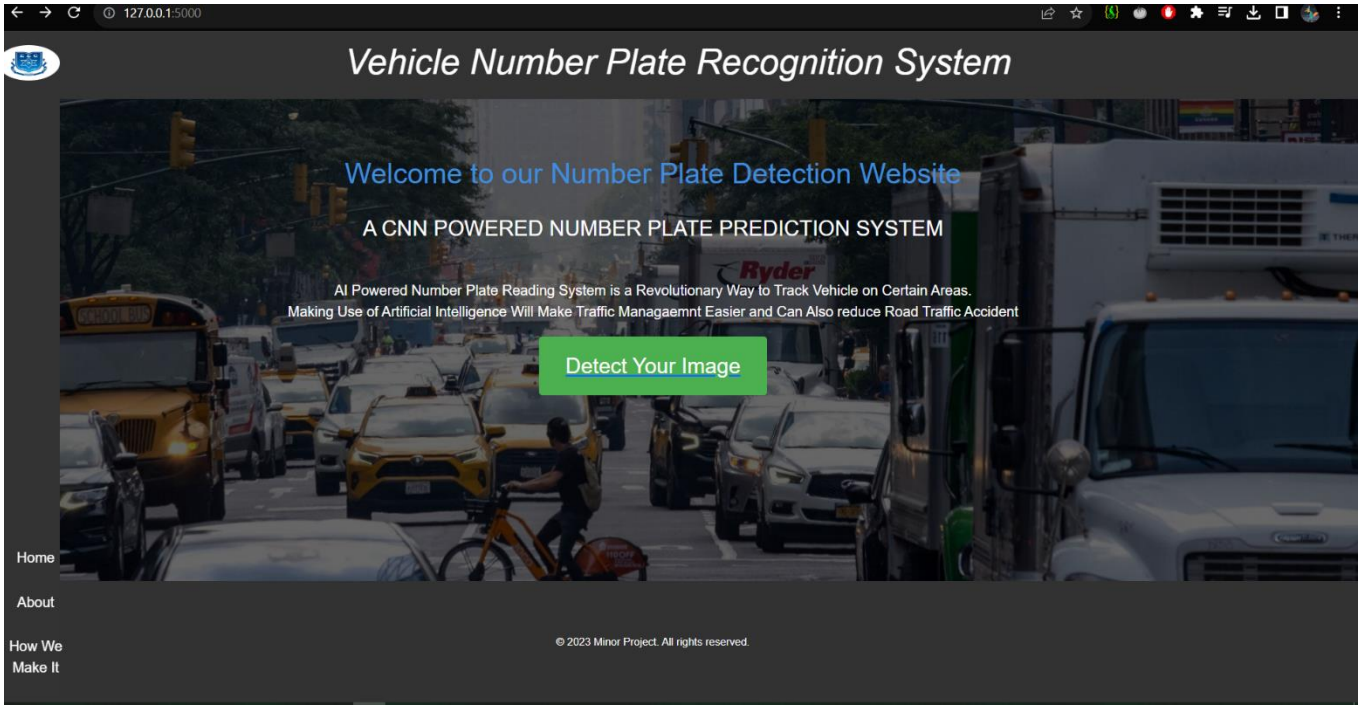
[6] Alamsyah, N., Fauzan, M. N., Putrada, A. G. & Pane, S. F., 2022. Autoencoder Image Denoising to Increase Optical Character Recognition Performance in Text Conversion. s.l., 2022 International Conference on Advanced Creative Networks and Intelligent Systems (ICACNIS), Advanced Creative Networks and Intelligent Systems (ICACNIS), 2022 International Conference on,20221123,IEEE Xplore Digital Library. (accessed Jun. 5, 2023).

[7] A. Jaware, “Final report ANPR - automatic number plate recognition - A report on automatic number plate,” Studocu, https://www.studocu.com/in/document/banaras-hindu-university/corporate/final-report-anpr-automatic-number-plate-recognition/9270712 (accessed Jun. 6, 2023).

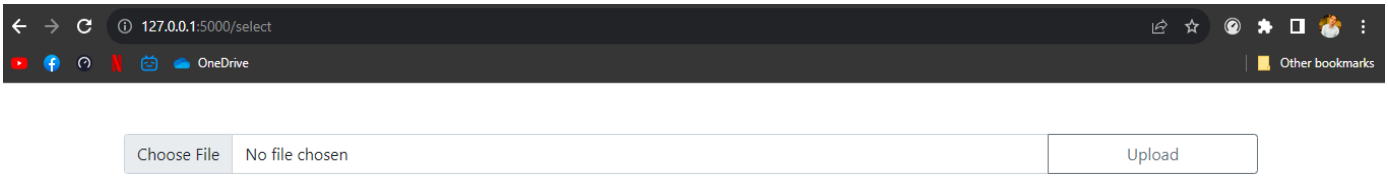
APPENDIX

Appendix 1

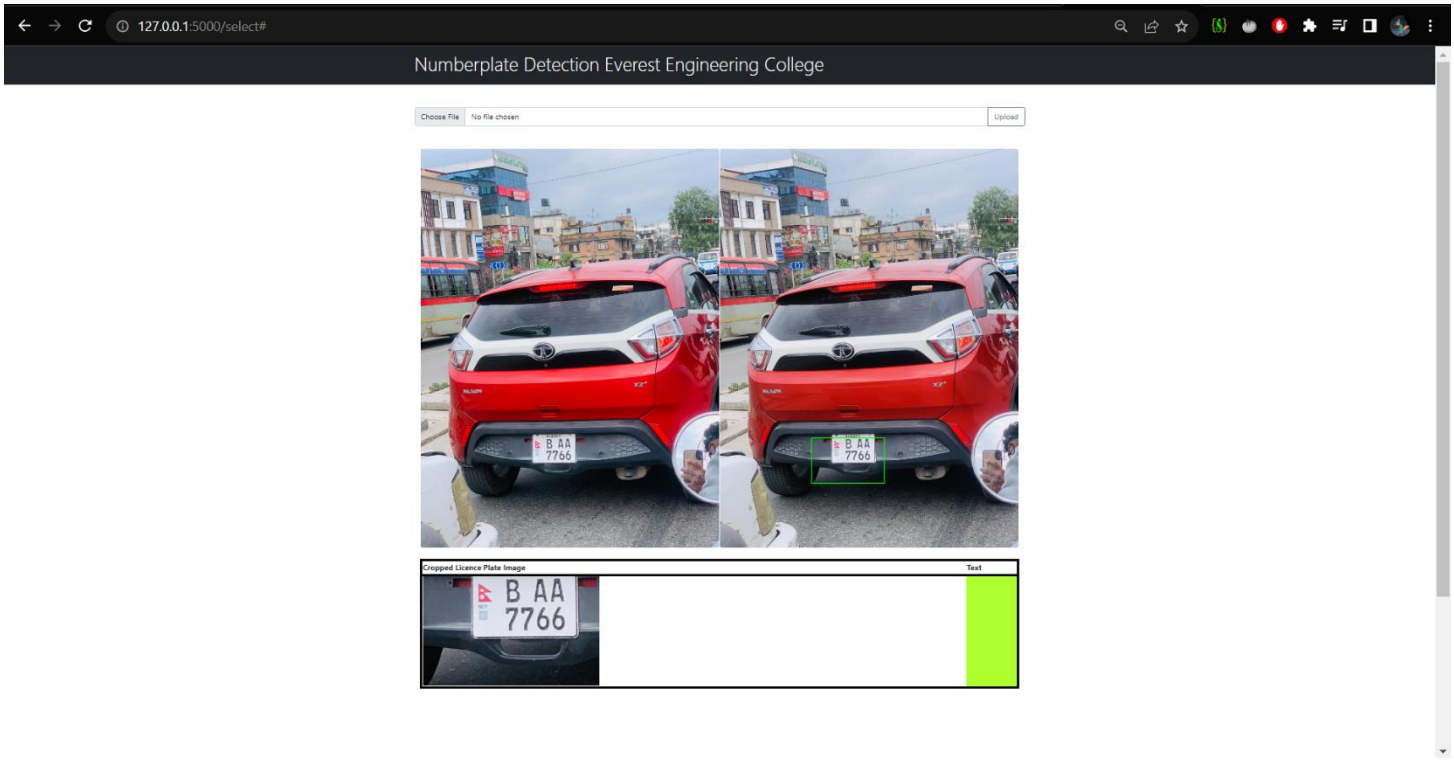
Appendix 2



Appendix 5



Appendix 4


127.0.0.1:5000/select#
Numberplate Detection Everest Engineering College
Choose File
No file chosen
Upload
B AA
7766
Cropped Licence Plate Image
Text

Appendix 5