P.E.S COLLEGE OF ENGINEERING,

MANDYA

(An Autonomous Institute under Visvesvaraya Technological University, Belagavi)

A DISSERTATION REPORT ON

"TRAFFIC ANALYSIS USING VIDEO PROCESSING"

Submitted in partial fulfilment of the requirement

For the award of the

BACHELOR OF ENGINEERING DEGREE

Submitted by

Ashwini C Jadhav	**4PS18EC012**
Bhoomika H J	**4PS18EC019**
Shamanth Kumar H P	**4PS18EC095**
Sneha R	**4PS18EC099**

Under the guidance of

Prof. Kumar. N. Krishna Murthy,
Assistant Professor

Department of Electronics and Communication Engineering
P.E.S. College of Engineering, Mandya.

2021-2022

P.E.S COLLEGE OF ENGINEERING

MANDYA-571401

(An Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that,

Ashwini C Jadhav	**4PS18EC012**
Bhoomika H J	**4PS18EC019**
Shamanth Kumar H P	**4PS18EC095**
Sneha R	**4PS18EC099**

have successfully completed the project work entitled **"Traffic Analysis Using Video Processing"** in partial fulfilment for the award of degree of **Bachelor of Engineering in Electronics and Communication Engineering** of **P.E.S College of Engineering, Mandya, VTU Belagavi** during the year **2021-2022**. It is certified that all corrections/suggestions indicated in internal assessment have been incorporated in the report deposited in the Library. The project has been approved as it satisfies the academic requirements in respect of project work prescribed for the degree **in Bachelor of Engineering.**

Signature of the Guide
Prof. KUMAR. N. KRISHNA MURTHY,
Assistant Professor

Signature of the HOD
Dr. M.J ANAND,
Associate Professor & HOD

Dr. R. M. MAHALINGE GOWDA (Principal)
PES College of Engineering, Mandya

Project work viva-voice examination			
Sl.No	Examiners		Date
	Name	Signature	
1			
2			

P.E.S COLLEGE OF ENGINEERING

(An Autonomous Institute under VTU, Belagavi)

MANDYA – 571401

Department of Electronics and Communication Engineering

DECLARATION

We **Ashwini C Jadhav, Bhoomika H J, Shamanth Kumar H P, Sneha R**

students of 8th semester Bachelor of Engineering in Electronics & Communication, PESCE, Mandya, hereby declare that the project work being presented in the dissertation entitled **"Traffic Analysis Using Video Processing "** is an authentic record of the work that has been independently carried out by us and submitted in partial fulfilment of the requirements for the award of degree **Bachelor of Engineering in Electronics & Communication,** affiliated to **Visvesvaraya Technological University (VTU), Belagavi** during the year 2021- 2022.

The work contained in the thesis has not been submitted in part or full to any other university or institution or professional body for the award of any other degree or any fellowship.

Place: Mandya

Date: 11/07/2022

Ashwini C Jadhav

Bhoomika H J

Shamanth Kumar H P

Sneha R

ACKNOWLEDGEMENT

It gives immense satisfaction and great pride to present our project report on **"Traffic Analysis Using Video Processing".** The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without mentioning the people who made it possible. So, with deep gratitude we acknowledge all those guidance and encouragement served as bacon light and crowned my efforts with success.

With due respect, we sincerely thank our guide Prof. Kumar. N. Krishna Murthy, Assistant Professor, Department of Electronics and Communication Engineering, P.E.S. College of Engineering, Mandya, for his encouragement and support during project work.

We have immense pleasure in expressing our heartfelt gratitude to Dr. M J Anand, Associate Professor and H.O.D, Department of Electronics and Communication Engineering, P.E.S. College of Engineering, Mandya, for the valuable guidance and encouragement lent by him through and also all the staff members who extended the preparatory steps of this Project.

We express sincere thanks to Dr. R.M. Mahalinge Gowda, Principal, P.E.S. College of Engineering, Mandya for providing excellent and technically sound environment. We also would like to thank our parents and friends for their support and motivation.

We are extremely grateful to Ms. Geetha M S, DCP, Crime and Traffic, Mirza road, Nazarbad, Mysore, for sharing the traffic data for our project execution and their precious time.

Lastly, we express our sincere gratitude to all those who directly or indirectly helped throughout this project work.

Ashwini C Jadav

Bhoomika H J

Shamanth Kumar H P

Sneha R

ABSTRACT

Growing population density because of urbanization and immigrants from rural region is the cause for traffic congestion. This will cause them to reach their destination after the hour of need. In order to avoid this hustle and to make better choice in selection of road path, there is need of engineering solution to rectify the problem. Meanwhile, transportation is inevitable, problem can be overcome by better traffic management and traffic analysis. Traffic analysis is the analysis of number of vehicles in particular place for specific interval of time.

One of the solutions for the above stated problem is smart traffic control system. Using a surveillance camera to calculate density status in that defined place and time, video processing technique is incorporated. This density counting algorithm is realized by the comparison between one frame of the video and the reference image followed by looking for the vehicles in the desired region. The RGB image is the input data which will be converted to grey level images. Using video processing technique, the vehicles movement is detected and density status is stored in database. This database is updated every interval of time in each day and this implementation is performed using python programming.

This everyday updated dataset is processed in order to find average traffic density with respect to time which will in turn generate traffic intensity chart of specified region of given day. This predicted result can be accessed by users through a webpage.

Table of Content:

List of Figures:

List of Tables:

CHAPTER-1

INTRODUCTION

1.1 PROBLEM STATEMENT

Increase in population and vehicle density causes traffic overcrowding which further leads to many problems, namely

- Excess fuel consumption.
- Increase in travel time and cost.
- Degrades air quality due to vehicle emission.
- Road accidents.
- Blockage of emergency vehicles.

These problems can lead to Global warming. In some scenarios when outsiders visit urban areas, they will get stuck in between roads. There is a need of showing them less traffic path in prior. By this local people and outsiders can plan their route and avoid getting into traffic congestion.

1.2 AIM

The aim of the project is to analyze the pattern of traffic for over a period of time of a road and give the traffic intensity chart of given day with respect to the time requested by the user.

1.3 GENERAL INTRODUCTION

Our proposed idea is to analyze traffic and to avoid problems like traffic jam, wastage of time and road accidents. The efficient management and monitoring of vehicle traffic is most important as increase in number of vehicles on road. The aim of the project is to analyze the pattern of traffic for over a period of time and give out the result as traffic chart of particular day requested by the user.

Instead of implementing tedious solutions like building new infrastructures like highways and freeways or widening the existing roads, we can deploy Intelligent Transportation Systems (ITS). Initial step to combat traffic problems through smart way is by traffic data collection process using installed instruments like inductive loop detectors, surveillance cameras and microwave detectors.

Cameras are typically installed at the high-risk potential spots of roads to provide appropriate data for traffic flow monitoring, vehicle speed measurement, fast detection of incidents, automatic detection of dangerous traffic situations, and so on. A key point is that the output of these cameras is stored as video sequences which provides the advantage of traffic data analysis using computer vision techniques like video processing.

Images from these videos are extracted as frames in a second and processed which gives us density of frames. This density status is cross checked with movement detection part and then traffic intensity level is finalized. This will be updated to the database. Data manipulation is made for the given road and day. Traffic chart is thus generated which in turn can be accessed by the users through a webpage.

1.4 MAIN SYSTEM SOFTWARE

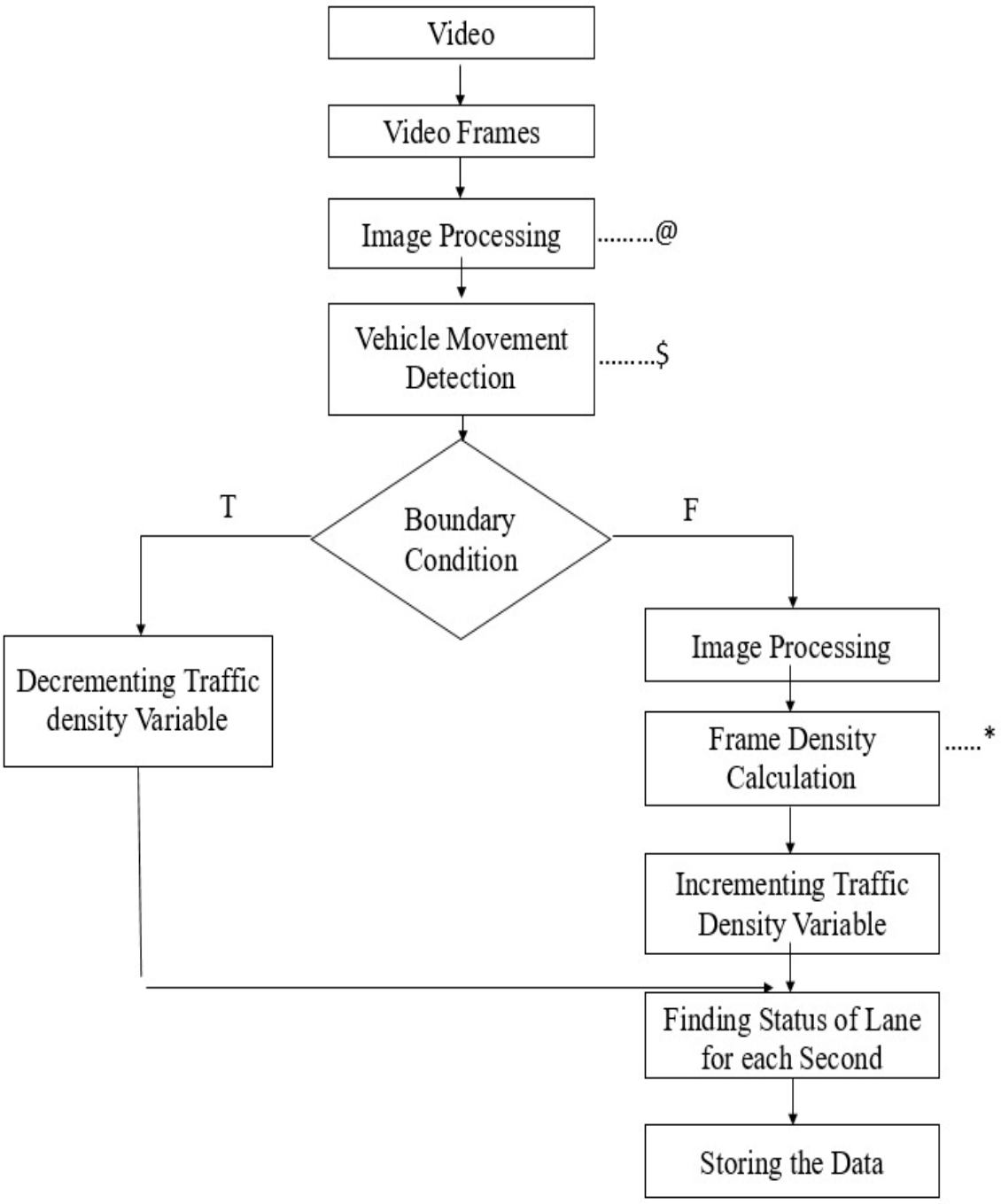


Fig 1.1 Flow Diagram of Main System Software

1.5 APPLICATIONS

- Planning of Road and Infrastructure:

Careful planning and evaluation of the roads and infrastructure to meet the future traffic volume is necessary. Automatic traffic measurements improve the prediction capacity of any traffic model as larger time periods can be captured. The traffic analysis can be conducted on highways using security cameras. The system accurately detects vehicles such as cars or trucks moving in both directions on the highway.

- Intelligent Traffic Systems (ITS) and Smart-City Applications:

Intelligent Traffic Systems (ITS) improve traffic with regard to safety, comfort and emission. Intelligent Traffic Systems consist of sensing, logic and action components. An example could be a network of traffic cameras, a server and traffic lights. In such scenario cameras are the sensors used to perceive the system. Image processing and traffic optimization software runs on the server and drives the traffic lights.

- Real-time Traffic Control:

Cameras mounted on traffic lights provide information on the queue length, the traffic velocity and vehicle categories. Depending on the traffic a smart algorithm controls the traffic light to optimize throughput and reduce congestion. Having such a system installed on multiple crossroads the system can be further optimized.

1.6 CHALLENGES

Current video-based systems are sensitive to operating environment conditions like weather or illumination which may result in less accuracy and reliability. In this section, the main challenges of vehicle detection are discussed:

- Camera Motions:

When there is a motion in camera's field of view, like the videos captured by unstable or vibrating cameras, processing the video will be a challenging issue. The result of this phenomenon is usually represented as motion blur in video scene which affects both detection and tracking steps harder. Motion blur may be avoided by temporal de-blurring or estimating a single motion blur kernel for the entire image. The reason of this phenomenon is environmental causes like winds.

- Low Light Conditions:

In nights and other low light situations like tunnels, cars can't be simply detected by their visual features and the only part of them that can be recognized are headlight/taillight lamps. So other parts that do not produce light or exposed to light from a very close distance will appear to have dark color. This lack of visual features can affect detection and tracking processes. The other challenge here is to pair detected lamps to be considered as an individual car. Image binary conversion using an adequate threshold value can deal with such situations. At night videos, vehicle headlights/taillight and bad illumination that may cause strong noise can also lead to many difficulties for the detection task.

- Object Motion Speed:

The speed of the moving vehicles plays an important role in the detection task. If the object is moving very slowly, the temporal differencing method will fail to detect the portions of the object preserving uniform region. On the other hand, a very fast-moving vehicle leaves a trail of ghost region behind it in the foreground scene. Intermittent motions of objects cause ghosting artifacts in the detected motion.

- Vehicle Shadows:

In sunny weather situations cast shadows always accompany the moving vehicles. These shadows can be regarded as a part of the vehicle and may affect the detection task. In such circumstances, casting shadow elimination or using edge detection to separate shadow from vehicles are two common approaches

which can result in better accuracy in vehicle detection. Shadows cast by foreground objects often complicate further processing steps subsequent to background subtraction.

1.7 MOTIVATION

The Motivation of this project is to focus on the issues related with traffic density. As vehicle population increases, ITS (Intelligent Transportation Systems) becomes more significant and mandatory in today's overpopulated world. Vital problems in transportation such as mobility and safety of transportation are considered more, especially in metropolitans. ITS mainly aims to obtain safer traffic conditions, comfort in transportation, and to increase the road-traffic efficiency by improving the functionality of cars and roads. For this purpose, information systems for locations, and warning systems for vehicle safety have previously been implemented to enhance driver's ability to sense the surrounding environment. ITS applications can be applied to different areas such as traffic infrastructure management and intelligent vehicles. In this thesis, vision-based ITS application is considered for road traffic monitoring using static cameras. In the traffic analysis application, the video cameras are placed on posts above the ground to obtain complete view of the road and passing vehicles. Road traffic monitoring aims at the acquisition and analysis of traffic density.

1.8 OBJECTIVE

1. Prediction of congestion and help user to avoid congestion and save time.
2. To display traffic chart and level of congestion on a user interface.

A GUI is prepared for the user to input the date and location of which he needs the traffic density. The back-end compares the input date and location with that of the database and gives the result.

- The result consists of the traffic density of that particular road at the input date.
- The output web page consists of traffic intensity graph (bar graph) and a small map of that location. The graph has time stamps at 1 hour interval in x-axis and traffic intensities in y-axis.

CHAPTER 2

LITERATURE SURVEY

In [2], X. Fu, Z. Wang, D. Liang, J. Jiang proposed the extraction of object in motion in real-time video sequence. In this paper they explained the conversation of RGB color image to Gray level image it gives more acceptable results in video processing. This paper presents an efficient moving object segmentation algorithm for real-time web-based multimedia communication systems. To extract the moving objects regions from the sequence, the foreground of moving objects isn't used directly, but the background image is obtained firstly, based on motion information. The algorithm proposed in this paper is motivated by the assumption that moving objects usually have intensity changes between successive frames. First, obtaining background mask technique is used to construct a reliable background image from the accumulated frame difference information between successive frames in a sequence. The moving object region is then separated from the background region by comparing the current frame with the constructed background image. Then, a post-processing using morphological filter operation is used to remove noise regions and produce a smoother shape boundary.

In [3], P. Choudekar, S. Banerjee, M. K. Muju presents a methodology that improves the algorithms based on adaptive background mixture models by using more intelligent updating equations which allows it to gain speed and more accuracy, as well as to adapt more effectively to changing environment. As the problem of urban traffic congestion spreads, there is a pressing need for the introduction of advanced technology and equipment to improve the state-of-the-art of traffic control. Traffic problems nowadays are increasing because of the growing number of vehicles and the limited resources provided by current infrastructures. The simplest way for controlling a traffic light uses timer for each phase. Another way is to use electronic sensors in order to detect vehicles, and produce signal that cycles. This paper proposes a system for controlling the traffic light by image processing. The system will detect vehicles through images instead of using electronic sensors embedded in the pavement. A camera will be installed alongside the traffic light. It will capture image sequences. The image sequence will then be analyzed using digital image processing for vehicle detection, and according to traffic conditions on the road traffic light can be controlled.

In [4], Dr Ravindra Kumar and Reha Justin proposed a system of vehicle counting and detection using image processing methods. In this paper, they have used edge detection algorithm and Kalman filter to detect and track the vehicle. Vehicle counting process provides appropriate information about traffic flow, vehicle crash occurrences and traffic peak times in roadways. An acceptable technique to achieve these goals is using digital image processing methods on roadway camera video outputs. This paper presents a vehicle counter classifier based on a combination of different video image processing methods including object detection, edge detection, frame differentiation and the Kalman filter. An implementation of proposed technique in this paper has been performed using python programming language. It describes the methodology used for image processing for traffic flow counting and classification using different library and algorithm with real time image.

In [5], M. Seki, H. Fujiwara, and K. Sumi presented a novel method for detecting and segmenting foreground objects from video, with both stationary and moving background objects and subject to gradual or sudden changes is presented. Background subtraction is a useful and effective method for detecting moving objects in video images. The proposed method of the paper assumes that image variations are caused only by moving objects (i.e., the background scene is assumed to be stationary), however, its applicability is limited. This paper proposes a background subtraction method that robustly handles various changes in the background. The method learns the chronological changes in the observed scene's background in terms of distributions of image vectors. The proposed method of paper expresses actual changes in the background using a multi-dimensional image vector space. This enables the method to detect objects with the correct sensitivity.

In [6], M. Lei, D. Lefloch, P. Gouton, K. Madani have addressed urban area problems, which proposes an enhanced version of a sigma-delta background estimation method, which is optimized for urban traffic scenes that are frequently affected by vehicles moving slowly or temporarily stopped. This paper presents a video-based solution for real time vehicle detection and counting system, using a surveillance camera mounted on a relatively high place to acquire the traffic video stream. The two main methods applied in this system are: the adaptive background estimation and the Gaussian shadow elimination. The former allows a robust moving detection especially in complex scenes. The latter is based on color space HSV, which is able to deal with different size and intensity shadows. After these two

operations, it obtains an image with moving vehicle extracted, and then operation counting is affected by a method called virtual detector.

In [7], Dr. Stalin Alex and Dr. Amitabh Wahi proposed a BSFD(Background Subtraction Frame Difference) algorithm for moving object detection. Advantages and drawbacks of two common algorithms often employed in the moving target detection, background subtraction technique and frame distinction methodology are analyzed and compared in this paper. Then supported the background subtraction methodology, a BFSD target detection rule is projected. The background image used to process the next frame image is generated through superposition of the current frame image and therefore the current background image with certain chance. The algorithm makes the objects that keep long-standing, however not be detected as a part of the background.

In [8], Tanvi Sable, Nehal Parate, Dharini Nadkar, Swapnil Shinde proposed a Density and Time-based Traffic Control System using Video Processing. The idea is to control the traffic by deciding the traffic density on each roadside and control the traffic signal smartly by utilizing the density data. In this paper, an automated system based on processing of real time videos is proposed for detection of vehicles and recording count of them. The System will consist of various stages which includes Object Car Detection and Signal variation based on density. Captured video will be converted into frames and which will be pre-processed for object detection then detected object count will be used to obtain the density and manipulate the signal accordingly. The density count algorithm works by contrasting the ongoing edge of live video by the reference picture and via looking through vehicles just in the district of intrigue (for example street region).

In [9], RGB to gray scale conversion and Gaussian filtering concepts have been given in detail by Rafael C. Gonzalez and Richard E. Woods. Gray levels represent the interval amount of quantization in gray scale image processing. At present, the most commonly used storage method is 8-bit storage. There are 256 gray levels in an 8-bit gray scale image, and the intensity of each pixel can have from 0 to 255, with 0 being black and 255 being white we. Another commonly used storage method is 1-bit storage. There are two gray levels, with 0 being black and 1 being white a binary image, which, is frequently used in

medical images, is being referred to as binary image. In grayscale images, we can easily objectify the algorithm since the 3D image of objects are created by only 2 spatial dimensions and one brightness dimension. On the other hand, in RGB, HSI, and other color spectrums, this visualization is way harder since there are extra dimensions that can't be easily visualized by the human brain and grey scale images are easier to be operated on. To find edges based on brightness and coloring, it is lot of work ahead. The complexity of the code will substantially increase and will require additional support, debugging and more. Doing the same in grayscale is much easier. Modern computers can do parallel programming and process a megapixel image in just milliseconds. More complicated tasks like facial recognition, OCR, resizing, etc… will obviously take much longer. Whatever time is required to process the image or just get some information out of it, most users want to be done as fast as possible. Now if we are working on a single image and we process it in a triple channel such as RGB, it takes about 3 times longer than in Grayscale.

A Gaussian Filter is a low pass filter used for reducing noise (high frequency components) and blurring regions of an image. The filter is implemented as an Odd sized Symmetric Kernel (DIP version of a Matrix) which is passed through each pixel of the Region of Interest to get the desired effect [1]. The kernel is not hard towards drastic color changed (edges) due to it the pixels towards the centre of the kernel having more weightage towards the final value then the periphery. A Gaussian Filter could be considered as an approximation of the Gaussian Function. A large number of wavelet-based methods are available to achieve a good noise reduction, while preserving the significant image details. The wavelet denoising procedure usually consists of shrinking the wavelet coefficients. Shrinkage estimators can also result from a Bayesian approach. The drawback of wavelet-based methods is that the process is complex and consequently the time consumption is very high.

Thresholding, morphological transformation techniques have been explained along with the context of mathematical morphology as a tool for extracting image components that are useful in the representation and description. We are interested also in morphological techniques for pre- or postprocessing, such as morphological filtering, thinning, and pruning. The basic thresholding technique is Binary Thresholding. For every pixel, the same threshold value is applied. If the pixel value is smaller than the threshold, it is set to 0, otherwise, it is set to a maximum value. Morphological transformations are some simple operations based on the image shape. It is normally performed on binary images.

It needs two inputs, one is our original image, second one is called structuring element or kernel which decides the nature of operation. Two basic morphological operators are

Erosion and Dilation. Then its variant forms like Opening, Closing, Gradient etc also comes into play. Closing operation is useful in closing small holes inside the foreground objects, or small black points on the object. It also tends to smooth sections of contours, it generally fuses narrow breaks and long thin gulfs, eliminates small holes and fills gaps in the contour. The idea of the morphological filters is to shrink and let grow process. The word “shrink” means using median filter to round off the large structures and to remove the small structures and in grow process, remaining structures will grow back by the same amount. The morphological concepts involve an image and a structuring element. Morphological transformation called morphological reconstruction that involves two images and a structuring element. One image, the marker, contains the starting points for the transformation. The other image, the mask, constrains the transformation. The structuring element is used to define connectivity.

CHAPTER 3

SYSTEM REQUIREMENT

The system along with data access is divided into three domains

1. Video Processing
2. Data storage and extraction
3. Web Development

3.1 SYSTEM SPECIFICATION

1. Video Processing:

   Hardware requirements:

   i. Stationary Camera or Videos:
      Frame rate: 25-30 frames/second
      Video duration: 17 hours/day
   ii. Hard disk: Minimum 128 GB for 7 days video

   Software requirements:

   i. Operating System: Windows
   ii. Language: Python
   iii. Module: OpenCV2
   iv. IDE: Spyder, VS code

2. Data storage and extraction:

   Hardware requirements

   i. Storage space: 1MB (for CSV file)

   Software requirements

   i. Application: Microsoft excel
   ii. Language: Python
   iii. Charset: UTF-8
   iv. Module: csv, numpy, pandas

3. Web development:

   Software requirements

   i. Server: Python
   ii. Language: HTML5, CSS3, JavaScript, Python
   iii. Framework: Flask
   iv. Modules: matplotlib, calender, numpy, os

- Spyder: Spyder is an open-source cross-platform integrated development environment (IDE) for scientific programming in the Python language. It is written completely in Python. It is designed by scientists and is exclusively for scientists, data analysts, and engineers. It is also known as the Scientific Python Development IDE and has a huge set of remarkable features

- Python: Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems. It is very popular, mainly because of its simplicity and code readability. Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis.

- VS Code: Visual Studio Code is an open-source, cross-platform source code editor that has become famous, particularly in the web development community. It's fast, extensible, customizable, and has tons of features. It has many advantages over other IDE. It used for full stack web development.

- Pandas: Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python.

- 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.

- Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is a plotting library used for 2D graphics in python programming language. It can be used in python scripts, shell, web application servers and other graphical user interface toolkits

- Flask: Flask is a small and lightweight Python web framework that provides useful tools and features that make creating web applications in Python easier. It gives developers flexibility and is a more accessible framework for new developers since you can build a web application quickly using only a single Python file.

- OpenCV2: OpenCV2 (Open-Source Computer Vision Library) is an open-source library that includes several hundreds of computer vision algorithms. OpenCV2 has a modular structure, which means that the package includes several shared or static libraries. The following modules are available:
  - Video - a video analysis module that includes motion estimation, background subtraction, and object tracking algorithms.
  - Image processing - an image processing module that includes linear and non-linear image filtering, geometrical image transformations (resize, affine and perspective warping, generic table-based remapping), color space conversion, histograms, and so on.
  - objdetect - detection of objects and instances of the predefined classes (for example, faces, eyes, mugs, people, cars, and so on).

3.2 NON-FUNCTIONAL REQUIREMENT

a) Performance requirement: Updates must be frequently done in database.

b) Space requirement: If the system is directly taking input video footage, then additional space is not needed. If the system is taking already recorded video, then it should be able to handle large amount of data depending on camera specification.

c) Reliability requirement: The system should work under normal conditions and should perform the required functions successfully.

d) System availability: The proposed system can be made as an automated process, so it can stay open for certain number of hours a day. It should provide good performance.

e) Server availability: Deployed server should be made available 24*7 irrespective of active users.

f) Security Requirements: Users can just view information provided in webpage. Different access levels will be implemented so that some actions are prohibited to some users. Admin will have access to configure the system.

CHAPTER 4

SYSTEM DESIGN

4.1 DETAIL DESIGN

The system consists of three parts:

1. Video processing
2. Vehicle movement detection
3. Frame density calculation

4.1.1 Video processing

It is required to find status of the lane by considering both movement detection and density calculation of frames. In both task, similar video processing technique is applied. The difference is consideration of frame in both the tasks. In movement detection, consecutive frames are taken and in density calculation, current frame and background image is taken. Frames are extracted one by one. The function used here is,

Syntax: cv2.VideoCapture (String or index)

Parameters: String represents a path to open video file or image file sequence and index represents for capturing device

Image subtraction and processing

Basic steps followed in image processing is given below:

1. Downsampling each frame and background image.
2. Normalization of each frame and background image.
3. Apply frame differencing technique on every pair of images
4. Convert output image from RGB model to Grayscale model
5. Apply gaussian blur on the output image of the previous step
6. Apply image thresholding on the output image
7. Perform morphological operation-closing on the output image

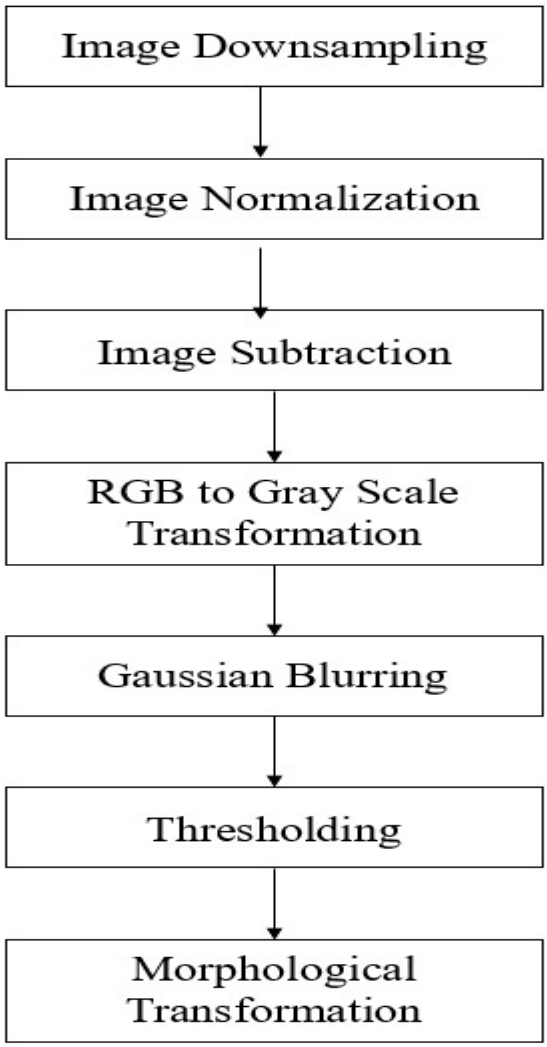


Fig 4.1 Flow Diagram of Image Processing

Downsampling:

To downsample an image, opencv provides built-in function. Image is reduced to one fourth of the size.

Syntax: cv2.pyrDown (image)

Parameter: An image is passed as an input to the function.

Normalization:

In image processing, normalization is done to alter the intensity of pixels of any given image. The normalize () function is used to perform image normalization using the OpenCV library in Python.

Syntax: cv2.normalize(src1, alpha, beta, norm_type=cv2.NORM_MINMAX)

Parameters:

- src1: The source image which needs to be processed
- A range of values needs to be specified using alpha and beta. The pixel values range between these two values.
- NORM_MINMAX: This ensures that the alpha value is taken as the starting point and the beta value as the maximum value.

Frame differencing:

To calculate absolute difference between two arrays when they have the same size and type.

Syntax**:** cv2.absdiff (src1, src2, dst)

Parameters:

- src1: first input array
- src2: second input array
- dst: output array that has the same size and type as input arrays.

RGB to Grayscale transformation:

To convert RGB color space to Grayscale color space.

cv2.cvtColor() method is used to convert an image from one color space to another.

Syntax: cv2.cvtColor(src, code)

Parameters:

- src: It is the image whose color space is to be changed.
- code: It is the color space conversion code, cv2.COLOR_BGR2GRAY is considered for transformation.

Gaussian blurring:

This blurs an image using a Gaussian filter. The Gaussian filter is a low-pass filter that removes the high-frequency components. It is highly effective in removing Gaussian noise from an image.

Syntax: GaussianBlur(src, ksize, sigmaX)

Parameter:

- src: A Mat object representing the source (input image) for this operation.
- ksize: A Size object representing the size of the kernel.

- sigmaX: A variable of the type double representing the Gaussian kernel standard deviation in X direction.

Thresholding:

The basic Thresholding technique is Binary Thresholding. For every pixel, the same threshold value is applied. If the pixel value is smaller than the threshold, it is set to 0, otherwise, it is set to a maximum value.

Syntax: cv2.threshold(source, thresholdValue, maxVal, thresholdingTechnique)

Parameters:

- source: Input Image array (must be in Grayscale).
- thresholdValue: Value of Threshold below and above which pixel values will change accordingly.
- maxVal: Maximum value that can be assigned to a pixel.
- thresholdingTechnique: The type of thresholding to be applied.

Morphological transformation:

Closing involves dilation followed by erosion in the outer surface (the foreground) of the image. It is useful in closing small holes inside the foreground objects, or small black points on the object.

Syntax: closing = cv2.morphologyEx(src, cv2.MORPH_CLOSE, kernel)

Parameters:

- src: It is the input image.
- cv2.MORPH_CLOSE: erode (dilate (src, kernel))
- kernel: Structuring element used for Closing.

4.1.2 VEHICLE MOVEMENT DETECTION

To identify movement of vehicles, the binarized image obtained from image processing stage is passed to the contour step to define the contour for the detected objects.

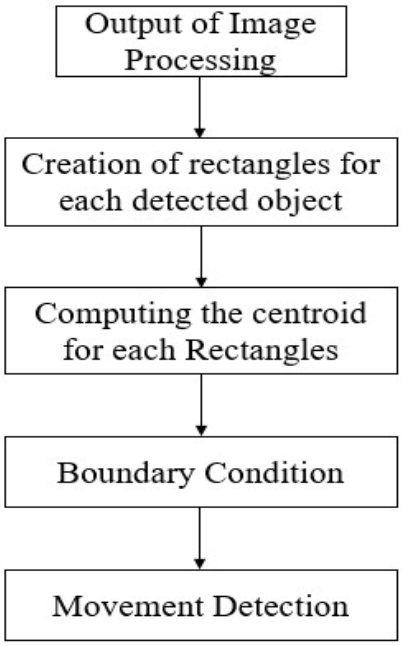


Fig 4.2 Vehicle movement detection (continued from $)

Finding contour:

The contours are a useful tool for shape analysis and object detection and recognition.

Syntax: cv2.findContours(src, contour_retrieval, contours_approximation)

Parameters:

- src: 2-dimensional binary images are preferred.
- contour_retrieval: cv2.RETR_TREE is passed as second argument It retrieves all the contours.
- contours_approximation: cv2.CHAIN_APPROX_SIMPLE is passed as third argument. Since we do not require all the co-ordinates of each object, it removes all redundant points and compresses the contour, thereby saving memory. For a rectangular box only 4 points is taken into account.

Computing centroid for contours:

The centroid is the centre point of the object. With the help of co-ordinates, centroid of a contour is calculated. Centroid of rectangle is defined as the center point where all the diagonals intersect each other.

Boundary condition:

If the movement of vehicle is from top to bottom then the imaginary line will be parallel to x-axis. Offset is considered for boundary condition.

If (x, y) is the co-ordinate of a centroid of an object, then for object movement detection, it should satisfy the condition, if y < (line_height+offset) and y > (line_height-offset).

If the movement of vehicles is inclined then the imaginary line will have the slope. Boundary condition can be given as,

slope=abs((line_height–length_of_matrix)/ (width_of_matrix - 0)) ……..... (4.1)

is derived from the slope of a straight line between two points says (x1, y1) and (x2, y2),

slope = (y2 –y1)/ (x2 – x1) ……..…… (4.2)

Decrementing traffic density variable:

Priorly defined traffic density variable is rapidly reduced if any movement of vehicles is detected in any frame. As more movement of vehicles is there, traffic density variable value is also rapidly reduced.

4.1.3 FRAME DENSITY CALCULATION

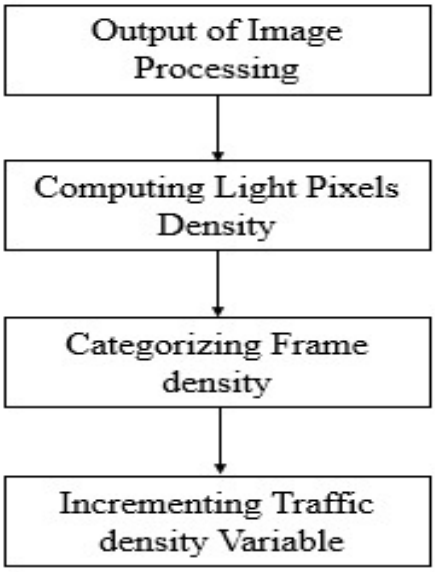


Fig 4.3 Frame density calculation (continued from *)

Frame density is calculated by using frame differencing algorithm. The difference between current frame and reference background image is found. The output of image processing is further used to calculate frame density.

Computing light pixels density:

The processed image in matrix form has bi-level values, where 0 denotes dark pixel and 255 denotes light pixel. By this density of the frame is found.

Syntax: count_nonzero (x)

Parameters: Counts the number of non-zero values in the array x.

Formula to find density in percentage:

D1= (numpyOnes / dimension) *100……..…… (4.3)

numpyOnes represents number of light pixels and dimension of the frame represents product of size attributes of frame.

Incrementing traffic density variable:

Priorly defined traffic density variable value is incremented which is proportional to density status. Although it is possible set ranges for frame density(D1), it can be classified as high, moderate or low density and each will have different incrementing value.

Finding status of the lane for each second:

After processing frames for each second, based on the value of traffic density variable status of the lane is decided. Traffic Density variable will hold the value that falls under a set of ranges. These ranges are meant for very high, high, moderate, low, very low traffic density of the lane.

Storing the data:

As data can be stored in various forms, comma-separated values type of file is used to here to store the results. Proper file handling has to be done. The key function for working with files in Python is the open () function.

The open () function takes two parameters; filename, and mode.

There are four different methods (modes) for opening a file:

"r" - Read - Default value. Opens a file for reading, error if the file does not exist

"a" - Append - Opens a file for appending, creates the file if it does not exist

"w" - Write - Opens a file for writing, creates the file if it does not exist

"x" - Create - Creates the specified file, returns an error if the file exists

It has 5 columns,

1. Date
2. Time
3. Day

4. Intensity
5. Comment

4.2 USE CASE DIAGRAM

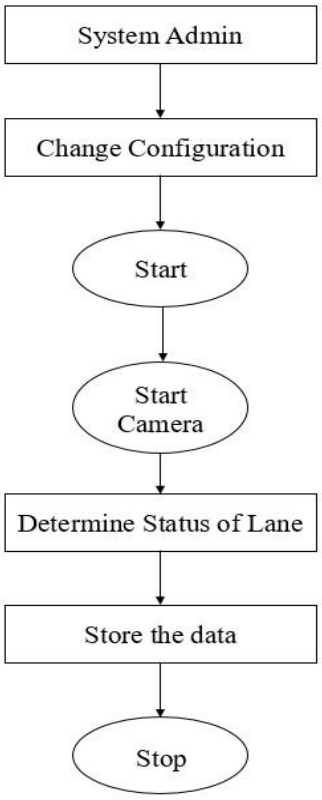


Fig 4.4 Use Case Diagram of the system

If the system is implemented, it has to be configured as per camera specifications and placement of camera. The area covered by camera will not always be favorable, it will have unwanted regions that are being captured. By image cropping it is possible to acquire region of interest. Parameters for this kind of configuration has to be explicitly considered.

4.3 USER INTERFACE DESIGN

A responsive webpage is designed with the help of web technologies along with Flask framework. Web technologies involves HTML5 for building elements of webpage, CSS3 for styling the webpage and JavaScript for validation of inputs entered by users.

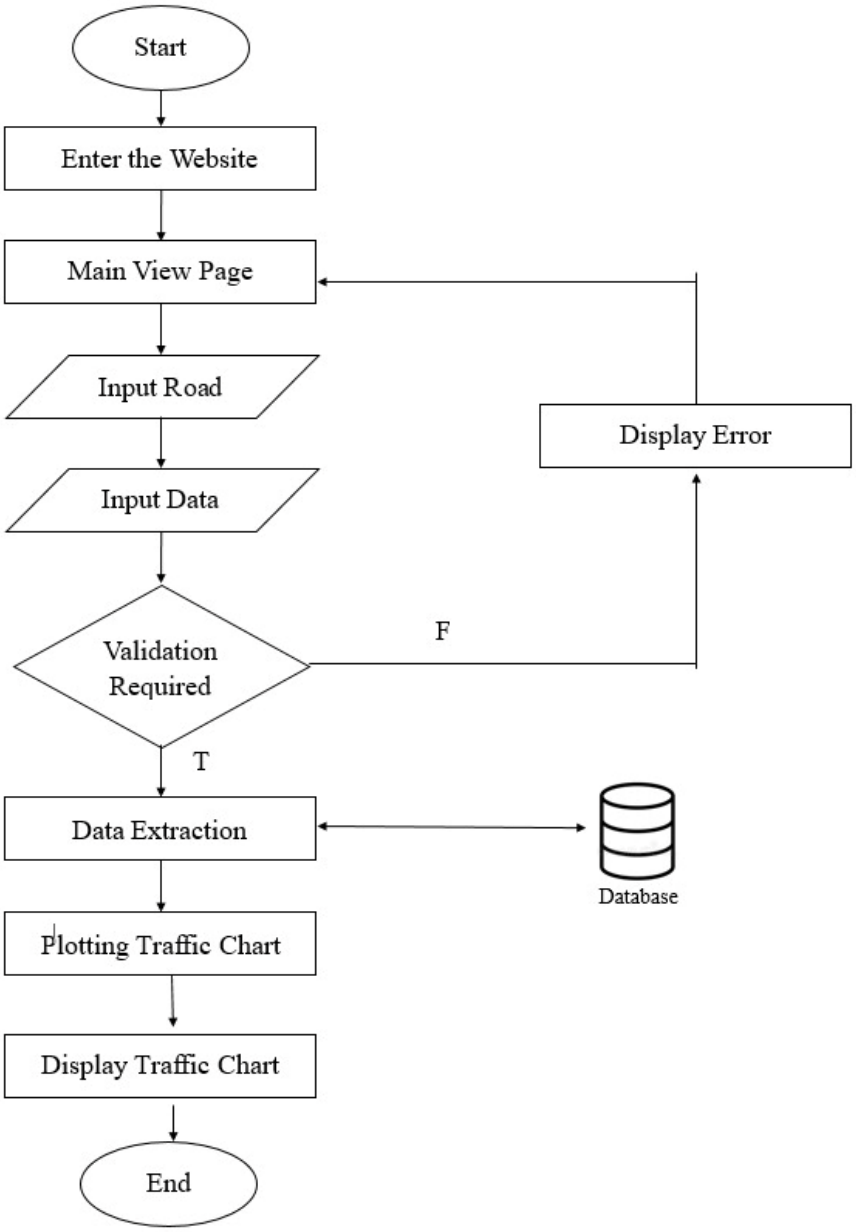


Fig 4.5 Flow diagram of GUI

The website has components:

1. HTML pages
2. Styling of webpage by CSS
3. Validation of inputs by JavaScript
4. Data extraction by pandas
5. Plotting traffic chart by matplotlib

HTML Page-1:

The website consists of two pages. In first page it has a form. The form method used is post method. It consists of two input fields for selection of Road and Date. A dropdown type of selection is provided to select desired road. Each road is assigned with unique values which

will be passed to backend. In date field, entered date will be passed as it is to backend and day should be found out from this raw data.

Styling of webpage by CSS:

With the help of CSS, responsive webpage is designed. Responsive web design is about creating web pages that look good on all devices. A responsive web design will automatically adjust for different screen sizes and viewports.

By inserting meta tag, ie., <meta name="viewport" content="width=decive-width, initial-scale=1.0"> responsive web design is made. Responsive text sign can also be achieved. The text size can be set with a "vw" unit, which means the "viewport width". That way the text size will follow the size of the browser window: For images attributes, it can set with percentage unit.

Validation of inputs by JavaScript:

Each field is assigned with a unique ID. By this ID, it is possible to get inner elements of the division block. JavaScript helps to get those values instantly even before passing values to backend. This helps to perform validation at frontend itself. If the input value doesn't satisfy certain conditions then, then an error message is displayed.

Syntax to get form values: document.forms['form']['parameter1']

Retrieves a collection, in source order, of all form objects in the document.

Syntax to get elements of a division by its ID: document.getElementById ('ID1')

Returns a reference to the first object with the specified value of the ID attribute.

Data extraction by pandas:

This form values are passed to backend method where flask model comes into picture. Name of week is found out by taking date which was entered by user. Based on chosen road, specific file is read. There will be many csv files which is stored in system. Path to each file is priorly mentioned.

Syntax: pandas.read_csv(filepath_or_buffer)

This reads a comma-separated values (csv) file into DataFrame.

DataFrame has five columns- Date, Day, Time, Intensity and Comment. Date and Comment columns will be dropped from Data Frame.

Syntax: DataFrame.drop([labels], axis=1, inplace=True)

Parameters:

- labels: single label or list-like. Index or column labels to drop.
- axis: {0 or 'index', 1 or 'columns'}, default 0; Whether to drop labels from the index (0 or 'index') or columns (1 or 'columns').
- inplace: bool, default False; If False, return a copy. Otherwise, do operation inplace and return None.

It removes rows or columns by specifying label names and corresponding axis, or by specifying directly index or column names. When using a multi-index, labels on different levels can be removed by specifying the level.

The DataFrame will have unwanted days data. Traffic intensity data of the specified day has to be taken.

Syntax: DataFrame.loc[DataFrame ['Day'] == day]

It is generally for accessing a group of rows and columns by label(s).

The DataFrame will now have traffic intensity data of a particular day but this will be not in order. In order to group based on time to further process, groupby method is used.

Syntax: DataFrame.groupby(['Time'])

A groupby operation involves some combination of splitting the object, applying a function, and combining the results. This can be used to group large amounts of data and compute operations on these groups.

Final traffic chart is prepared by taking mean of all intensity data with respect to time. More the data, more accurate will be the result. It is achieved by applying inbuilt method of pandas library.

Syntax: DataFrame.mean ()

Plotting traffic chart by matplotlib:

A new list is formed and is passed as y-axis argument which has new intensity levels of particular day and time. Time stamp from is passed as x-axis argument in graph. A bar graph is plotted using matplotlib library. Limits of y-axis is specified and width of bar is also specified. This graph is saved in static folder. The old image which was loaded is deleted every time when a new traffic chart is requested.

Syntax: matplotlib.savefig('static\trafficChart.jpg')

HTML Page-2:

The output is shown through in the second page with an image consists of traffic chart with respect to time and day of the week. By specifying file name of the image and passing it into img tag of html, graph is displayed. It is possible to embed a road map with the help of google maps embed option. This is also displayed along with traffic chart. Link to embed road map must be priorly declared. Special days are priorly declared if in case date entered by user falls on any one of the special days that are declared. A message is shown if it is a special day. By using collections special days are declared. For instance,

```
specialDays = {
   '2022-01-26':'Republic Day',
   '2022-08-15':'Independence Day',
   '2022-08-31':'Ganesh Chaturthi',
   '2022-09-25':'Mahalaya Amavasya',
   '2022-10-26':'Balipadyami / Deepavali',
   '2022-12-25':'Chirstmas Day'
        }
```

Hosting website and file structure:

A separate directory is maintained for rendering web pages. All HTML pages are stored under folder called Templates. Images containing traffic charts are stored under folder called Static. In order to render web pages render_template () function is used. It renders a template from the template folder with the given context.

It can be hosted on local server. Website is made accessible to everyone by hosting it in python server using third party domains.

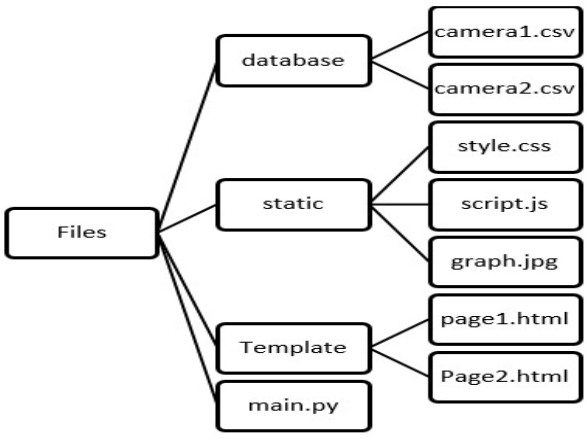


Fig: 4.6 File structure in web deployment

CHAPTER 5

PROPOSED METHOD

5.1 VIDEO PROCESSING

Input can be given directly from any stationary cameras to the system or by providing video clips from stationary cameras. Video is consisting of frames as images which are retrieved as frames in a second. Image processing is performed over these retrieved frames. In order to identify movement of vehicles, frame differencing is done between n^{th} frame and $n\text{-}k^{th}$ frame.

Each frame is compared with the empty road picture also. The difference between empty lane picture and each frame of video will represent density of lane. The output obtained from the image processing algorithm is the density of lane corresponding to vehicles coming from a specific direction. This data is stored in csv file.

5.1.1 Image Acquisition

Considering a live video processing, the system starts with an image acquisition process in which the live video is taken by camera which is mounted on a signal stand and processed. Already recorded video can also be given as input. The function used is cv2.VideoCapture.

5.1.2 Image cropping

The frames from the video are extracted which are processed further. If there is any surrounding building or objects apart from lane it should be removed. It is done by image cropping in which the focus is on region where the vehicles will be present. Cropping helps to obtain the ROI for the system which can help to achieve higher accuracy. It also reduces dimension of the frame thereby increases processing speed.

5.1.3 Image Downsampling:

Normally, for image processing the image used will be of constant size. But in some occasions, it needs to be worked with different resolution. For example, while searching for something in an image, in which size of object is unknown. In that case, a set of the same image with different resolutions is created and then search for object is

done. By doing so, a M×N image becomes M/2×N/2 image. So, area of image reduces to one-fourth of original area.

5.1.4 Image Normalization

Due to change in environmental conditions, there will fluctuations in intensity level or reflection of road. For this reason, normalization is done. By this way, it is possible to consider only one background image for the entire function of proposed system. In image processing, we use normalization to alter the intensity of pixels of any given image. This way, we can control the image contrast, which further helps in image segmentation and feature extraction. It cleans the image from high and low-frequency noise.

5.1.5 Frame Difference Method

Frame differencing is a technique where the computer checks the difference between two video frames. This method is based on forming a precise background image and using it for separating moving objects from their background. The background image is specified either manually, by taking an image without vehicles, or is detected in real-time by identifying the empty frame during video processing and setting it as background. The detection is then achieved by means of subtracting the reference image from the current image. Images are taken as arrays for computation purpose. Absolute difference between two arrays when they have the same size and type. The function used is cv2.absdiff.

$dst(I)=saturate(|src1(I)-src2(I)|)$……. (5.1)

saturate () here is used to convert both the matrices to similar type for the operation to take place.

5.1.6 RGB to grayscale transformation

RGB images contains lot of data and it takes time for processing, to minimize this processing time the RGB color images are converted to gray scale and passed to next stages. The function used is grey = cv2.COLOR_BGR2GRAY. The function converts an input image from one color space to another. Note that the default color format in OpenCV is often referred to as RGB but it is actually BGR (the bytes are reversed).

The equation for rgb to gray scale conversion is given below:

RGB to Gray: 0.299 R + 0.587 G + 0.114 B…….(5.2)

5.1.7 Gaussian blur

In Gaussian Blur operation, the image is convolved with a Gaussian filter instead of the box filter. The Gaussian filter is a low-pass filter that removes the high-frequency components are reduced. Because of its mathematical tractability in both the spatial and frequency domains, Gaussian noise models are used frequently in practice. In fact, this tractability is so convenient that it often results in Gaussian models being used in situations in which they are marginally applicable at best. Gaussian blurring is highly effective in removing Gaussian noise from an image. The function used is cv2.GaussianBlur.

5.1.8 Thresholding

Thresholding is performed in order to obtain presence/absence information of an object in motion. Most techniques work with some blur and threshold, to distinct real movement from noise. Because frame could differ too when light conditions in a room change and camera auto focus, brightness correction etc. This is one of the simplest, but less effective techniques, which operates on still images. It is based on the notion that vehicles are compact objects having different intensity form their background. Thus, by thresholding intensities in small regions we can separate the vehicle from the background. This approach depends heavily on the threshold used, which must be selected appropriately for a certain vehicle and its background. Adaptive thresholding can be used to account for lighting changes, but cannot avoid the false detection of shadows or missed detection of parts of the vehicle with similar intensities as its environment.

If pixel value is greater than a threshold value, it is assigned one value (may be white), else it is assigned another value (may be black). The function used is cv2.threshold.

If $f(x,y) > T$,

Then $f(x,y) = 255$

else $f(x,y) = 0$ where,

$f(x,y)$ = co-ordinate of Pixel Value

T = Threshold Value.

5.1.9 Morphological Transformations

Morphological transformations are some simple operations based on the image shape. It is normally performed on binary images. It needs two inputs, one is our original image, second one is called structuring element or kernel which decides the nature of operation. Two basic morphological operators are Erosion and Dilation.

Closing: Closing is an important operator from the field of mathematical morphology. Like its dual operator opening, it can be derived from the fundamental operations of erosion and dilation. Like those operators it is normally applied to binary images, although there are gray level versions. Closing is similar in some ways to dilation in that it tends to enlarge the boundaries of foreground (bright) regions in an image (and shrink background color holes in such regions), but it is less destructive of the original boundary shape.

As with other morphological operators, the exact operation is determined by a structuring element. The effect of the operator is to preserve background regions that have a similar shape to this structuring element, or that can completely contain the structuring element, while eliminating all other regions of background pixels.

In Closing, dilation is followed by erosion. It is useful in closing small holes inside the foreground objects, or small black points on the object. The function used is cv2.morphologyEx, which performs advanced morphological transformations using an erosion and dilation as basic operations.

5.1.10 Contours and its hierarchy

In order to identify movement of vehicles, the binarized image obtain from closing stage is passed to the contour step to define the contour for the detected objects. Contours can be clarified as basically as a curve joining to every single continuous point having same shade or intensity. The function used is cv2.findContours. The function retrieves contours from the binary image. The contours are a useful tool for shape analysis and object detection and recognition.

Sometimes objects are in different locations. But in some cases, some shapes are inside other shapes. Just like nested figures. In this case, we call outer one as parent and inner one as child. This way, contours in an image has some relationship to each other.

This leads to specify how one contour is connected to each other, like, is it child of some other contour, or is it a parent etc. Representation of this relationship is called the Hierarchy.

5.1.11 Validating contours

There will be some small objects that could appear in frames and these should not be considered as vehicles. Prior minimum dimension of contours is specified. Each contour is checked if its dimension is greater than the specified dimension, then it is considered as valid contours and taken into account.

5.1.12 Movement detection

Identifying vehicles movement is quite challenging, if there is any movement detected it should be identified during video processing. Frame differencing between current frame and previous frame is done. This will capture if there's any vehicle movement. Frame difference interval can also be increased for considerable interval. An imaginary line is drawn to assist movement detection process. If the movement of vehicle is from top to bottom then the imaginary line will be parallel to x-axis. If the movement of vehicles is inclined then the imaginary line will have slope. It can be given as,

slope=abs((line_height–length_of_matrix)/(width_of_matrix - 0))……..…….(5.3)

Contours will be generated in previous stage and by finding the co-ordinates of centroid of contour and checking if it has crossed the imaginary line movement of vehicles can be detected.

prod=int(slope*x+y) ……....…….(5.4)

x and y are the co-ordinates of centroid of the contour.

Value of prod justifies whether there is any movement of vehicles is detected or not.

This is the first parameter for identifying congestion in the lane.

5.1.13 Calculating traffic density

The density of lane is calculated by using frame differencing algorithm. The difference between current frame and reference background image is found. Then RGB to grayscale transformation is done and thresholding is made in order to have bi-level values. It is represented in matrix form, where 0 denotes dark pixel and 255 denotes white pixel. By finding number of white pixels, we will find density of the frame.

The background can change significantly with shadows cast by buildings and clouds, or simply due to changes in lighting conditions. With these changing environmental conditions, the background frame is required to be updated regularly. This can be tackled by two methods. That is, either by regularly updating background image or by performing normalization of frames along with background image. There are several background updating techniques. The most commonly used are averaging and selective updating. In averaging, the background is built gradually by taking the average of the previous background with the current frame. If we form a weighted average between the previous background and the current frame, the background is built through exponential updating.

In selective updating, the background is replaced by the current frame only at regions with no motion detected; where the difference between the current and the

previous frames is smaller than a threshold. Instead of updating selective regions, whole background image is updated. Based on specified criterion, status of lane is identified. This is second parameter for identifying congestion in the lane.

5.2 STATUS OF THE LANE

By considering two parameters Movement detection and Frame density, status of the lane is identified. There are two different approaches while concluding status of the lane. In first method, checking if the road is empty or not during beginning of the process itself and then if any vehicle is present then movement detection is checked. This approach is efficient during low traffic intervals. In second approach, vehicle movement is first checked, if there is no vehicle movement then, density of frame is calculated. If the frame density is high and movement of vehicles is not detected, then it accounts for traffic in the lane. Traffic status is divided into five categories, they are- very low, low, moderate, high, very high.

5.2.1 Identifying density of the lane in one frame

Frame density is calculated for each frame. Based on this density, a variable is incremented over an interval. For example, if density is very high, the variable is incremented by 8; if the density if very low, the variable is kept as it is. A snippet of code is shown which follows proposed algorithm.

```
if density>700:
    pass #error in frame
elif density <7:
    pass #movement detection can be avoided
elif density >=350: #very high
    trafficDensity = trafficDensity +8
elif density >=200: #high
    trafficDensity = trafficDensity +4
elif density >=100: #moderate
    trafficDensity = trafficDensity +2
elif density >=50: #low
    trafficDensity = trafficDensity +1
elif density >=7: #very low
    pass
```

Considering video of 30 frames per second, frame density of all 30 frames is found out and the variable will be holding some value at the end of 30th frame. Background image is also updated here. If there is any movement of vehicle is identified (second parameter) while processing these frames, the value of the variable is decremented.

5.2.2 Identifying status of the lane over an interval

After one second interval, the value of variable is checked with specified condition, for example if it is less than 50, status of the lane is considered as low traffic. Based on these conditions, status of the lane is identified for an interval of one second. A snippet of code is shown below which follows proposed algorithm.

```
if trafficDensity <22:
        t=1
        #print("very low traffic")
    elif trafficDensity <75:
        t=2
        #print("low traffic")
    elif trafficDensity <150:
        t=3
        #print("moderate")
    elif trafficDensity <176:
        t=4
        #print("high traffic")
    else:
        t=5
        #print("very high traffic")
```

For large interval, additional variable is used to store status of the lane which was identified for one second. After processing one minute duration of the video, the highest frequency of lane status is taken into account. Similarly, the same approach is done for one hour interval. Thus, density of the lane for a day is identified.

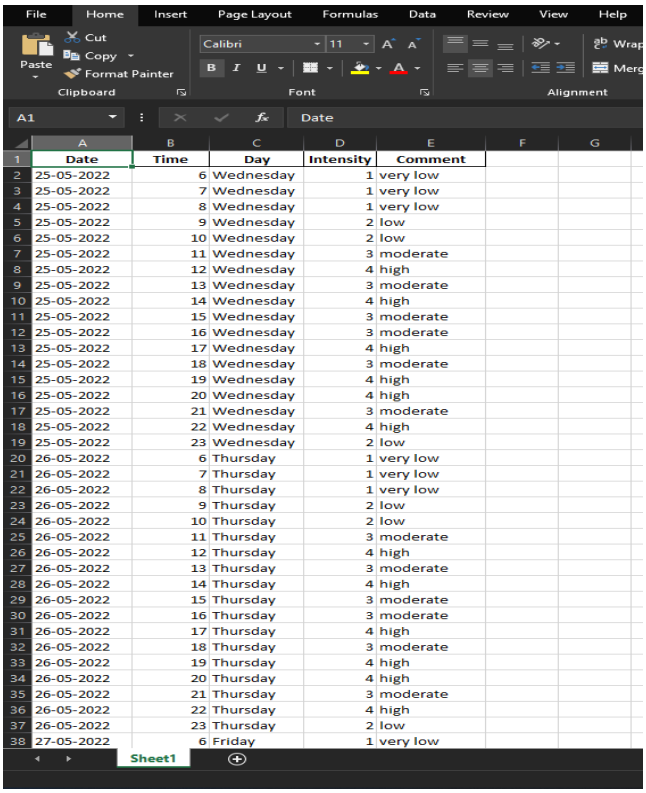
5.3 STORING THE DATA

After each interval of time, status of the lane has to be stored in database along with date and time. As data can be stored in various forms, comma-separated values type of file is used. Header of this file has 'Date', 'Time', 'Day', 'Intensity', 'Comment'. In large scale implementation, a separate csv file is meant for each camera. The system will become more accurate if the dataset is large.

CHAPTER 6

IMPREMENTATION AND EXPERIMENTATION ANALYSIS

6.1 DATABASE USED



	A	B	C	D	E
1	Date	Time	Day	Intensity	Comment
2	25-05-2022	6	Wednesday	1	very low
3	25-05-2022	7	Wednesday	1	very low
4	25-05-2022	8	Wednesday	1	very low
5	25-05-2022	9	Wednesday	2	low
6	25-05-2022	10	Wednesday	2	low
7	25-05-2022	11	Wednesday	3	moderate
8	25-05-2022	12	Wednesday	4	high
9	25-05-2022	13	Wednesday	3	moderate
10	25-05-2022	14	Wednesday	4	high
11	25-05-2022	15	Wednesday	3	moderate
12	25-05-2022	16	Wednesday	3	moderate
13	25-05-2022	17	Wednesday	4	high
14	25-05-2022	18	Wednesday	3	moderate
15	25-05-2022	19	Wednesday	4	high
16	25-05-2022	20	Wednesday	4	high
17	25-05-2022	21	Wednesday	3	moderate
18	25-05-2022	22	Wednesday	4	high
19	25-05-2022	23	Wednesday	2	low
20	26-05-2022	6	Thursday	1	very low
21	26-05-2022	7	Thursday	1	very low
22	26-05-2022	8	Thursday	1	very low
23	26-05-2022	9	Thursday	2	low
24	26-05-2022	10	Thursday	2	low
25	26-05-2022	11	Thursday	3	moderate
26	26-05-2022	12	Thursday	4	high
27	26-05-2022	13	Thursday	3	moderate
28	26-05-2022	14	Thursday	4	high
29	26-05-2022	15	Thursday	3	moderate
30	26-05-2022	16	Thursday	3	moderate
31	26-05-2022	17	Thursday	4	high
32	26-05-2022	18	Thursday	3	moderate
33	26-05-2022	19	Thursday	4	high
34	26-05-2022	20	Thursday	4	high
35	26-05-2022	21	Thursday	3	moderate
36	26-05-2022	22	Thursday	4	high
37	26-05-2022	23	Thursday	2	low
38	27-05-2022	6	Friday	1	very low

Fig 6.1 CSV sheet

The data is stored in CSV file and it has 5 columns.

The dataset has 5 columns,

1. Date
2. Time
3. Day
4. Intensity
5. Comment

“Comment” column is restricted for readability purpose only.

Intensity levels are classified as follows,

Very high – 5

High – 4

Moderate – 3

Low – 2

Very low – 1

6.2 IMPLEMENTATION TECHNIQUES

The stored data is accessed and made available to meet the objectives. It can be implemented by few methods; one way is by using a webpage. A responsive webpage is designed with the help of web technologies along with Flask frame work.

Web technologies involves HTML5 for building elements of webpage, CSS3 for styling the webpage and JavaScript for validation of inputs entered by users.

The website consists of two pages. In first page it consists of two input boxes- Road and Date. User needs to select the road out of given option in which they intend to travel in first input field and needs to choose date of travel in second input field. Validation for input is also done.

The output is shown through in the second page by computing the data available in the dataset. From the frontend part, road value and date value are passed to backend. Name of the week is found out with by Date value. Now pandas library is used to read to csv file and further data extraction is made.

From this data, an image file consists of traffic chart is passed from backend to frontend which is shown in second page. If the entered date falls on any special day which already declared in the program, a message is displayed saying it is a special day.

6.3 DATA EXTRACTION

Extracting the intensity levels for a particular day and time is done by reading it into a Data Frame with the help of pandas library. For calculation purpose, only Day, Time and Intensity columns are considered, other columns will be dropped from Data Frame. Required day is taken into account and data related to that is extracted.

For a given day, a data frame of all three columns data is extracted. It is grouped by Time column. The intermediate result is then further processed to figure out the mean value of all intensity level with respect to time of that particular day. This will generate a mean intensity level of that particular day.

6.4 DATA REPRESENTATION

In order to graphically represent the traffic intensity levels, matplotlib modules is used. This will be 2-D graph which has time in x-axis and intensity in y-axis. The scale of x-axis is one hour and y-axis is 1 unit. The outcome of data extraction is passed as y-axis argument and for x-axis, it will be a range of time from 06:00 to 22:00. A bar graph is

shown for easy understandability to user. Different colors is assigned for bars based on their intensity level.

Light green – very low traffic

Yellow –low traffic

Gold – moderate traffic level

Orange – high traffic

Red – Very high traffic

High traffic is the result of high density but vehicles will be moving slowly, whereas very high traffic is same but without any movement of vehicles.

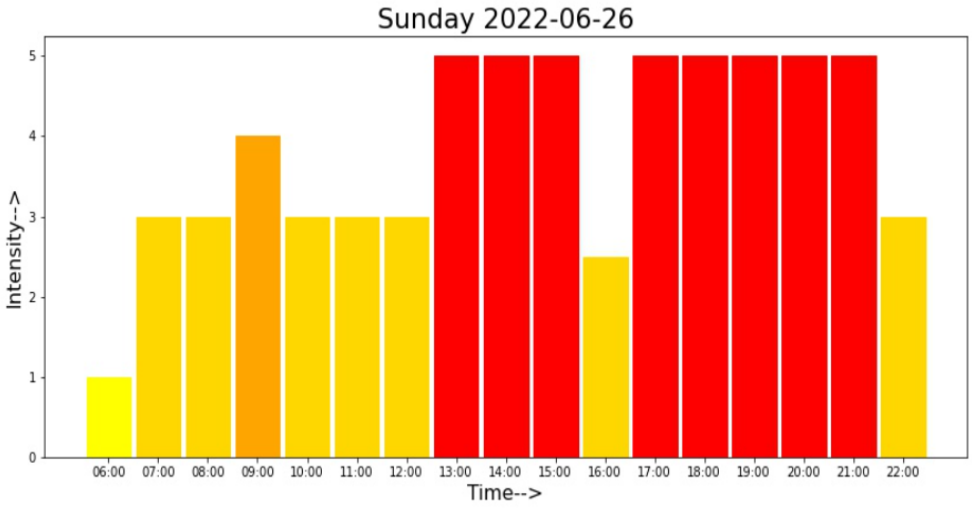


Fig 6.2 Traffic chart

6.5 ACCURACY OF RESULT

Table 6.1 Vehicle movement detection and lane density

Time interval (10s)	No. of vehicles movement detected by system	Actual number of vehicles movement observed	Accuracy of movement detection	Status of the lane by system	Actual status of the lane
1	6	7	85%	Very low	Very low
2	21	19	89%	Moderate	Moderate
3	17	14	78%	Moderate	Moderate
4	19	16	81%	Moderate	High

6.6 TIME COMPLEXITY

The time required for video processing by the proposed system mainly depends on dimension of frame. As increase in resolution image causes increase in size of image. Images are represented as 2-D arrays while video processing. Thus, size of matrix also increases and time required for computation is also increased.

For a frame having resolution of 300*400, Image processing follows multiple operations which involves image downsampling, image normalization, image subtraction, color space transformation, filtering, thresholding, closing and computation of light pixel density. These process works on every pixel of an image and takes sufficient amount of time. This can be reduced by image cropping and image downsampling so that region of interest is only considered. It has significant effect in reducing time complexity and accuracy of proposed method can also be increase. Anyways, system must be configured in order to set limits to crop an image.

Frame rate supported by opencv2 module depends on the computations that is done after video capturing. As long as time taken for the system to perform operation on a frame is less than 1/FPS of video, frame drop cannot be seen. For example, in a 30 FPS video, the system should take less than one second in order to process 30 frames.

Storing and extraction of data from a file has time complexity in the order of size of the file. Time required for further computation using pandas and numpy module depended on the size of data.

6.7 DISCUSSION ON RESULTS

Status of the lane is decided by two parameters, one is movement of vehicles and another one is frame density that is calculated in video process. Sometimes the movement of vehicles doesn’t get captured and multiple movements is detected for a single movement of vehicle. The problem here considered is obviously complex, and the provided results need further improvement for example: refinements of the object detection algorithms.

The idea behind vehicle movement detection is shown in Fig 6.3. Proposed system detects the vehicle and draws contour over it. Centroid of each rectangular box is also shown in same figure. An imaginary line is drawn to assist vehicle movement detection procedure.

Fig 6.4 shows bi-level representation of an image, which will assist for density calculation of the image.

Fig 6.5 contains outputs of image which has undergone RGB to Gray transformation, Gaussian blurring, Thresholding and Closing transformation respectively. The output of proposed system which will be stored in a csv file in shown in Fig 6.6. To know the traffic levels of desired day, a responsive webpage is developed as shown in Fig 6.7 and Fig 6.8. The same can be accessed through mobile as shown in Fig 6.9.

6.8 SNAPSHOTS OF RESULTS

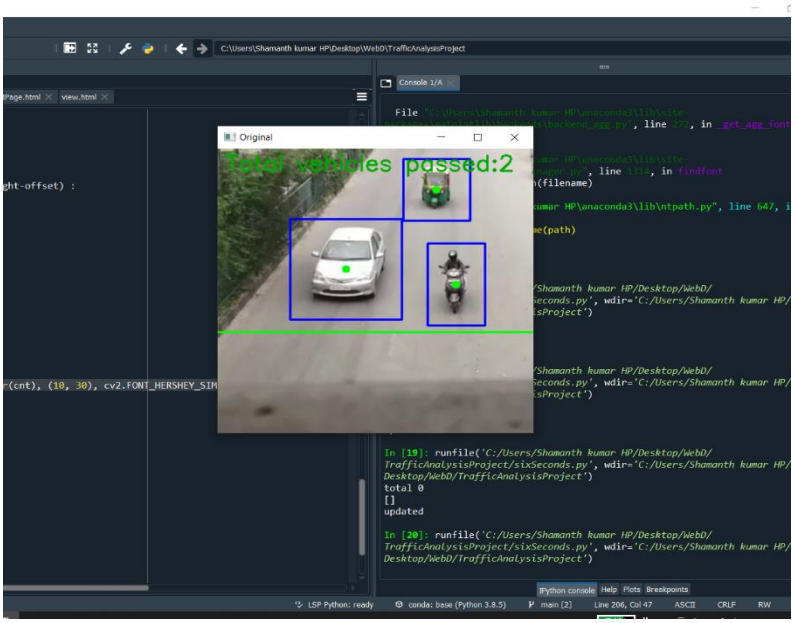


Fig 6.3 Movement detection

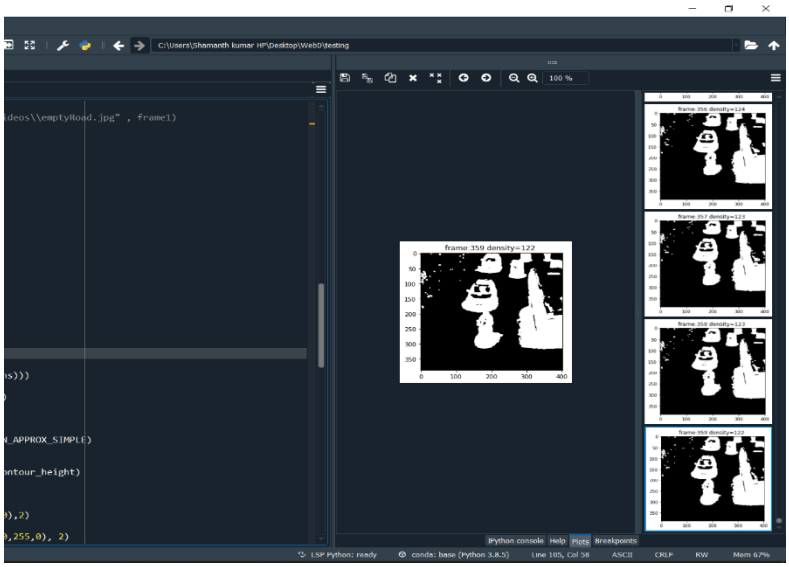


Fig 6.4 Density calculation

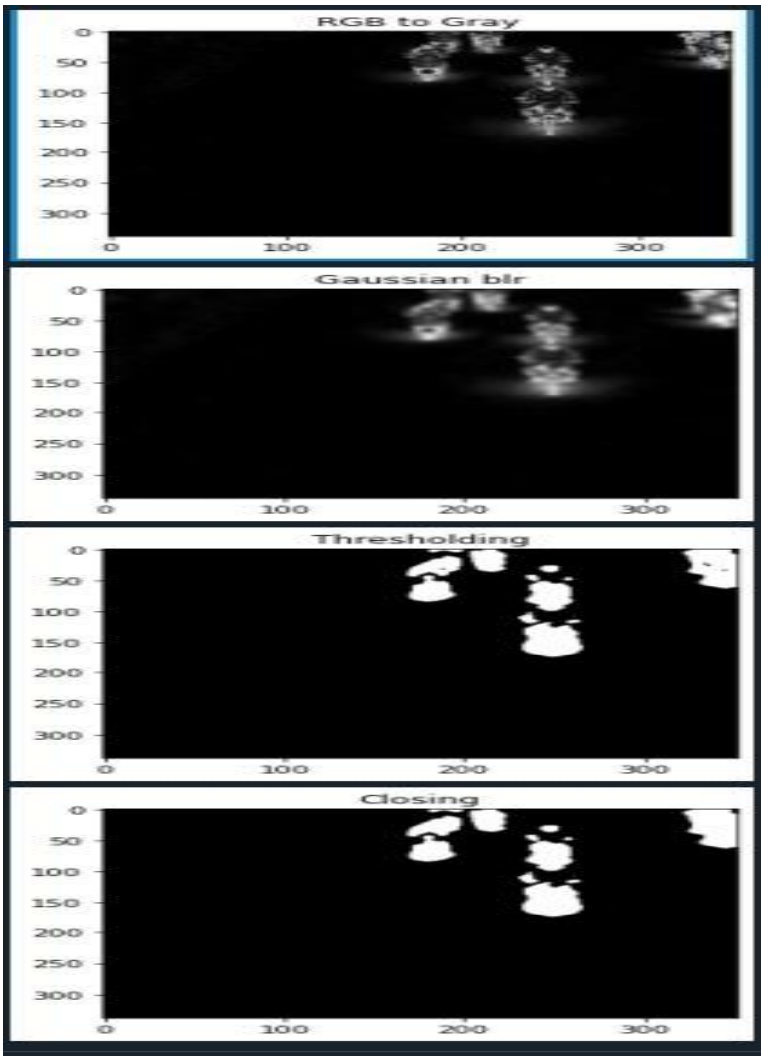
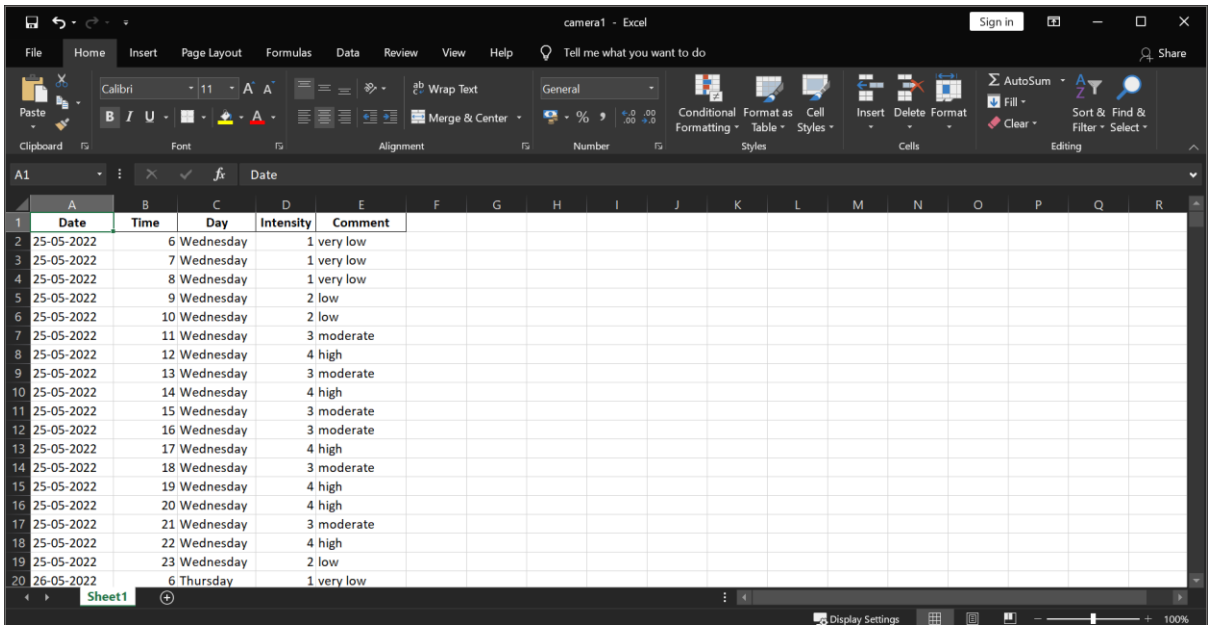


Fig 6.5 Image processing



camera1 - Excel

	Date	Time	Day	Intensity	Comment
2	25-05-2022	6	Wednesday	1	very low
3	25-05-2022	7	Wednesday	1	very low
4	25-05-2022	8	Wednesday	1	very low
5	25-05-2022	9	Wednesday	2	low
6	25-05-2022	10	Wednesday	2	low
7	25-05-2022	11	Wednesday	3	moderate
8	25-05-2022	12	Wednesday	4	high
9	25-05-2022	13	Wednesday	3	moderate
10	25-05-2022	14	Wednesday	4	high
11	25-05-2022	15	Wednesday	3	moderate
12	25-05-2022	16	Wednesday	3	moderate
13	25-05-2022	17	Wednesday	4	high
14	25-05-2022	18	Wednesday	3	moderate
15	25-05-2022	19	Wednesday	4	high
16	25-05-2022	20	Wednesday	4	high
17	25-05-2022	21	Wednesday	3	moderate
18	25-05-2022	22	Wednesday	4	high
19	25-05-2022	23	Wednesday	2	low
20	26-05-2022	6	Thursday	1	very low

Fig 6.6 Data stored in CSV file

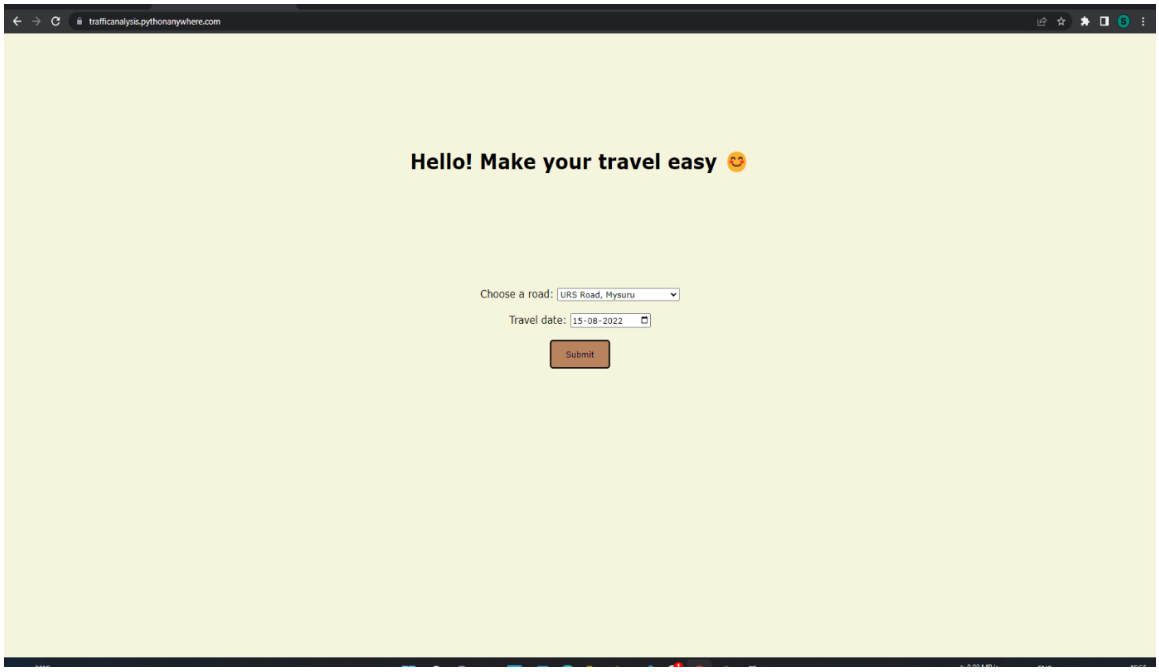


Fig 6.7 Responsive Website in desktop:

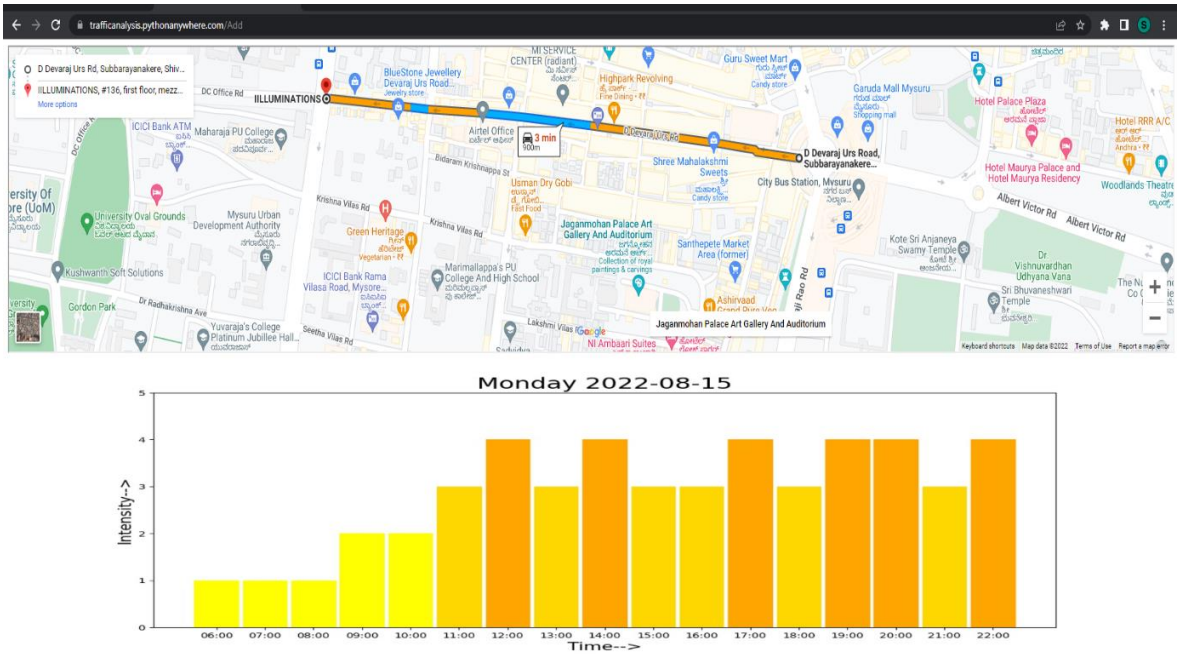


Fig 6.8 Output of webpage

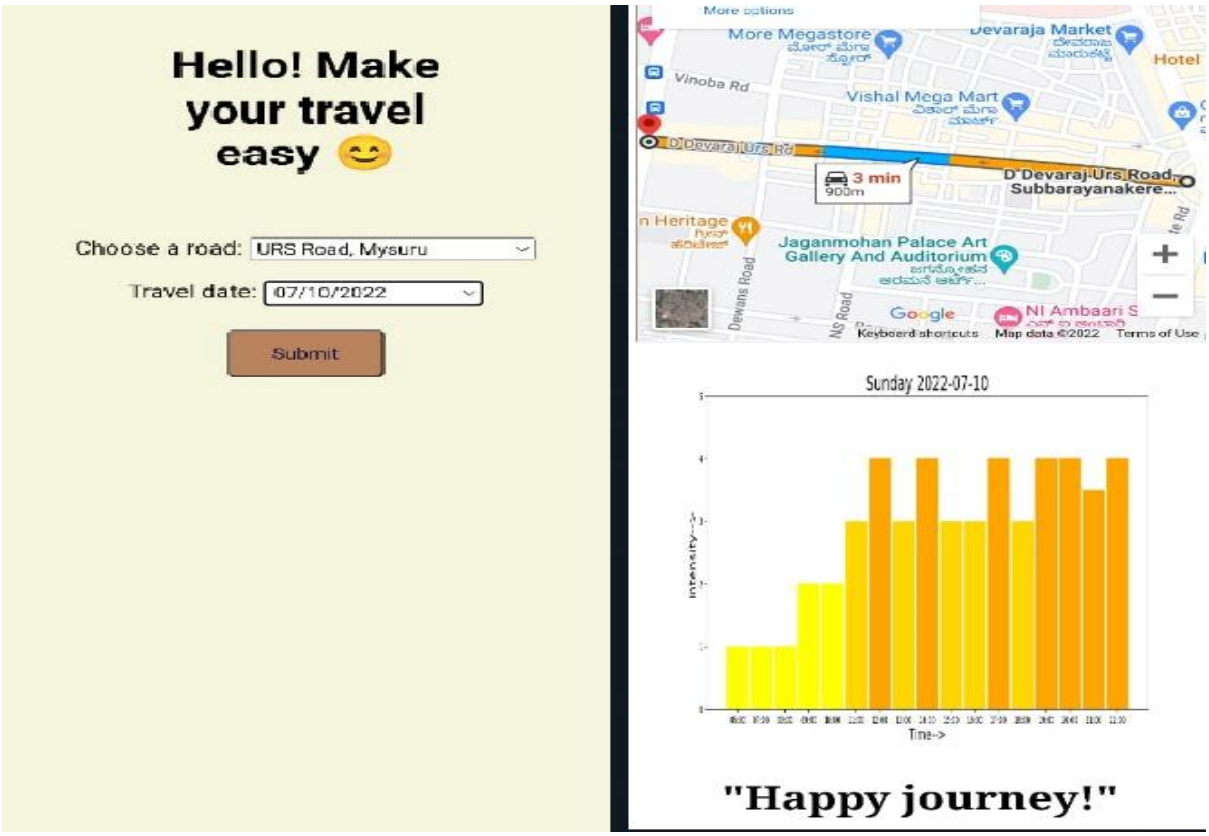


Fig 6.9 Responsive Website in mobile:

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

Smart cities appear as “the next stage of urbanization, subsequent to the knowledge-based economy, digital economy, and intelligent economy”. Road accidents have been a major cause for concern across the Indian subcontinent. In 2020 alone, the country reported nearly 132 thousand fatalities due to road accidents. Each year, about three to five percent of the country’s GDP was invested in road accidents. Almost 70 percent of the accidents involved young Indians. Better insights into the causes of road congestion and its management, are of vital significance to avoid or minimize loss to public health, deaths and injuries, and other socio-economic losses and environmental damages.

Many roads traffic modelling, analysis, and prediction methods have been developed to understand the causes of road traffic congestion, and to prevent and manage road congestion. The forecasting or prediction of road traffic characteristics, such as speed, flow and occupancy, allows planning new road networks, modifications to existing road networks, or developing new traffic control strategies. Deep learning is among the leading-edge methods used for transportation related predictions. However, the existing works are in their infancy, and fall short in multiple respects, including the use of datasets with limited sizes and scopes, and insufficient depth of the deep learning studies.

Based on our project using video processing is a good technique to control road congestion. It is also more consistent in detecting vehicle presence since it utilizes genuine traffic frames. It envisions the reality so it works far superior to systems which depend on the detection of vehicles. This work can be upgraded further by proposing a framework for controlling the traffic density. That will decrease our serious issue of everyday life, traffic jam. Using video processing, object density and movement detection the proposed system achieves good accuracy in identifying the objects and estimating lane density. Based on lane density the traffic issue can be resolved in greater extent.

Rapid growth in the number of vehicles and lack of efficient monitoring of vehicles has led to traffic jams. The method used above of detecting the density of vehicles and processing the amount of traffic, can be used for controlling the traffic, avoiding traffic congestion, accidents, etc. Using this method at each crossway could help in a continuous journey of the people.

i. The future updated project can include traffic prediction of specific roads that can be used as an alternate for a lane, and during the time interval of high or very high traffic density the alternate route suggestions can be provided.

ii. To make this more user friendly, accessing of intensity chart through web page can be replaced by mobile application.

iii. Currently, the output traffic intensity is given for every hour of time interval, the precision can be increased by scaling this to every quarter or half hour.

REFERENCE:

[1] Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, published by Pearson Education, Inc, publishing as Prentice Hall, 3rd Edition, 2009.

[2] X. Fu, Z. Wang, D. Liang, J. Jiang, "The Extraction of Moving Object in Real- Time Web-Based Video Sequence," 8th International Conference on Digital Object Identifier, Vol. 1, pp. 187-190, 2004

[3] P. Choudekar, S. Banerjee, M. K. Muju, "Real Time Traffic Light Control Using Image Processing," Indian Journal of Computer Science and Engineering, Vol. 2, No. 1, ISSN: 0976-5166.

[4] Reha Justin, Dr Ravindra Kumar, "Vehicle detection and counting method based on digital image processing in python," International Journal of Electrical Electronics and Computer science Engineering, ICSCAAIT-2018, ISSN:2348- 2273,2018.

[5] M. Seki, H. Fujiwara, and K. Sumi, "A robust background subtraction method for Changing background", Fifth IEEE Workshop on Applications of Computer Vision, 2000.

[6] M. Lei, D. Lefloch, P. Gouton, K. Madani, "A video based real-time vehicle counting system using adaptive background method", IEEE International conference on Signal Image Technology and Internet Based Systems (SITIS'08), pp. 523-528, 2008.

[7]1d Stalin Alex, 2dr. Amitabh Wahi "BSFD: background subtraction frame difference algorithm for moving object detection and extraction"

[8] Tanvi Sable, Nehal Parate, Dharini Nadkar, Swapnil Shinde, "Density and Time-based Traffic Control System using Video Processing"

[9] Rafael C. Gonzalez , Richard E. Woods, "Digital Image Processing" Fourth Edition, Global Edition, ISBN 978-0-13-335672-4, 2018

[10] S. Sri Jamiya and E. Rani, "A survey on vehicle detection and tracking algorithms in real time video surveillance"

[12] E. Bas, A.M. Tekalp, and F.S. Salman, “Automatic vehicle counting from video for traffic flow analysis”, IEEE Intelligent Vehicles Symposium, 2007.

[13] R.T. Collins, et al., “A system for video surveillance and monitoring”, VASM final Report, Robotics Institute, Carnegie Mellon University, 2000, pp.1-68.

[14] N. Seenouvong, U. Watchareeruetai, C. Nuthong, K. Khongsomboon, “A Computer Vision Based Vehicle Detection and Counting System”, IEEE 8th International conference on Knowledge and Smart Technology (KST), pp.224- 227, 2016.

[15] A. Suryatali, V.B. Dharmadhikari, “Computer Vision Based Vehicle Detection for Toll Collection System Using Embedded Linux”, International Conference on Circuit, Power and Computing Technologies (ICCPCT), pp. 1-7, 2015.

[16] Shreeraz memon, Sania bhatti, “A Video Based Vehicle Detection, Counting and Classification System”, Computer system and software engineering, pp. 34- 41, 2018.

[17] Ye Tian, Yangli Wang, Rui Song, “Accurate Vehicle Detection and Counting Algorithm for Traffic Data Collection”, International Conference on Connected Vehicles and Expo, 2015.

[18] R. A. Hadi, G. Sulong, and L. E. George, "Vehicle detection and tracking techniques: a concise review," arXiv preprint arXiv:1410.5894, 2014.

[19] Georgios Vigos, Markos Papageorgioua, Yibing Wangb, “Real-time estimation of vehicle-count within signalized links”, Journal of Transportation Research Part C: Emerging Technologies, Volume 16, Issue 1, pp.18–35, February 2008.

[20] Anthony J. Venables, “Evaluating Urban Transport Improvements”, Journal of Transport Economics and Policy, Vol. 41, No.2, pp. 173- 188, May, 2007.

[21] Salamh, A.B.S. (2017) “Investigation the Effect of Using Gray Level and RGB Channels on Brain Tumor Image”, In: Computer Science & Information Technology (CS & IT), pp. 141–148.

[22] M Hansen, Yu B. Wavelet thresholding via mdl for natural images. IEEE Trans Inf Theory 2000;46(5) pp:1778-88.

[23] E Simoncelli, E Adelson. Noise removal via Bayesian wavelet coring. Proc IEEE Int Conf Image Process 1996 pp:379-82.

[24] Evans, A. N. and Liu, X. U. [2006]. "A Morphological Gradient Approach to Color Edge Detection," IEEE Trans. Image Proc., vol. 15, no. 6, pp. 1454-1463.

[25] Kim, C. (2005) "Segmenting a Low-Depth-of-Filed Image Using Morphological Filters and Region Merging." IEEE Trans Image Proc. vol. 14, no. 10, pp. 1503-1511.