A Major Project Final Report on

Number Plate Recognition and Speed Estimation

Submitted in Partial Fulfillment of the Requirements for

The Degree of **Bachelor of Engineering in Information Technology**

under Pokhara University

Submitted by:

Kalpana Karn, 14418

Priya Ghimire, 14427

Sangeeta Gurung, 14436

Shreya Upadhyay, 14439

Shweta Shakya, 14440

Date:

8/12/2018

Under the supervision of

Er. Saroj Bista

Department of Information Technology

NEPAL COLLEGE OF INFORMATION TECHNOLOGY

Balkumari, Lalitpur, Nepal

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. It is an utmost privilege to express our sincere regards to the project supervisor, **Er. Saroj Bista** for his valuable guidance, encouragement and support, throughout the duration of the project. We deeply express our sincere thanks to him for encouraging and allowing us to present the project on the topic "Number Extraction and Speed Estimation" for the partial fulfillment of the requirements leading to the award of the degree of bachelors of engineering in information technology.

We take this opportunity to thank our associate professor, **Dr. Roshan Chitrakar** for providing us the opportunity to embark this project. We are also highly obliged to all the lecturers, who have directly or indirectly helped us to gain the successful completion of the project. We also express our respect and love to our guardians and all other family members for their love and encouragement throughout our project development period. Last but not the least, we express our thanks to all our beloved friends for their cooperation and support.

ABSTRACT

In the present context of Nepal, we see various surveillance system used on some parts of the city but they are not capable of extracting the number plates of the vehicles or the speed on which they run. The project titled “Number Plate Recognition and Speed Estimation” is a system which recognizes the number plate of the vehicles using neural network. It accomplishes the task in three successive steps which are character detection, number plate extraction and speed estimation. The acquisition of image or video is done by the user. Then, the system reads the image or video and the preprocessing of the image or the video occurs. When we provide image as an input, the character detection of the number plate image occurs preceded by the preprocessing. The detected characters are then segmented and extracted finally leading to the number plate extraction. When the video is given as input, firstly the system reads the video and a reference frame is selected. The image is then converted into black and white and bounding box is created. The speed is calculated by dividing the total displacement of moving object with the total time. The software requirements for this project is MATLAB software.

Keywords: *Neural network, Number plate recognition, Speed estimation, Filter, MATLAB*

List of Abbreviations

ANN:	Artificial Neural Network
ANPR:	Automatic Number Plate Recognition
LPR:	License Plate Recognition
AI:	Artificial Intelligence
MATLAB:	Matrix Laboratory
SRS:	System Requirement Specification
GUIDE:	Graphical User Interface Development Environment

Table of Contents

List of Figures

1. INTRODUCTION

The traffic on the roads, in the past few years has increased immensely as a result of which the problems such as over speeding, car theft and others has also increased. To overcome such problems vehicles number plate recognition and speed estimation plays an important role in traffic control system.

This project is being designed using MATLAB software which provides the information regarding number plate and speed of the vehicles.

1.1 Project Overview

The system provides us with number plate of the vehicle and speed of the vehicle. The two successive steps that are included in this project are:

Number Plate Recognition

It consists of two parts:

I. Number Plate Character Segmentation

Here we extract character from the input image. The unwanted noise is removed from the content and is left with number plate characters. The characters are then separated.

II. Number Plate Character Recognition

Recognition is done using artificial neural network. Firstly, the rgb image is converted into grayscale and then to black and white. After that, we extract the features of the characters using zoning algorithm. The extracted features, with the help of neural network function, helps us to recognize the character of the number plate.

Speed Estimation

Speed of the moving vehicle is estimated by using simple formula for speed. The video is read from which the area of interest is extracted using bounding box construction and the total displacement of the moving object is calculated in the consecutive frame by centroid method. The division of this by the total time gives us the speed.

1.2 PROBLEM STATEMENT

As we take a look at the existing surveillance system, all we find is the close circuit cameras. The existing manual system is not that effective and is hard on the traffic officers. It requires much of the storage for storing the video data and then using a separate system to identify the required entity. This also adds up to the processing time and causes much delay in case of urgency instead of fast processing.

The systems, if were efficient enough to store the record of the number plates of the vehicles and the speed they are travelling with, would not only help in faster execution but would also be helpful for the future reference and investigation. It also would have lessened the manual efforts to do the same work. So, how do we overcome this?

It becomes very important for us to design a system which is capable of recognizing the number plate characters and estimate the speed of the moving vehicles through an easy and efficient way. The need to combat this situation is where the motivation for the development of our project comes from. This project, when used with the existing surveillance system, aids security, execution speed and future reference.

1.3 OBJECTIVES

The main objectives of this project are:

- To implement neural network for character recognition.
- To recognize the number plate of vehicles.
- To estimate the speed of vehicles.

1.4 SCOPE AND IMPORTANCE

- This project performs segmentation of characters.
- It recognizes the characters using neural network.
- It estimates the speed of vehicles in motion.

This project can be useful in following areas:

- This system can be used in the entrance of restricted areas for security control.
- The number plate can be used to retrieve more information about the vehicle and its owner which can be used for further processing.
- This system can be used to check and limit the speed of vehicle's in and around accident prone areas.

2. LIMITATIONS

The project is unable to provide the expected outcomes under following circumstances:

- Poor lighting and low contrast due to overexposure, reflection or shadows.
- The vehicle is too far.
- Part of the number plate being blocked by an object or dirt.
- Inefficient capture due to improper placement of camera angle.
- The number plates having the characters in two rows.

3. LITERATURE REVIEW

Several approaches have been proposed for recognizing the number plates and the estimation of speed of vehicles from images and video frames. Lots of these approaches have been implemented and are being used worldwide but are not that effective in Nepal. Various algorithms have been and are being used for character segmentation and character recognition. Some of which are explained as follows.

3.1 Binary Image Processing

Processing a RGB image takes up much of the processing time and space. The compilation also becomes quite difficult and the matrix obtained is also 3x3. So, binary image processing has emerged quite popular for character segmentation. It is easy to use and takes up lesser processing time.

For the seriously degraded number plates, a morphology-based adaptive approaches are used. Here, for the identification of noises, thickening and pruning algorithms are applied to the binary image.

Another method used for the character segmentation is contour tracking and modelling. Here, firstly the ordinary fast marching approach is used which helps to determine the coarse location of each character. This is followed by the special fast marching approach so as to segment the exact boundaries. After further processing, a final merged results of segmentation are obtained.

3.2 Gray-Level Processing

There is no any unique thresholding technique that could be dynamically used in almost all situations. So, in many cases, histogram processing techniques are used for character segmentation. Here, the intensity gradient histograms are firstly generated after which thresholding values are determined.

As we know that, a single global threshold doesn't always produce the required results, so we also see the use of local adaptive thresholding. Here, the threshold is determined by dividing the image into mxn blocks. After determining the threshold values for each block, we could see the dynamic binarization method as a result.

These methods are generally used for the system incorporating the sensors. Here, the character size and distance from the vehicle to the sensor are pre-estimated. Noises are removed and image is enhanced so as to detect, centralize and normalize the number plate image in the character segmentation phase.

3.3 Classifiers

For the recognition of characters, classifiers are often seen been used. If we use the Hidden Markov model, the character recognition begins with the preprocessing and parameterization of the region of interest which was obtained after the character segmentation process. It greatly demands the good analysis of character as a result of which there is restriction of distance. Classification could be either multistage or a combination of multiple classifiers parallel.

When working through the artificial neural network(ANN), the characters of the number plates are identified using a feed forward neural network. Here the training is done using back propagation algorithm which propagates the error backwards so as to make the necessary changes in the weights resulting in the lesser value of error each time than the previous one. The recognition of character also greatly depends upon the hidden layers and neurons used. A character is recognized correctly only when its corresponding output neuron is nearly equal to one and all other output neurons are nearly equal to zero.

3.4 Template Matching

Pattern matching or template matching is the technique which is appropriate only for the characters that are of single, fixed font, specified size and are not rotated at any angle. In some systems that use template matching for the character recognition, the calculation of root mean square for all the shifts occurring of template over a subimage is the basis of recognition process.

3.5 Existing Systems

Few of the recognition systems in use are:

Automatic Number Plate Recognition System

Automatic Number Plate Recognition(ANPR) is used to recognize registered number plates. ANPR is used by police forces around the world for law enforcement purposes. It can be used to search stolen vehicles and solving other crimes. It is described as a form of mass surveillance. ANPR was invented in 1976 in the United Kingdom but was not widely used until the 1990s. It uses closed circuit television, road rule enforcement cameras or cameras specially designed for the task.

License Plate Recognition System

License Plate Recognition System(LPR) enables high performance automatic detection and recognition of license plates in real time or from recorded footage. LPR system is AI driven system that collects images, timestamps and coordinates. It is used to identify stolen vehicles and improve site security by distinguishing problematic vehicles. This system is similar to the earlier discussed ANPR in the sense that it is also unable to determine the speed of vehicle.

Real Time Speed Estimation from Video Sequences

Here, there is the use of calibrated camera which is mounted on the required place. The camera calibration is done using the reference equation which is fit for particular hardware. This system firstly uses adaptive background subtraction along with the algorithm that takes the differences from three frames. This system uses regression model for training. It detects the height of the number plate as a result of which it becomes helpful for the calculation of relative distance. Finally, the speed is calculated by using the frame rate. Here, if the background changes, the camera need to be calibrated again.

Most of the existing systems usually focus on either the number plate recognition or a separate system for speed estimation.

4. METHODOLOGY

4.1 Number Plate Character Detection

In order to detect the character of number plate, we firstly perform image acquisition. The input image of number plate is a RGB image as a result of which, we convert it into grayscale. We do so because pre-processing a RGB image would generate a 3x3 matrix where each red, green and blue components have 255 different values. This not only increases the complexity but also the processing time.

As we move forward in the pre-processing, the image after median filtering is dilated and also subjected to eroding. Then morphological gradient of these are used in edge enhancement of the character. We then fill the region of interest of image. In order to ensure character isolation, thinning is performed. Finally, the detected characters are bounded by green rectangles.

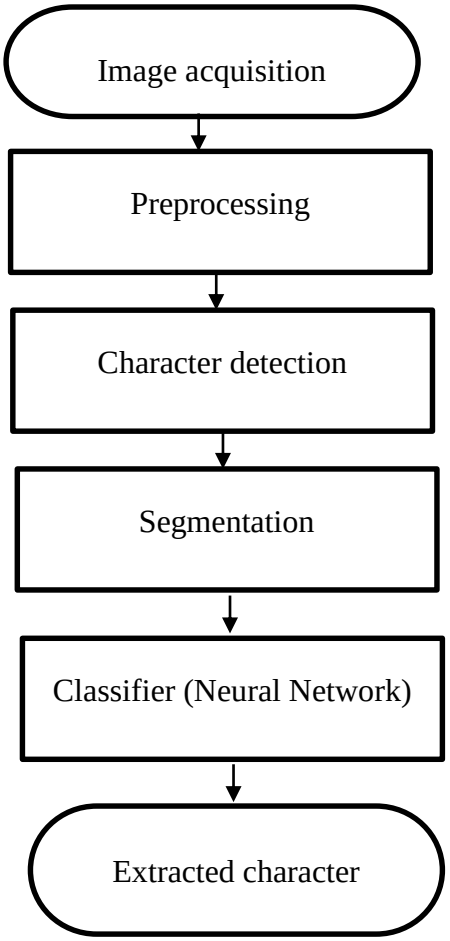

Figure 1: Number plate extraction

4.2 Number Plate Character Recognition

The detected characters are stored in the form of array images. Each image is then subjected to feature extraction which is then fed to the neural network for the actual extraction of character and identification of the classifier.

The extraction of features is done by using zoning algorithm. It is one of the most effective methods for extracting distinctive characteristics from pattern recognition. For this project, the individual images are resized to 50*50 pixels and the images are zoned into 9 equal size. Feature extraction are applied to individual zoned.

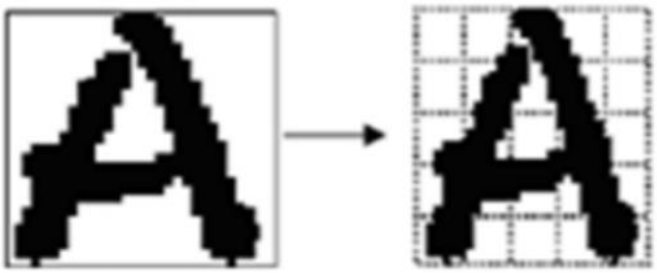

Figure 2: Zoning

The neural network that we use is a two- layer feed forward network. Hidden layer and output layer comprises the two layers. The hidden layer has sigmoid activation function. The numbers of hidden layer neurons also play an important role here. The number of hidden layer neurons used here are 500. The neural network is trained by using back propagation algorithm. This algorithm, firstly, propagate the input values through the networks and calculates errors, and then we back propagate the errors through the network backwards to adjust the connection weights to minimize the error. The feature extraction function provides us with 85 features which is the input of our neural network. There are 13 possible outputs through this network.

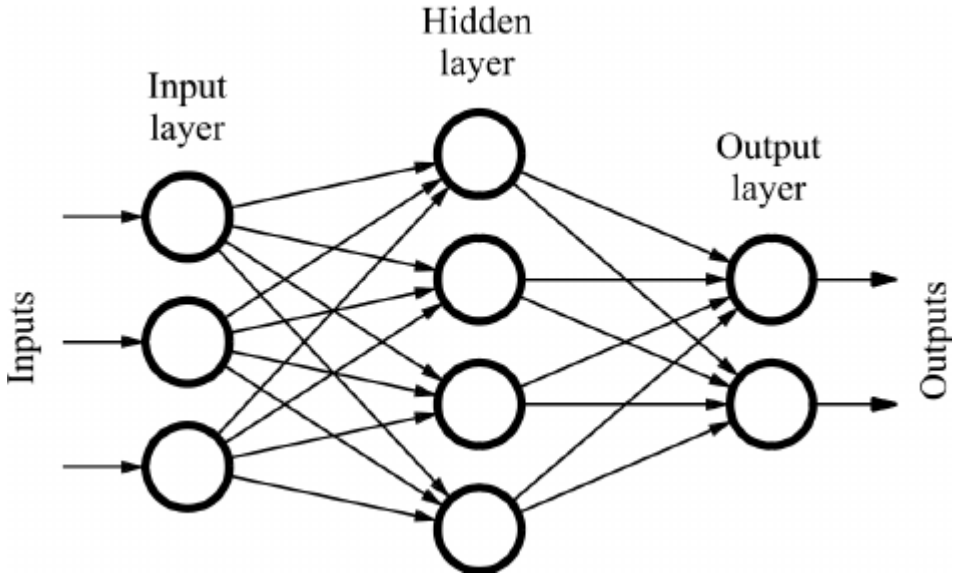

Figure 3: Feed forward neural network

4.3 Speed Estimation

Firstly, we take video as an input. The video is converted into grayscale. After doing this, the frames from the video is extracted until the complete duration of the video. There is a use of step function. The frame is not extracted in the required data type as a result of which, we convert its data type. We then use Lucas-Kanade method for optical flow estimation which uses the differential method. As for this method the flow is assumed to be constant for a pixel under consideration and its local neighborhood. We then define the threshold velocity using the mean value of the moving object as a result of which the segmented object is identified.

The segmented object is then bounded by the help of bounding box. We then calculate its centroid. The distance is then measured with the help of distance traveled by the centroid. Finally, the speed is calculated by using the simple speed formula which multiplies the distance with the frame rate and then divide it by the total number of frames.

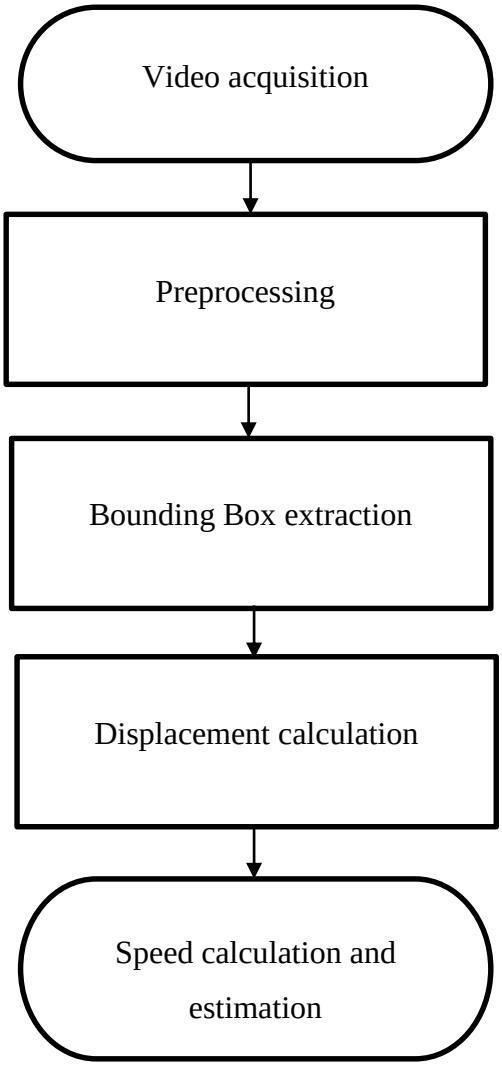

Figure 4: Speed estimation

4.4 Software Development Life Cycle

The framework we will be using for developing this project is Incremental model and extreme programming.

4.4.1 Incremental Model

This model combines linear sequential model with the iterative prototype model. New functionalities will be added as each increment is developed. The phases of the linear sequential model are: Analysis, Design, Coding and Testing. The software repeatedly passes through these phase in iteration and an increment is delivered with progressive changes

Analysis phase

In this phase, analysis will be performed in order to find out the requirements of the system. The outcome of this phase would be a SRS which is an acronym for "System Requirement Specifications".

Design phase

In this phase, the SRS would be translated into the system's design. Context Diagram, DFD, ER – Diagram, Use Case Diagram and Class Diagram will be developed.

Coding phase

In this phase, coding will be done according to the design and a working system will be developed by the end of this process.

Testing phase

In this phase, the system will be tested. With each testing, a list of changes to the system developed is suggested and the changes will be applied to the software and the software would be delivered as a successive increment until a satisfying system is achieved.

We use MATLAB programming and software for the successful completion of this project. The project is completed in three iterations. After the completion of first iteration, the character recognition part is completed. The completion of second iteration leads to number plate extraction. Finally, the completion of third iteration leads to number plate extraction with estimation.

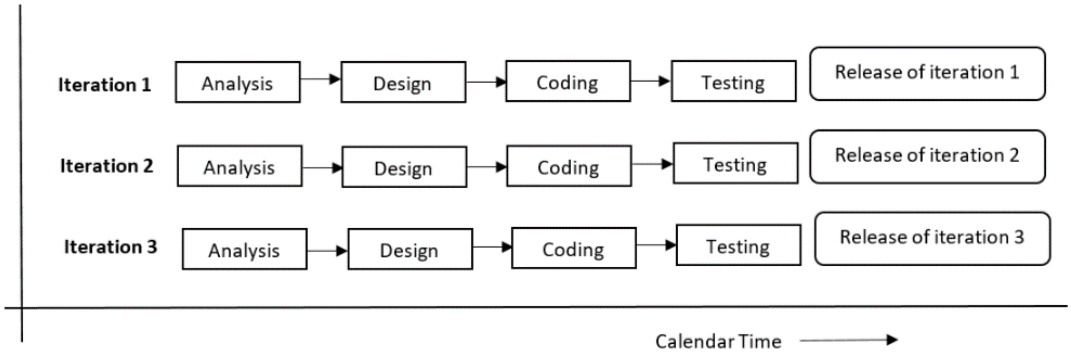

Figure 5: Incremental model

4.4.2 Extreme Programming

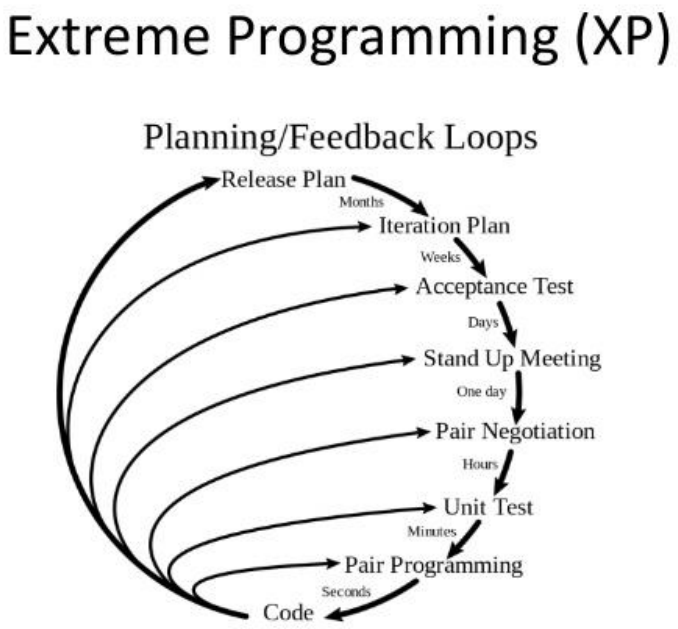

Figure 6: Extreme programming

The Extreme Programming software development process is an agile method focused on providing the highest value for the customer in the fastest way possible.

Design

The first step in extreme programming software developing process is design. The three major guiding techniques adopted during the design phase are considerations of simplicity, use of CRC cards, and spike solutions.

Iteration Meetings

Extreme Programming progresses through iterations. The iterations in Extreme Programming methods ensure development of objects and classes that, when combined, provide the complete product.

Coding

Extreme Programming software development process gives priority to the actual coding over all other tasks.

Standards related to coding include:

- Adherence to metaphor or standards on names, class names, and methods.
- Using uniform styles and formats to ensure compatibility among the work of different team members.
- Pair programming or developing code by two programmers working together on a single machine to produce higher quality code at the same or less cost.

Acceptance Test

Acceptance testing is of key importance in Extreme Programming. The customer provides the developer with functional or acceptance tests to validate the features. The developers subject the developed code to this test and continue to modify the code until the code passes the test. The test runs several times for confirmation.

Feedback

Extreme Programming adopts a system of continual feedback to make changes at the development change itself rather than at the end of the project. During the project development process, developers provide customers with acceptance test scores and demonstrations and take feedback. Such feedback might entail change in requirements, and the code is redone at this stage itself. A feedback system also exists between developers and project managers through daily meetings.

Completion

At the end of the iteration, the programmers deliver a working system without bugs to the customer. The developed code passing the specified acceptance test and the customer having no additional requirements from the user story developed at the start of the iteration ends the specific iteration.

4.5 System Design

There are a number of diagrams during system design which are listed below:

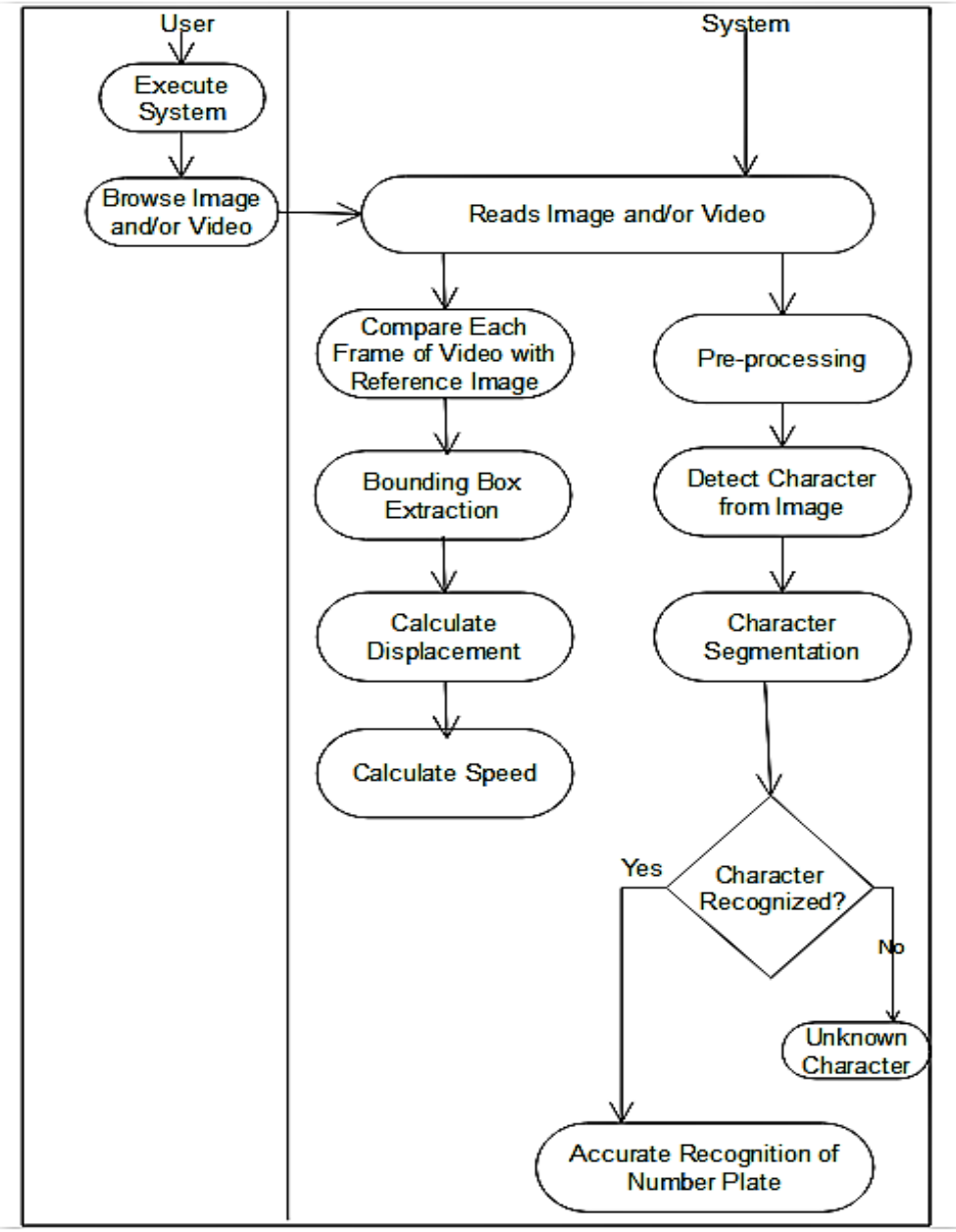

Figure 7: Activity diagram

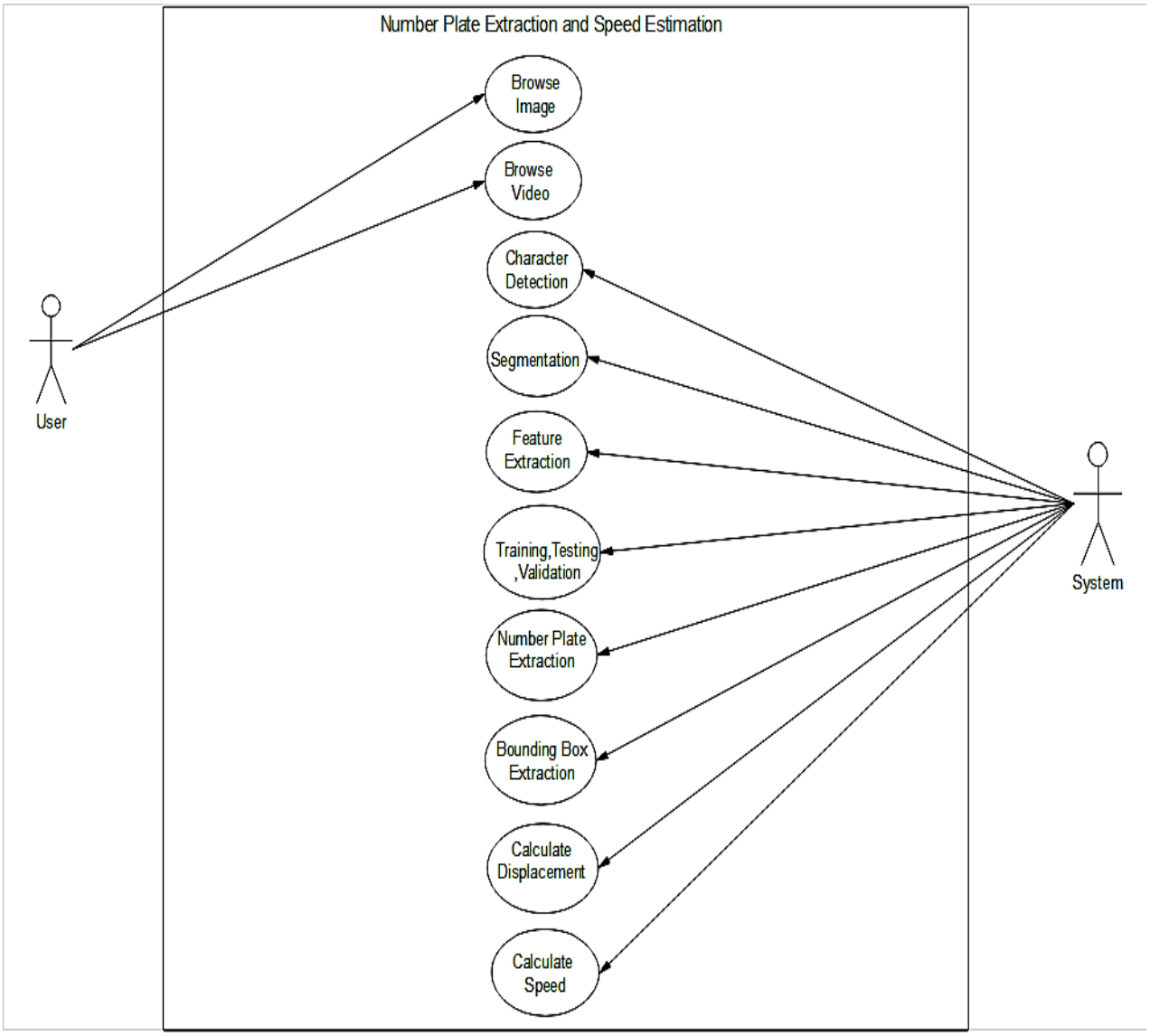

Figure 8: Use case diagram

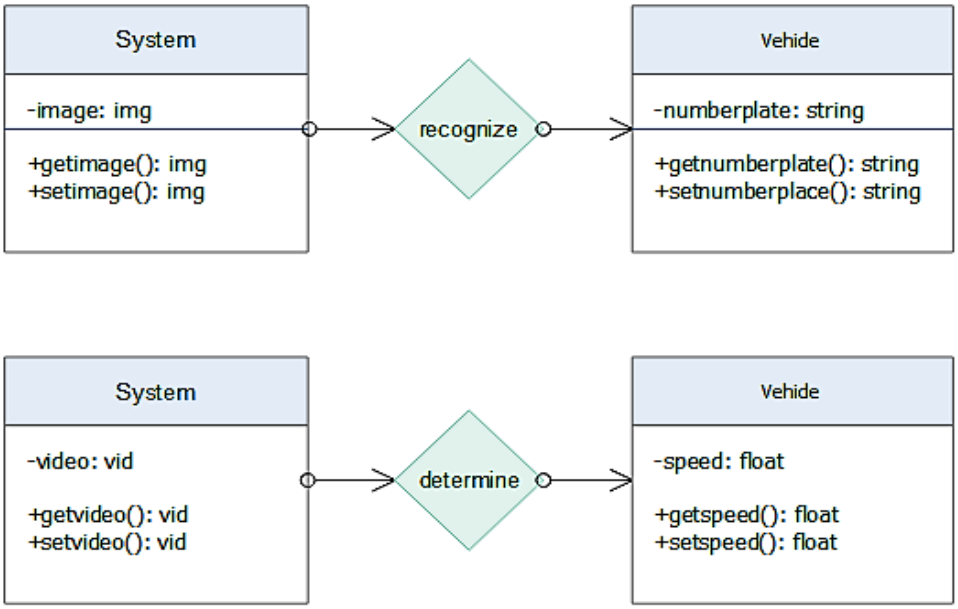

Figure 9: Class diagram

5. DELIVERABLES

At the end, this project will deliver:

- A system that recognizes number plates and also estimates the speed of the vehicles.
- A project report documentation**.**

6. RESULTS AND OUTPUTS

As our work progresses, the result that has been obtained is a system that extracts the number plate from the given image. When we first provide the image of a number plate as an input, the image is subjected to preprocessing where the image is converted to grayscale. In order to remove noise, median filter is used. Then, the morphological gradient of dilation and erosion is used for edge enhancement. After further processing, the characters are detected which are headed towards the character segmentation part. The segmented characters are then forwarded for the feature extraction. Finally, with the help of neural network, number plate is extracted. Also, when we give video as an input, we

6.1 Character Recognition

The foremost step in order to extract the characters of a number plate is to design a neural network which recognizes the characters. We train the neural network with the dataset of the characters which could be possibly present on the number plate after performing the feature extraction process.

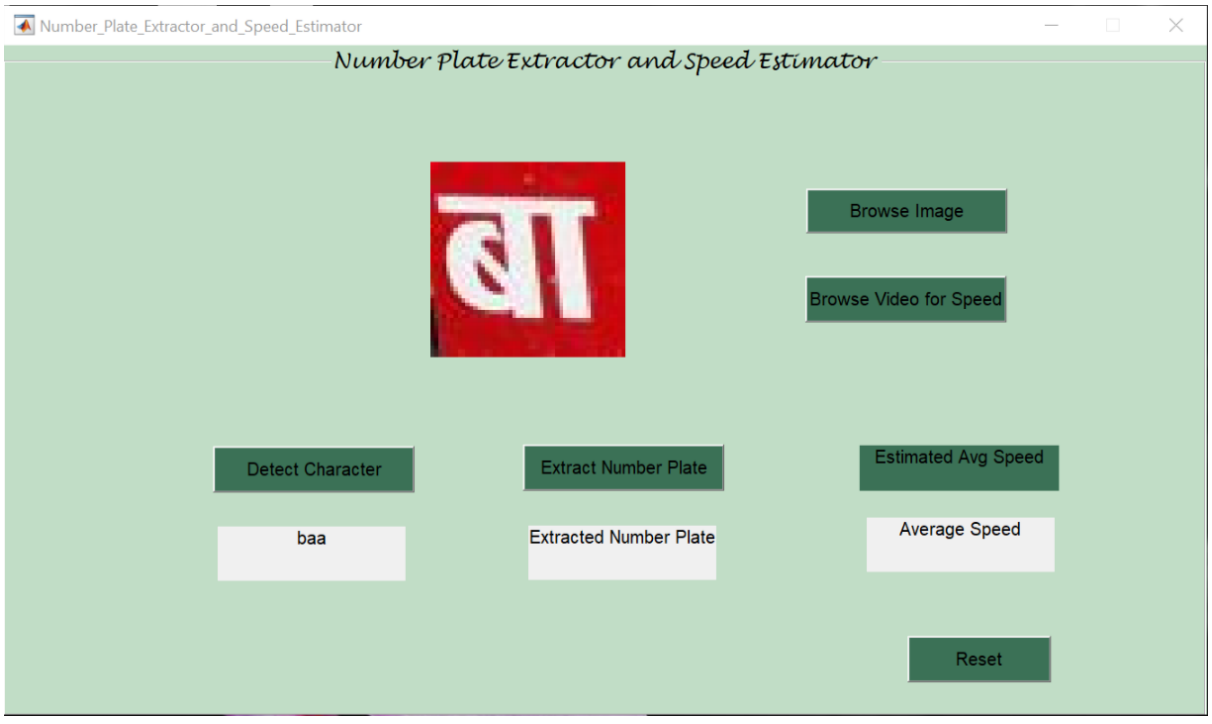

Figure 10: Character Recognition

6.2 Character Detection in Image

When we provide the image of a number plate as an input, the system is able to detect the characters present on it.

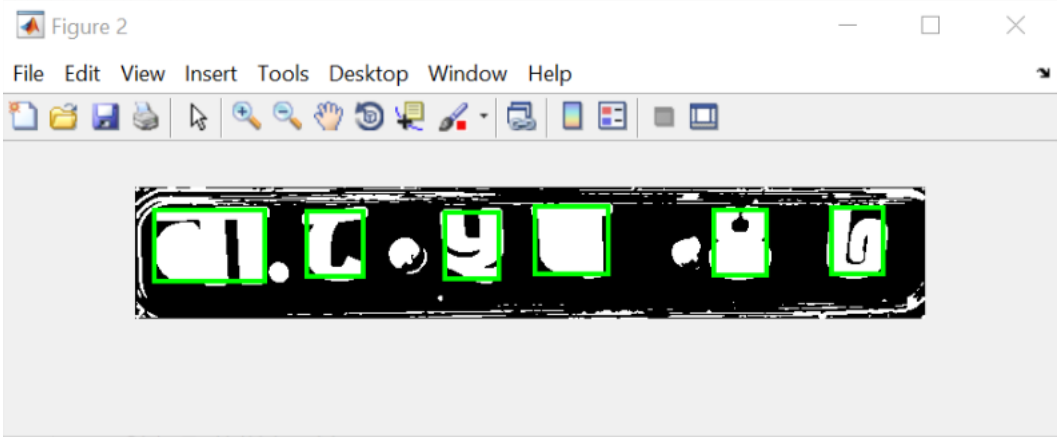

Figure 11: Character detection in image

6.3 Segmentation

As we detect the characters on the image, we perform character segmentation which leads to the number plate extraction process.

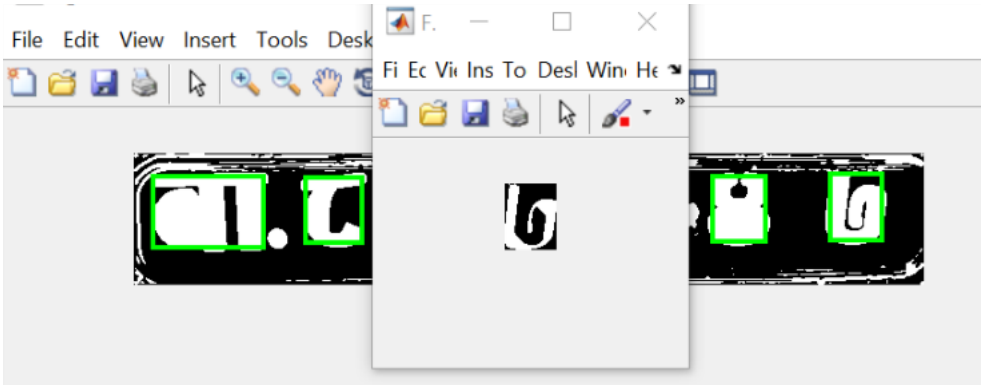

Figure 12: Segmentation

6.4 Number Plate Extraction

When we first provide the image of a number plate as an input, the image is subjected to preprocessing where the image is converted to grayscale. In order to remove noise, median

filter is used. Then, the morphological gradient of dilation and erosion is used for edge enhancement. After further processing, the characters are detected which are headed towards the character segmentation part. The segmented characters are then forwarded for the feature extraction. Finally, with the help of neural network, number plate is extracted.

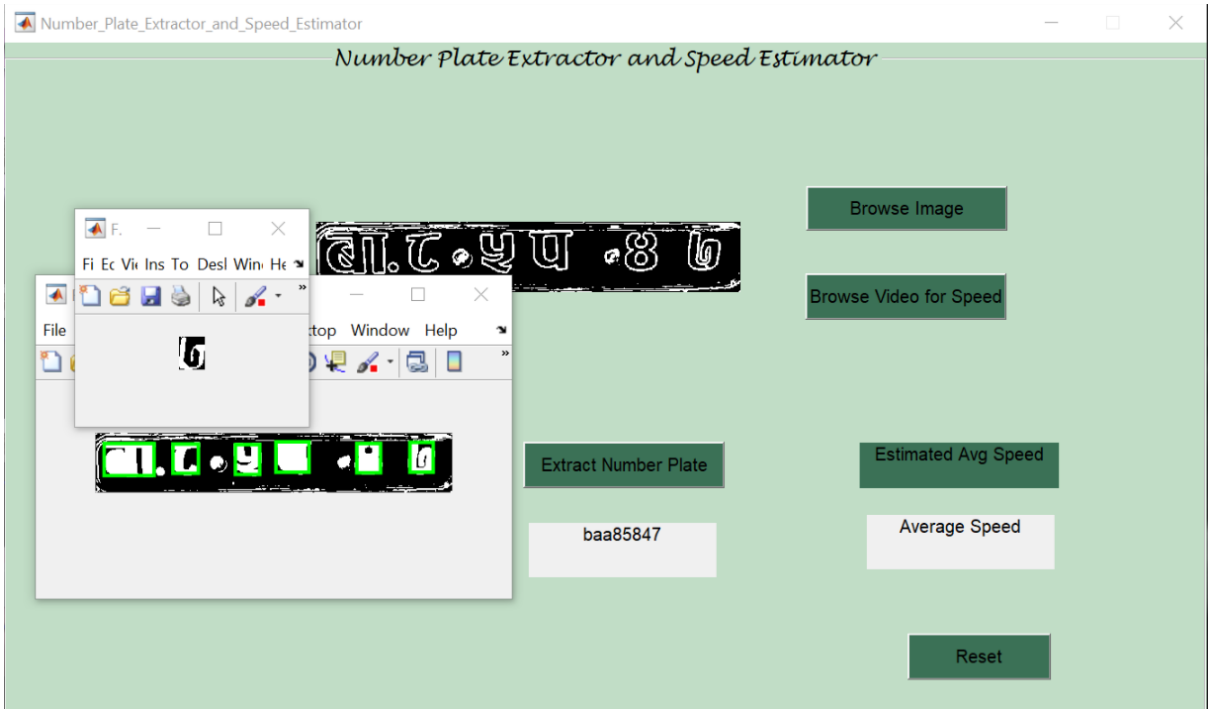

Figure 13: Number Plate Extraction

6.5 Speed Estimation

Firstly, we take video as an input which is converted into grayscale. After doing this, the frames from the video is extracted until the complete duration of the video. There is a use of step function. The frame is not extracted in the required data type as a result of which, we convert its data type. We then use Lucas-Kanade method for optical flow estimation which uses the differential method. As for this method the flow is assumed to be constant for a pixel under consideration and its local neighborhood. We then define the threshold velocity using the mean value of the moving object as a result of which the segmented object is identified. The segmented object is then bounded by the help of bounding box. We then calculate its centroid. The distance is then measured with the help of distance traveled by the centroid. Finally, the speed is calculated by using the simple speed formula which multiplies the distance with the frame rate and then divide it by the total number of frames.

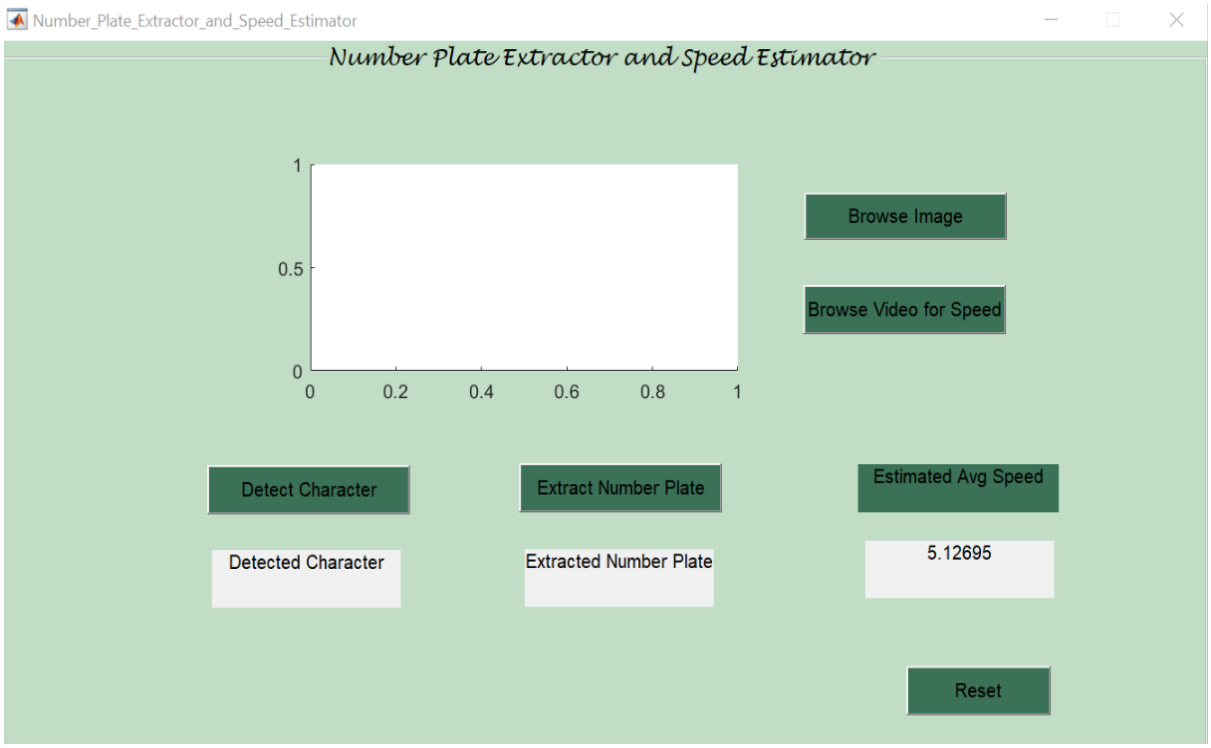

Figure 14: Speed Estimation

7. TASK AND TIME SCHEDULE

The project schedules will be performed as per the requirements and time constraints involved. Numerous informal suggestions, from the supervisors, which had assisted a lot in the development are not included in the chart. The vacation part and the part of examinations where the project couldn't be executed has also not been included in the given chart. Both the examination and the vacation part comprises almost two months. This project is scheduled to be completed in about 10 weeks. The figure below provides a better understanding of the tasks and time division.

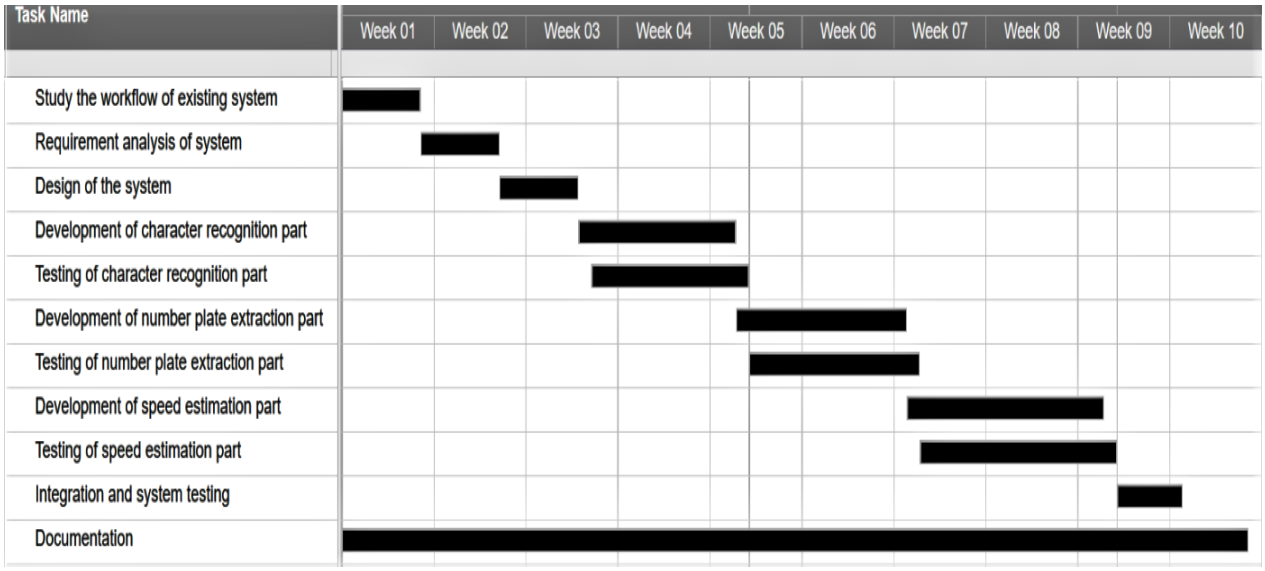

Figure 15: Gantt Chart

Division of Work

Task	Members
Requirement analysis and specification	Kalpana, Priya, Shreya
Undertake analysis of system	Sangeeta, Sweta
Design system	Kalpana, Shreya
Coding	Priya , Shreya, Sweta
Testing and debugging	Kalpana, Sangeeta, Sweta
Overall system test	Kalpana, Priya, Shreya
Documentation	Kalpana, Priya, Sangeeta, Shreya, Sweta

Table 1: Division of work table

8. FUTURE WORKS

This project extracts number plate from the input image while estimates speed from the input video so further future extensions could be made to this project. Some of them are enlisted below:

- Number plate extraction of the plates with the characters in two rows.
- Extraction of the number plates from the video.
- Real time implementation of the current system.

9. REFERENCES

[1] Rad, Arash Gholami, Abbas Dehghani, and Mohamed Rehan Karim. "Vehicle speed detection in video image sequences using CVS method." International journal of the physical sciences. –Academic Journals 5, no. 17 (2010): 2555-2563.

[2] Ibrahim, Osman, Hazem ElGendy, and Ahmed M. ElShafee. "Speed Detection Camera System using Image Processing Techniques on Video Streams."

[3] Liu, Wen, Fumio Yamazaki, T. Thuy Vu, and Yoshihisa Maruyama. "Speed detection of vehicles from aerial photographs." In Proc. 27th Asian Conference on Remote Sensing. 2007.

[4] Pornpanomchai, Chomtip, and Kaweepap Kongkittisan. "Vehicle speed detection system." In Signal and Image Processing Applications (ICSIPA), 2009 IEEE International Conference on, pp. 135-139. IEEE, 2009.

[5] Lee, Jong-Sen. "Digital image smoothing and the sigma filter." Computer Vision, Graphics, and Image Processing 24, no. 2 (1983): 255-269.

[6] Sobel, Irwin, and Gary Feldman. "A 3x3 isotropic gradient operator for image processing." a talk at the Stanford Artificial Project in (1968): 271-272

[7] Lewis, J. P. "Fast template matching." In Vision Interface, vol. 95, no. 120123, pp. 15-19. 1995.

[8] A Real-time Vehicle License Plate Recognition (LPR). VISL, Technion, 2003

[9] https://en.wikipedia.org/wiki/Automatic_number-plate_recognition

[10] Roger S. Pressman, Software Engineering - A Practitioner's Approach, Fifth Edition

[11] Gantt Chart : https://app.smartsheet.com/b/home

[12] https://www.mathworks.com/help/?s_tid=hp_ff_l_doc

[13] https://www.youtube.com/watch?v=p_g-g7C3uHw

[14] M. Kirby and L. Sirovich, "Application of the Karhunen- Loève procedure for the characterisation of human faces," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, pp. 831-835, Dec. 1990.

[15] K.K.Kim K.I.Kim J.B.Kim H. J.Kim, "Learning-Based Approach, for License Plate Recognition", in proc. IEEE Signal Processing Society Workshop, Neural Networks for Signal Processing, Vol. 2, 2000, pp. 614 - 623.

[16] Hsi-Jian Lee, Si-Yuan Chen and Shen-Zheng Wang, “Extraction and Recognition of License Plates of Motorcycles and Vehicles on Highways”, in proc Int. Conf. on Pattern Recognition (ICPR), 2004, pp. 356 – 359.

[17] F. Martin, M. Garcia, and J. L. Alba, “New methods for automatic reading of VLP’s (Vehicle License Plates),” in Proc. IASTED Int. Conf. SPPRA, 2002.

[18] L. Dlagnekov, License Plate Detection Using AdaBoost, Mar. 2004, San Diego, CA: Comput. Sci. Eng. Dept., Univ. California, San Diego. [Online].

[19] https://www.youtube.com/watch?v=4v8hJbD_B2s&t=1s