Density-based Traffic Control System using Image Processing

A Dissertation submitted for the partial fulfilment

Of the degree of

Bachelor of Engineering in

Electronics and Telecommunication

(Session 2022 - 2023)

Guided By

Mrs. Raksha Upadhyay

Professor

Group Members

Md Kashif Raza - 19T7037

Milisha Gupta - 19T7038

Sakshi Magre - 19T7054

Kiranjeet Kaur - 19T7081

Department of Electronics and Telecommunication

Institute of Engineering and Technology

Devi Ahilya Vishwavidyalaya, Indore (M.P)

(www.ietdavv.edu.in)

DEC - 2022

Recommendation

The dissertation entitled "**Density-based traffic control using Image Processing**" submitted by **Md Kashif Raza, Milisha Gupta, Sakshi Magre** and **Kiranjeet Kaur** is a satisfactory result of the bonafide work done under my guidance is recommended towards the partial fulfilment for the award of **Bachelor of Engineering in Electronics and Telecommunication** degree by **Devi Ahilya Vishwavidyalaya, Indore**.

Date:

Mrs. Raksha Upadhyay
(Associate Professor)
Project Guide

Endorsed By:
Head, Department of Electronics and Telecommunication

Dissertation

Approval sheet for the dissertation entitled "Density Based Traffic Control using Image Processing" submitted by is approved as partial fulfilment for the award of Bachelor of Engineering in degree by Devi Ahilya Vishwavidyalaya, Indore.

Internal Examiner **External Examiner**

Director

Institute of Engineering & Technology

Devi Ahilya Vishwavidyalaya, Indore (M.P)

Candidate Declaration

We at this moment declare that our project entitled "**Density-based traffic control using Image Processing**" is a result of my own work and that it has not been submitted anywhere else for any degree or diploma carried out under the supervision and guidance of **Mrs. Raksha Upadhyay (Associate Professor in Electronics and Telecommunications Department),** Institute of Engineering and Technology, Devi Ahilya Vishwavidyalaya, Indore. I also declare that I have duly acknowledged all sources of help, assistance and ideas in this work.

We hereby declare that all the information and data in this thesis are true and accurate to the best of my knowledge and belief.

We take full responsibility for the contents of this thesis and We guarantee that we have not violated any copyright or other laws and regulations in the course of this research.

Date:

Place:

Md Kashif Raza

Milisha Gupta

Sakshi Magre

Kiranjeet Kaur

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to all those who have contributed to the successful completion of this project.

We would like to especially thank our Guide, **Mrs. Raksha Upadhyay (Associate Professor)**, who gave us continuous guidance, assistance and inspiration to conduct this project and obtain promising results.

We would like to thank **Dr Uma Bhatt (Assistant Professor, E&TC)** and **Dr Sanjeev Tokekar (Director and HOD, E&TC, IET DAVV)** for their advice, help and appreciation which motivated us to complete our work.

We would also like to thank the **Electronics and Telecommunication** department Staff for helping us and providing us with the workspace whenever required.

ABSTRACT

The main purpose of this project is to control the traffic signals based on the density of each lane i.e., the number of vehicles travelling in each lane in real-time through capturing video or CCTV footage and intelligently changing the timings of signals opening each lane.

This prototype is developed with the minimum number of circuits and hardware. The system can contribute to efficiently managing the traffic flow, prioritizing the vehicle of importance, and dynamically changing the traffic corresponding to a higher number of vehicles.
This is also reducing the manual work of entering and updating the signals every day or every event, and having the number of service workers present to control the traffic.

The objective of this paper is to detect the number of vehicles in real-time and estimate the average density for each lane and change the timings of each signal in multiway intersection lanes.

As the population is surging rapidly and due to the increase in the number of vehicles and in general traffic day by day, the problem of traffic congestion, and lack of service providers this problem needs immediate attention to smoothen the traffic.

How we will overcome this problem with our proposed solution:

1. The number of vehicles can be counted so the system can precisely estimate the time for each lane to be free from congestion.
2. The system works in real-time hence no need to manually enter the values for each signal.
3. It can prioritize the vehicle based on the modelling for each vehicle to detect, for example, any Ambulance, Fire Fighter or Military Personnel can use this to open up the traffic signal for more time than others to let the vehicle move to its destination.
4.

List of Figures

Figure **Page No.**

TABLE OF CONTENTS

Chapter 1 - Introduction

1.Introduction

1.1 Overview and issues involved

In modern life we have to face with many problems, one of which is traffic congestion becoming more serious day after day. It is said that the high volume of vehicles, the inadequate infrastructure and the irrational distribution of the development are main reasons for increasing traffic jams.

We all know that world population is increasing day by day, so the need of transportation is also rising. This constant need leads to increase in number of vehicles. Most of the countries use traditional traffic signal system which has timer at its core to control the traffic lights in each lane. This system causes major time delays, slower speeds and longer journeys because of time wasted on lanes with lower densities or sometimes no density at all.

Primary causes of traffic congestion include the following:

- Traffic accidents blocking lanes
- Difficult weather conditions
- Unexpected spikes in traffic demand
- Traffic bottlenecks
- Disabled traffic control equipment

In many cases, multiple factors can occur at once. One cause of traffic congestion can lead to another, such as when the onset of challenging weather for road travel generates a rise in traffic as drivers rush home.

Traffic creates more problems than just commuting delays. Traffic congestion can cause compounding issues for delivery drivers and the supply chain, slowing the restocking of food and other essentials to communities that don’t have many local sources. High congestion can also influence the public’s perception of a community. It may discourage new businesses and potential residents from moving to the area, for fear of constant traffic.

With the uncontrolled population growth, travelling has turned out to be a really hectic task in today's world. The increase in travelling people has caused a drastic growth in traffic at every nook and corner of the city. This, in turn, is contributing to the wastage of precious fuel and time which leads to impatience and frustration of the people. Traffic congestion is a common problem that has arisen due to the increased number of vehicles on the road. In order to deal with this problem, researchers have proposed many solutions. One of the currently used models is the timer model. Traffic can be controlled to a great extent by using timers at each phase of the traffic. Another model used is with the help of electronic sensors which detects the presence of vehicles, and produce appropriate signals. The cause of traffic is dependent on many factors like peak time, special days, season, bad weather, or unexpected events like accidents, special events or constructional activities. Once we get stuck in traffic, we may have to wait for hours to get out of it. We can solve this problem to a great extent by implementing this density-based traffic control system using image processing which continuously manages the traffic lights based on traffic. This system uses image processing techniques such as background subtraction in order to find the count of vehicles present on the road (traffic density) which can be used to control the traffic signal light. The image processing tools which are present in MATLAB can be used to program the code for finding the count of vehicles in a lane. This idea can be represented using a timer (Seven Segment Display) and LEDs instead of real traffic lights.

1.2 Problem Definition

1.In 1982, the average person living in one of the country's 75 largest cities faced seven hours of travel delay per year," the report states. "By 2001, that figure had shot to 26 hours of delay per year, and the most severely congested periods of the day — once known as the 'rush hour' — stretched to cover nearly six hours of each day … with the average 'rush hour' trip taking nearly 40 percent longer than the same trip at other times of the day.

2. A metropolitan city like Mumbai has a car density of around 510 vehicles every kilometer, followed by Pune and Kolkata which has 359 and 319 vehicles per kilometer. Now controlling this kind of heavy traffic is a tedious task. So, we need some advanced traffic management system to control this traffic.

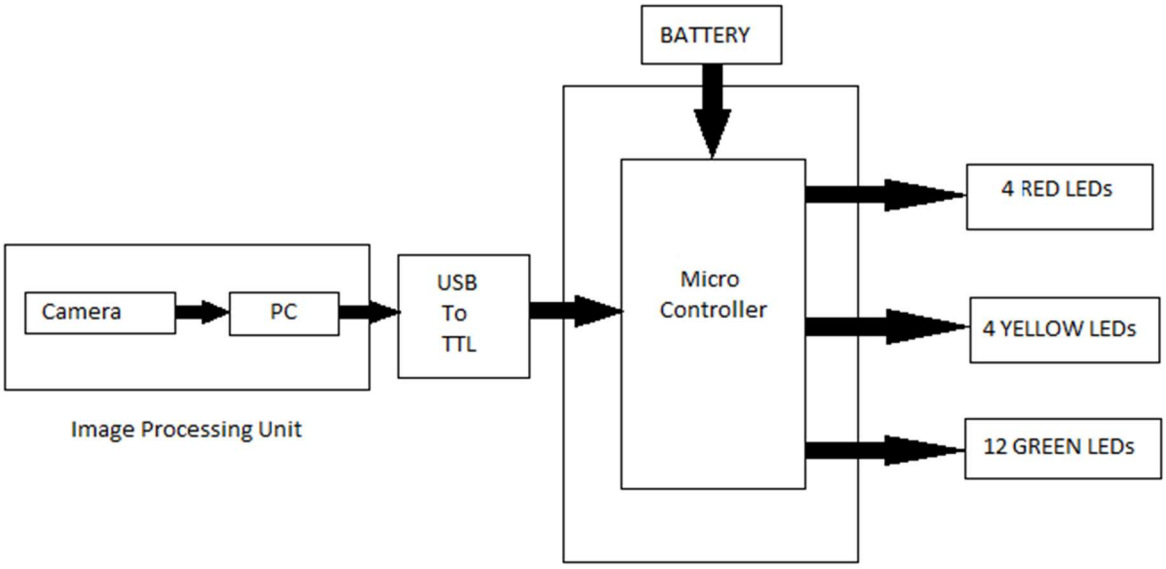
3. Once we get stuck in traffic, we may have to wait for hours to get out of it. We can solve this problem to a great extent by implementing this density-based traffic control system using image processing which continuously manages the traffic lights based on traffic.

4.The problem with the existing traffic management system is that it has a fixed timer and due to which it waits for a particular lane even when there is no vehicle which leads to wastage of time.

1.3 Proposed Solution

Here we propose a system which uses a webcam to capture images of each lane and these images are given to the MATLAB for image processing. After using various MATLAB algorithms density of each lane is calculated. A microcontroller is used to compare all the densities, prioritize them and show the output on the signal system.

In our system camera is dynamically capturing entire top view of the four lanes and sending it to the PC. PC will process the received images using MATLAB software and gives the information about the traffic density present in each lane. PC will send this information from processed image to the microcontroller using USB to TTL cable. USB to TTL is used for serial communication between PC and microcontroller. The information taken from PC is converted into Binary form followed by calculation of Density. Arduino will control the Traffic signal lights.



Block Diagram of Proposed System

To find out how density is calculated we have created our own database where we have considered matchboxes representing vehicles.

Fig-2: Reference Image

Fig-3: Lane 1

Fig-4: Lane 2

Let us consider above three images where an image with no vehicles is a reference image and following two images are two lanes with different densities of vehicles. MATLAB software is

used to compare these two lanes with reference image. By comparing the two images such that reference image and captured image we get difference between two images.

As we can see background of reference and captured image is same but number of moving objects which are vehicles is changing. This change is nothing but a difference between two images. A new image is formed after calculating the difference which is known as binary image. This binary image is then used to calculate density of vehicles in an image.

Fig-5: Output on MATLAB window

As we can see in the output image of MATLAB, after using various image processing algorithms density of both the lanes is calculated. This density is compared and the lane with highest density is shown.

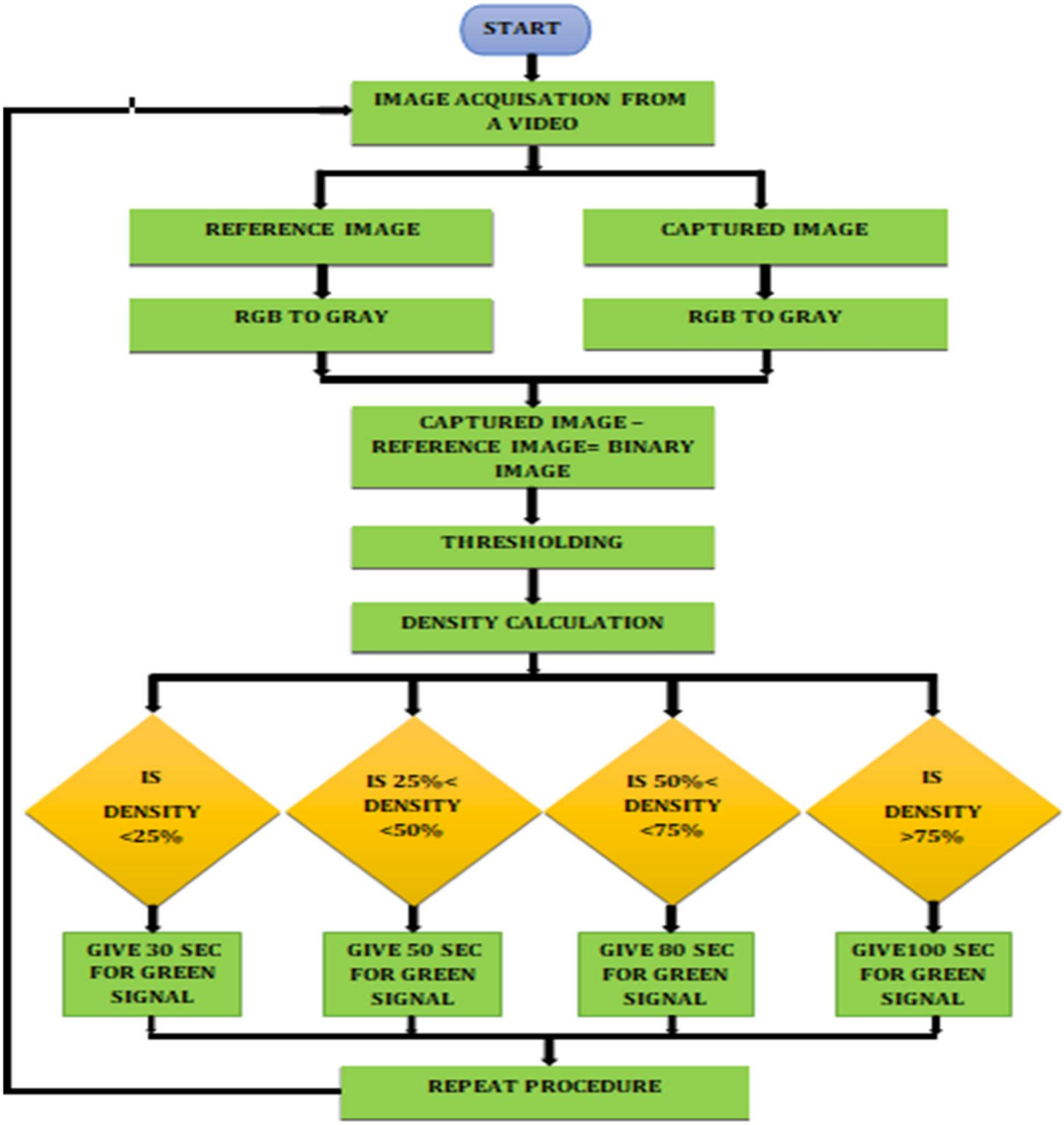
Consider a signal system which uses image processing to control the traffic. Now consider a camera taking videos of each lane continuously, so at what time a frame should be captured and given to the software for further processing. By using some MATLAB functions, we took a video from the hardware of the computer. We set the time for which a frame is to be captured. This captured image goes through various image processing techniques.

How our solution is better than existing solutions: -

Manual controlling refers to controlling traffic with the help of manpower. Traffic police are allotted to a specified area to control traffic. Controlling traffic signals using timers and electrical sensors is known as automatic controlling. In this method, a constant numerical value is uploaded into the timer at each phase of the traffic light. According to the changes in the timer value, the traffic lights automatically become ON or OFF. The electrical sensors check for the presence of vehicles on the road and provide appropriate signals at each phase, which controls the automatic switching off lights. The manual controlling system requires tremendous manpower for implementation. Since the number of traffic police is limited and making them do this sort of work in the hot tropical areas is cruel. We need to find a better method to control traffic in the cities. In an automatic traffic controlling system, the traffic light is controlled by timers at every stage or by using electronic sensors to detect the presence of vehicles. But in these methods, time could be wasted in showing a green signal when there are no vehicles at all. We can get over these limitations by controlling traffic lights with the help of image processing techniques.

Our proposed system controls traffic lights using image processing techniques. Here, the vehicles are detected by considering the captured images instead of using timers or electronic sensors which are placed on the pavement. A web camera is placed at the traffic light which captures images of the road through which the vehicle count is estimated and traffic is controlled. Using image processing in traffic control is found to be a better technique than the

existing methods. It helps decrease traffic congestion without any wastage of time caused by showing green signal on an empty road without any vehicles. It is a better way to determine the presence of vehicles since it makes use of real-time images which makes it better than systems depending on the vehicle's metal content only.


START
IMAGE ACQUISATION FROM A VIDEO
REFERENCE IMAGE
CAPTURED IMAGE
RGB TO GRAY
RGB TO GRAY
CAPTURED IMAGE – REFERENCE IMAGE= BINARY IMAGE
THRESHOLDING
DENSITY CALCULATION
IS DENSITY <25%
IS 25%< DENSITY <50%
IS 50%< DENSITY <75%
IS DENSITY >75%
GIVE 30 SEC FOR GREEN SIGNAL
GIVE 50 SEC FOR GREEN SIGNAL
GIVE 80 SEC FOR GREEN SIGNAL
GIVE100 SEC FOR GREEN SIGNAL
REPEAT PROCEDURE

Chapter 2: Literature Survey

2. Literature Survey

2.1 Methodology

We have divided the approach into four steps:

i) Image Acquisition

The first step is to capture the images as reference so as to get the data for further processing. In our system the camera is dynamically capturing the entire top view of the four lanes and sending it to the PC. It is acquired from the hardware present in the computer.

ii) RGB to Grayscale image conversion

This is the step undertaken to further simplify the process of image subtraction. Using the thresholding boundary assigned, pixels with luminous intensity greater than the thresholding level are assigned white and others with black color.

iii) Vehicle count extraction

We compare the image captured with our reference image to calculate the difference of the two images. Reference image is the image which is taken without traffic. This difference is used to calculate the count of the vehicles in that particular lane.

iv) Updating the signals in Arduino

The count of the lanes is used further to calculate the timing for green lights in that lane. This is then implemented in the hardware by providing dynamic timing for signal lanes.

2.2 Technologies and Tools

MATLAB Software

MATLAB, short for matrix laboratory, is a fourth-generation high-level programming language and interactive environment for numerical computation, visualization and programming.

It allows matrix manipulations; plotting of functions and data; implementation of algorithms; creation of user interfaces; interfacing with programs written in other languages, including C, C++, Java, and FORTRAN; analyze data; develop algorithms; and create models and applications.

It has numerous built-in commands and math's functions that help in mathematical calculations, generating plots, and performing numerical methods.

Features of MATLAB

Following are the basic features of MATLAB −

1. **High level language**

   It is a high-level language for numerical computation, visualization and application development.

2. **Interactive environment**

   It also provides an interactive environment for iterative exploration, design and problem solving.

3. **Library of mathematical functions**

   It provides a vast library of mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration and solving ordinary differential equations

4. **Built-in Graphics**

   It provides built-in graphics for visualizing data and tools for creating custom plots.

5. **Development tools improve performance**

MATLAB's programming interface gives development tools for improving code quality maintainability and maximizing performance.

6. **Tools for custom graphical interfaces**

It provides tools for building applications with custom graphical interfaces.

7. **Easy integration with external applications**

It provides functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET and Microsoft Excel.

MATLAB for Image Processing

Obtaining the number of objects contained in an image. The number of objects contained in an image can be determined through the following steps:

- Removal of objects whose numerical value is not required.
- Making the image have a uniform background by removing its original background.
- Changing the image to a grayscale image.
- Creating a binary version of the image will allow for a numerical analysis of the objects to be analyzed.

Arduino UNO

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller;

can be simply connected to a computer with a USB cable or powered by an AC-to-DC adapter or battery to get started.

Arduino Uno Features

This board comes with all the features required to run the controller and can be directly connected to the computer through USB cable that is used to transfer the code to the controller using IDE (Integrated Development Environment) software, mainly developed to program Arduino. The features of Arduino Uno.

1. **More frequency and number of instructions per cycle**

   Atmega328 microcontroller is placed on the board that comes with a number of features like timers, counters, interrupts, PWM, CPU, I/O pins and based on a 16MHz clock that helps in producing more frequency and amount of instructions/cycle.

2. **Built-in regulation**

   This board comes with a built-in regulation feature which keeps the voltage under control when the device is connected to the external device.

3. **Flexibility & Ease of use**

   There are 14 I/O digital and 6 analog pins incorporated in the board that allows the external connection with any circuit with the board. These pins provide the flexibility and ease of use to the external devices that can be connected through these pins.

4. **Configurable pins**

   The 6 analog pins are marked as A0 to A5 and come with a resolution of 10 bits. These pins measure from 0 to 5V, however, they can be configured to the high range using analog Reference () function and AREF pin.

5. **Quick Start**

   Reset pin is available in the board that reset the whole board and takes the running program in the initial stage. This pin is useful when the board hangs up in the middle of

the running program; pushing this pin will clear everything up in the program and start the program right from the beginning.

6. **Greater Flash Memory**

   13KB of flash memory is used to store the number of instructions in the form of code.

7. **Low Voltage Requirement**

   Only 5 V is required to turn the board on, which can be achieved directly using a USB port or external adapter, however, it can support external power sources up to 12 V which can be regulated and limited to 5 V or 3.3 V based on the requirement of the project.

8. **Plug & Play**

   There is no hard and fast interface required to connect the devices to the board. Simply plug the external device into the pins of the board that are laid out on the board in the form of the header.

9. **USB interface**

   Arduino Uno comes with a USB interface i.e.; a USB port is added on the board to develop serial communication with the computer.

10. **Power alternatives**

    Apart from USB, battery or AC to DC adapter can also be used to power the board.

11. **More Storage**

    There is a provision of Micro SD card to be used in the boards to make them store more information.

Arduino uno with MATLAB

MATLAB addresses several challenges with traditional Arduino programming. The products support two primary workflows:

1. **Read, Write, and Analyze Data from Arduino Sensors**

MATLAB support package for Arduino allows MATLAB programs that read and write data to Arduino and access connected devices such as motors, LEDs, and I2C devices. Because MATLAB is a high-level interpreted language, prototyping and refining algorithms for Arduino projects is easy, and results can be seen from I/O instructions immediately, without recompiling. MATLAB includes thousands of built-in math's, engineering, and plotting functions that can be used for Arduino programming.

2. **Develop Algorithms that Run Standalone on the Arduino**

Simulink support package for Arduino lets the development of algorithms in Simulink, a block diagram environment for modelling dynamic systems and developing algorithms, and run them standalone on your Arduino. The support package extends Simulink with blocks for configuring and accessing Arduino sensors, actuators, and communication interfaces. After creating a Simulink model, it can be simulated, tune algorithm parameters until it is just right, and download the completed algorithm for standalone execution on the device.

Benefits of using MATLAB for Arduino programming:

1. Read and write sensor data interactively without waiting for code to compile.
2. Develop algorithms and analyze sensor data using thousands of pre-built functions for signal processing, machine learning, mathematical modelling, and more
3. Quickly visualize data using the vast array of plot types in MATLAB

2.3 Existing Solutions

The existing system of traffic light management does not consider the density of vehicles in the given lanes. It is a static system. It allots a fixed amount of time to each lane irrespective of the traffic density. This process is repeated in a cyclic order to provide the passage of vehicles from all the lanes.

Advantages of Existing Solutions:

1. They help for movement of traffic securely without any collision.
2. They can increase the capacity of traffic handling at the intersection.
3. They help for the safe movement of slow-moving traffic by interrupting heavy traffic at regular intervals.
4. It is accurate and economical as compared to traffic police control.

Disadvantages of Existing Solutions:

1. They delay the traffic by stopping the vehicles at the intersection during peak hours.
2. They cause a significant increase in rear-end collisions.

3.Analysis

3.Analysis

3.1 Block Diagram

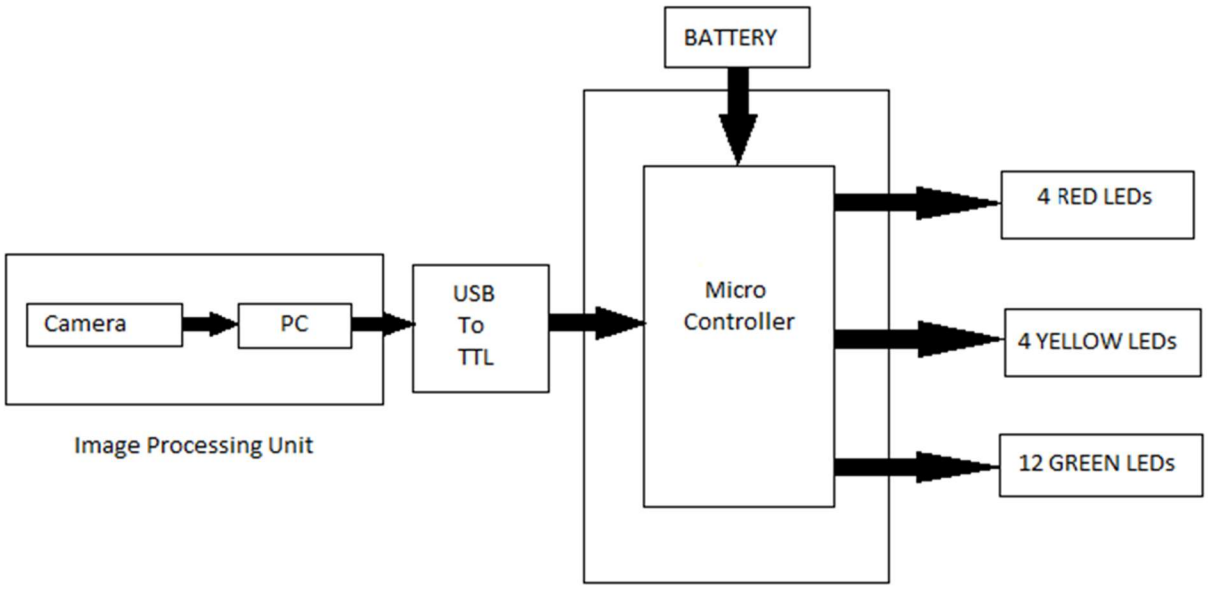
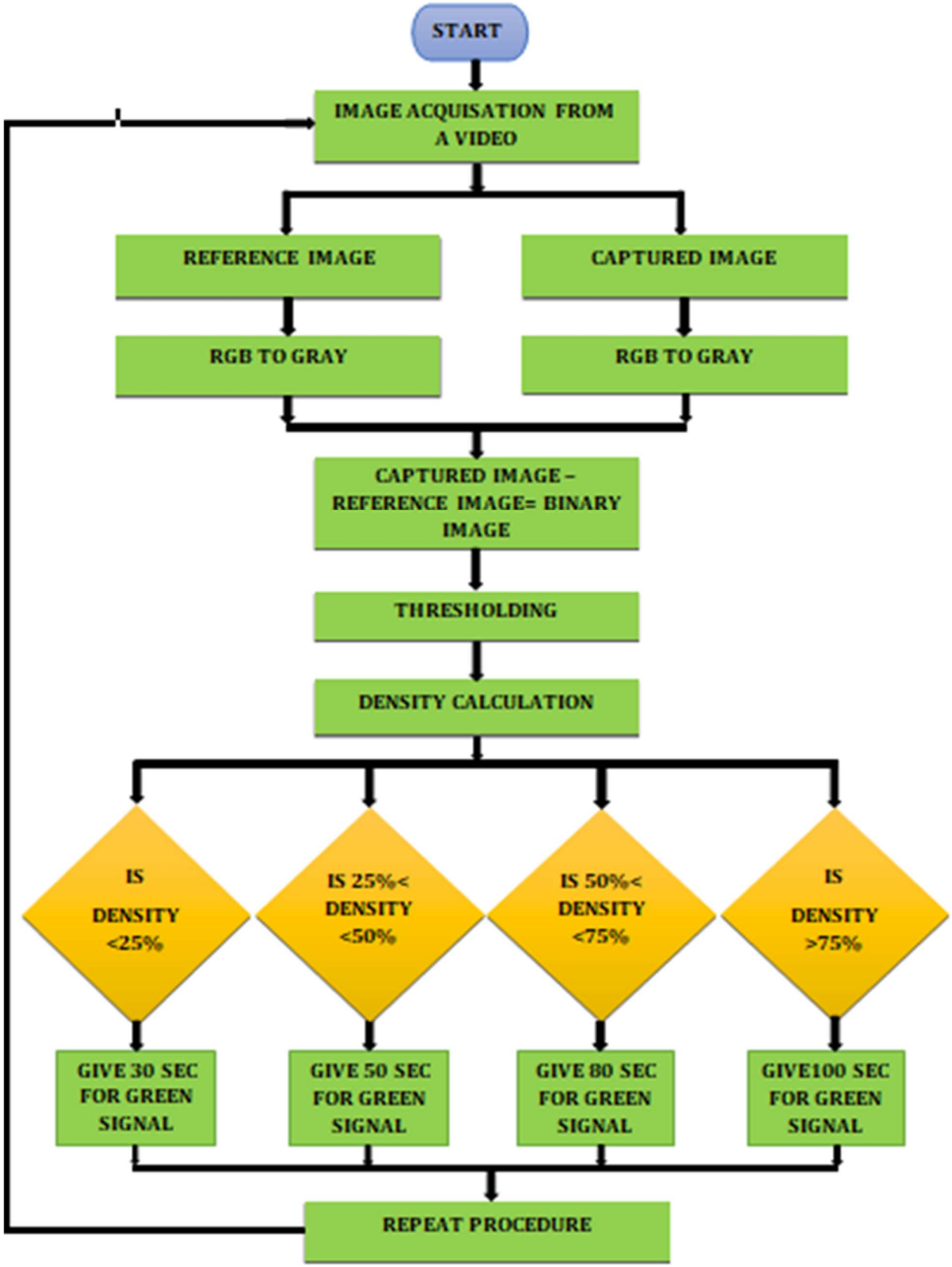


Fig-1: Block Diagram of Proposed System

In our system camera is dynamically capturing entire top view of the four lanes and sending it to the Scathe's captured images are then compared to a reference image of the empty road by image matching process. Canny edge detection method is used to determine the boundaries of the images. Here, the traffic is controlled depending on the percentage of match between the images. PC will process the received images using MATLAB software and gives the information about the traffic density present in each lane. PC will send this information from processed image to the microcontroller using USB to TTL cable. USB to TTL is used for serial communication between PC and microcontroller. The information taken from PC is converted into Binary form followed by calculation of Density. Microcontroller will control the Traffic signal lights.


START
IMAGE ACQUISATION FROM A VIDEO
REFERENCE IMAGE
CAPTURED IMAGE
RGB TO GRAY
RGB TO GRAY
CAPTURED IMAGE – REFERENCE IMAGE= BINARY IMAGE
THRESHOLDING
DENSITY CALCULATION
IS DENSITY <25%
IS 25%< DENSITY <50%
IS 50%< DENSITY <75%
IS DENSITY >75%
GIVE 30 SEC FOR GREEN SIGNAL
GIVE 50 SEC FOR GREEN SIGNAL
GIVE 80 SEC FOR GREEN SIGNAL
GIVE100 SEC FOR GREEN SIGNAL
REPEAT PROCEDURE

3.2 Working Description

The proposed system is implemented in MATLAB with an objective to reduce the traffic based on density. Four main steps are considered for the system: a) image acquisition b) RGB to grayscale transformation c) image enhancement and d) morphological operations.

A camera is installed and used to capture video of the highway. The video is recorded continuously in consecutive frames and each frame is compared to the initial captured image. The total number of cars present in the video is found out using image processing algorithms. If the total number of cars exceeds a predefined threshold, heavy traffic status is displayed as a message.

A. Image Acquisition The first step in the process of image processing is the acquisition of the image. Generally, an image is considered to be a two-dimensional function $f(\boldsymbol{a}, \boldsymbol{b})$ where a and b are spatial coordinates The value of the function at any point is known as the intensity, also known as the gray level of the image at that point. These a and b values must be converted to finite discrete values in order to form a digital image which is necessary to process through a digital computer. Each digital image is composed of pixels which are finite elements. A webcam is used for capturing video and frames are extracted to obtain images. The intensity values are proportional to the radiated energy by a physical source. Hence pixel values must be nonzero and finite. $0 < f(\boldsymbol{a}, \boldsymbol{b}) < \infty$.

B. RGB to Gray Conversion: The color images are in RGB format. In grayscale images, each pixel is represented using 8 bits and pixel values are represented using 256 levels varying from 0 to 255. The greyscale values are obtained as a weighted average of the individual R, G and B components as in (2). .0.3R + 0.59G + 0.11B

C. Image Enhancement: Image enhancement is the process of adjusting the pixel values of an image either in the spatial domain or in the frequency domain to improve the visual perception of the captured image. Image enhancement tools in MATLAB are used to obtain the grayscale version of the captured image with proper contrast and better quality. Image enhancement techniques used in the proposed method include noise removal using Wiener filter, Blob analysis and dilation.

D. Thresholding Image: Thresholding is a simple and effective method to differentiate an image into foreground and background. It's a segmentation process used to isolate objects from the background. If the histogram of the image is bimodal, a single global threshold can be used for segmentation. Automatic determination of the threshold value for each captured traffic image is done using algorithm as discussed below. 1. Select an initial estimate for the global threshold value T. 2. Segment the image using threshold T. This will produce two groups of pixels, P1 consisting of all pixels with intensity values > T, and P2 consisting of all pixels with values ≤ T. 3. Compute the average (mean) intensity values a1 and a2 for the pixels in P1 and P2 respectively. 4. Compute a new threshold value: New = 12 ($a1+a2$)5. Repeat steps 2 to 4 until the difference between the two values of threshold in successive iterations is smaller than a predefined parameter ΔT.

E. Foreground Detection: The aim of foreground detection is to detect changes occurring in the image sequences. Foreground detection is done to separate these changes taking place in the foreground from the background. All detection techniques are based on setting a reference background image and detecting changes which occur in the other images with respect to the reference image. Defining the background becomes challenging when the image contains shapes, shadows, and moving objects. A good foreground detection system must be able to develop a good background model and be robust to changes in lighting, repetitive movements (leaves, waves, shadows), and long-term changes. Here the foreground is detected using the automatically generated threshold value found out.

I. Vehicle Counting: There are many methods presently in use to detect vehicles on road such as motion detectors, installation of lasers on both sides of the road, etc., which increases the hardware requirements. The count of vehicles found can be used for controlling the traffic signal. To count the numbers of vehicles, two input images are given, one of the blank roads and the other with the vehicles on the road. The input image is then converted from RGB to greyscale. Now the two images are compared and the difference is taken using background subtraction and the difference image is converted into binary form. The blobs in the binary image are opened only when the blob area is greater than 2000. Using MATLAB, the exact number of vehicles, are determined and the count is displayed using a seven-segment display.

F. Time Allocation: Once the number of vehicles is found out, then time allocation is done based on the count. Time allocation is indicated using LEDs which are connected to our circuit. Also, a Seven Segment Display screen is attached to display the time allotted. When the count of cars is greater than 1 and less than 5, the timer shows 5 seconds and LED blinks for 5 seconds. When the count of cars is greater than 5 and less than 10, then the timer shows 10 seconds and LED blinks for 10 seconds.

3.3 Hardware Description

Hardware Module:

A USB based web camera to capture images of the traffic on road.

Arduino board: Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.: used to control LEDs representing the red and green lights. Arduino Uno Features

This board comes with all the features required to run the controller and can be directly connected to the computer through USB cable that is used to transfer the code to the controller using IDE (Integrated Development Environment) software, mainly developed to program Arduino. The features of Arduino Uno.

A timer module is used to display the remaining time.

3.4 Software Description:

SYSTEM REQUIREMENTS

500 GB free space and Min. 4 GB RAM

MATLAB

MATLAB is a programming platform designed specifically for engineers and scientists to analyze and design systems and products that transform our world.: version R2016a is used as the image processing software which comprises specialized modules that perform specific tasks. MATLAB coding is completed using the reference and captured images.

It has numerous built-in commands and math's functions that help in mathematical calculations, generating plots, and performing numerical methods.

Chapter 4 - Design

4. Design

4.1 Technology Selection

1. MATLAB for Image Processing

MATLAB is a good interactive tool in image processing and it has a wide area of application and specification. MATLAB provide a platform in which we can easily implement the algorithm that will perform the desired operation. MATLAB has several tools that are used to enhance the image according to our desires. MATLAB is used in two types of applications.

The first one is to enhance the visual information like sharpness, contrast, and brightness enhancement to make the image better perspective vision. In the Second one, we improve the image in such a way that it is suitable for the machine application. In this paper, we describe the important feature of MATLAB TOOL and its application for image processing.

MATLAB here is used to create frames of the video into two types of frames,

I. Reference Image

II. Current Working Image

The reference image is an image without any moving vehicles on road, generally kept on changing for the development or production level system of our application and the current working image frame which consists of the current frame without any distortions.

Due to MATLAB's ability to let the Image Processing tool be developed without huge lines of code and its library and cross-platform support for changing hardware ports and registers value with great ease using MATLAB packages is extremely useful.

MATLAB tool is used for the data analysis and enhancing the visualization of the image. This tool has been designed with powerful support for matrices and their operations. This tool has an excellent solution for graphics applications, and it also has powerful

programming language support. All these capabilities make the MATLAB tool very important and efficient for image processing purposes. MATLAB tool has much utility software inbuilt. This software is called the toolbox. I have to use the image processing toolbox in MATLAB. These tools are designed in such a way that they can perform a particular task. So basically, it is a group of software which is designed to perform multi-task which is related to image processing, mathematics, Simulink etc. particular tasks.

Here I am not going to describe all the tools boxes but I will describe the main application of the image processing toolbox. I will describe the introduced functions, MATLAB commands and image-processing techniques which I have used. MATLAB function accepts the various parameters which have been given as the input and after processing it produces some sort of output. We can give a matrix, a string, a graph or a figure as the input. Then we provide the processing functions or commands such as imread, imclose etc.

2. MATLAB for Arduino

Because MATLAB is a high-level interpreted language, prototyping and refining algorithms for your Arduino projects is easy, you can see results from I/O instructions immediately, without recompiling.

MATLAB's package for Arduino is powerful enough to use any development board and start building the logic for its development.

MATLAB Support Package for Arduino® Hardware enables you to use MATLAB to interactively communicate with an Arduino board.

For instance, you can read and write sensor data through the Arduino board and immediately see the results in MATLAB without compiling any code. Further, you can utilize thousands of built-in math, engineering, and plotting functions that are included with MATLAB to analyze and visualize data collected from your Arduino.

You can also create and distribute compiled standalone MATLAB applications that run MATLAB programs on systems to interface Arduino hardware over Serial without an installed version of MATLAB.

3. Arduino Uno Development Board

Arduino UNO is based on an ATmega328P microcontroller. It is easy to use compared to other boards, such as the Arduino Mega board, etc. The board consists of digital and analog Input/Output pins (I/O), shields, and other circuits.

The Arduino UNO includes 6 analog pin inputs, 14 digital pins, a USB connector, a power jack, and an ICSP (In-Circuit Serial Programming) header. It is programmed based on IDE, which stands for Integrated Development Environment. It can run on both online and offline platforms.

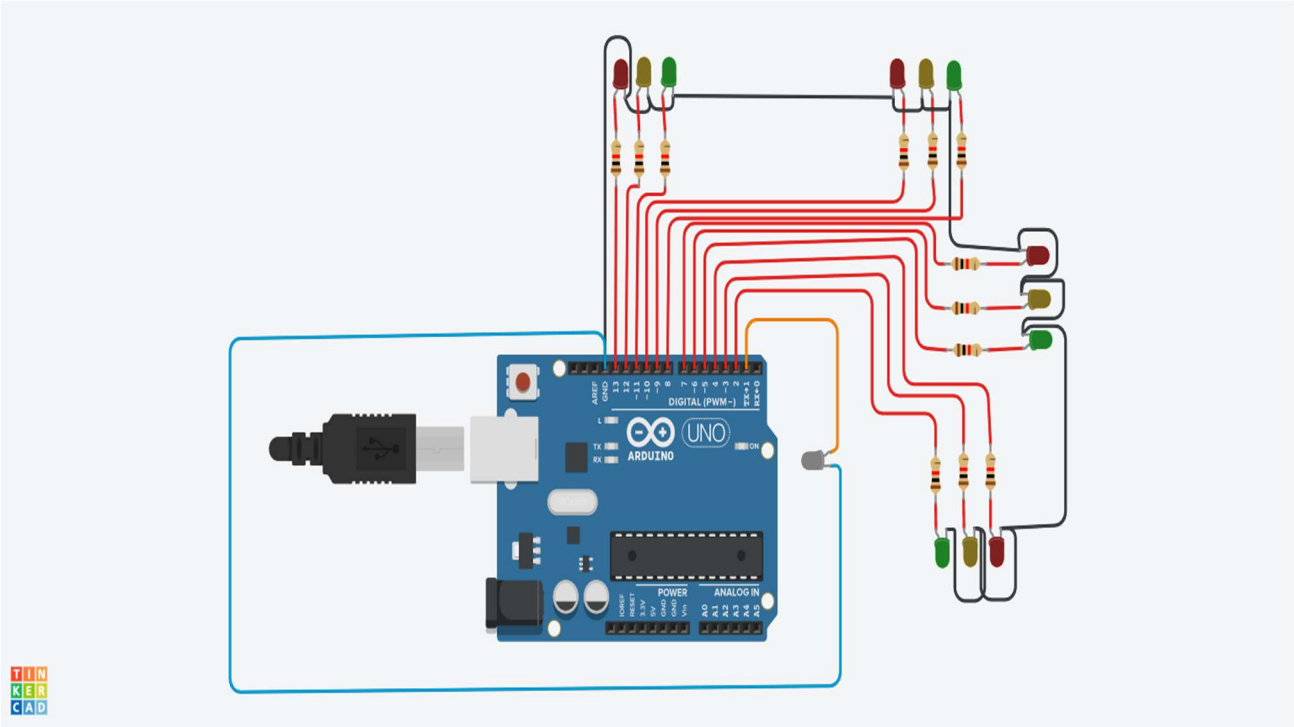
Arduino Uno board is used in this project in order to control the traffic lights represented by 3 mm LEDs (Red, Yellow, Green), because the development would be fast and highly interactive by using MATLAB with the same.

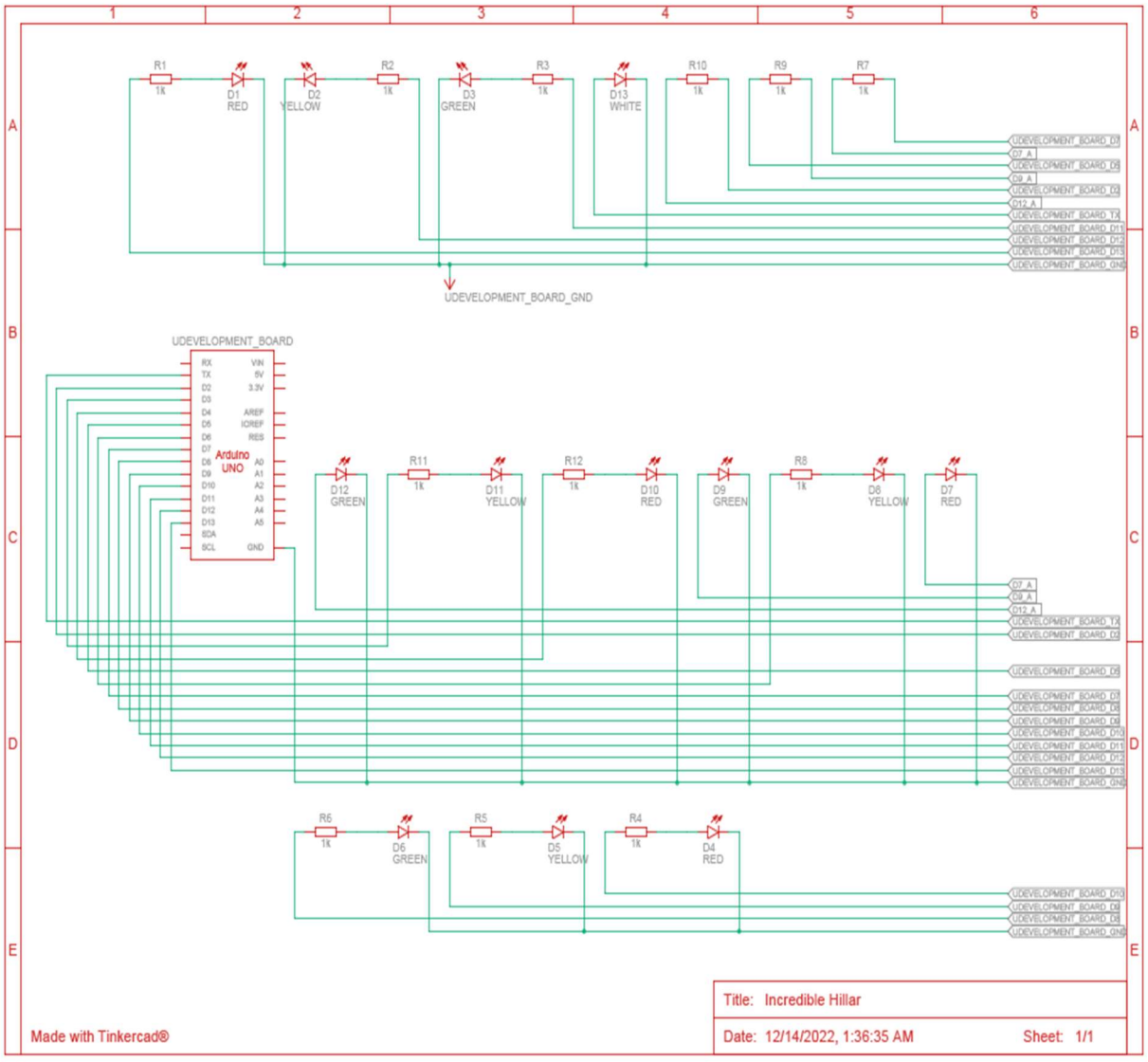
4. Tinker Cad

Tinker Cad circuit designing system is used by us to develop the circuit in graphical layout and simulate the same using C programming language for testing the working Arduino Uno board and connection.

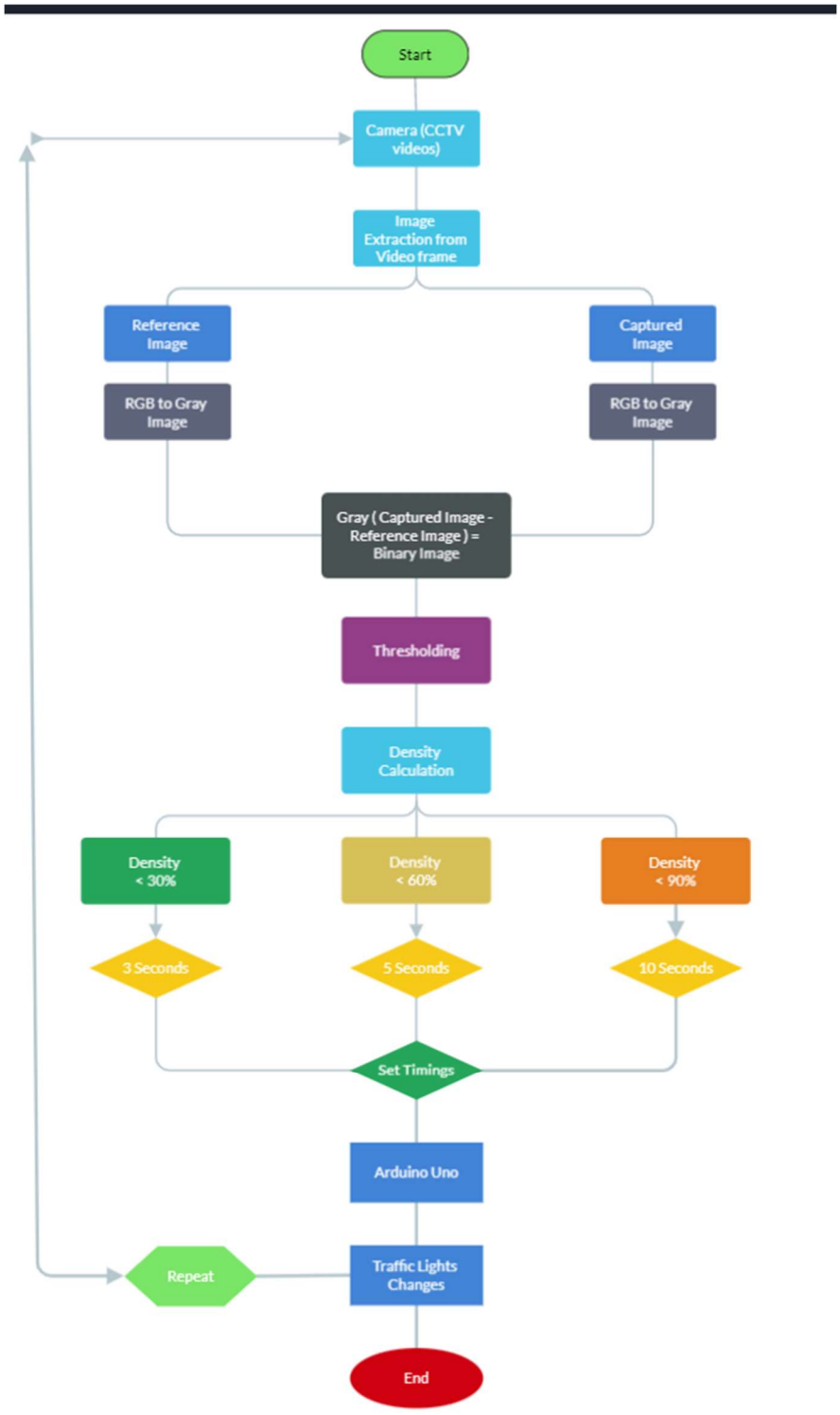
Tinker cad Circuits is a powerful feature where you can practice building circuits and programming. The supported microcontrollers are the Arduino Uno and Micro: Bit, so you can code in block-based or text-based formats interchangeably. If you'd like to practice both, Tinker cad can even translate one format to another, allowing you to toggle between both views.

4.2 Circuit diagrams for different sections




R1
1k
D1
RED
D2
YELLOW
R2
1k
D3
GREEN
R3
1k
D13
WHITE
R10
1k
R9
1k
R7
1k
UDEVELOPMENT_BOARD_GND
UDEVELOPMENT_BOARD
Arduino
UNO
D12
GREEN
R11
1k
D11
YELLOW
R12
1k
D10
RED
D9
GREEN
R8
1k
D8
YELLOW
D7
RED
R6
1k
D6
GREEN
R5
1k
D5
YELLOW
R4
1k
D4
RED
Made with Tinkercad®
Title: Incredible Hillar
Date: 12/14/2022, 1:36:35 AM
Sheet: 1/1

4.3 Program flow diagram with algorithm



Program flow diagram

Algorithm:

1. Initially the Image is captured from the video (for prototype, saved video) from which each frame is extracted.
2. The image is compared with the reference image, i.e., an image without any moving traffic, by converting them to their gray form from RGB.
3. The Gray Images then subtracted from each other, and the remaining part becomes our Binary image.
4. This binary image with thresholding then converted into Black and White image, where there is moving object present becomes white and remaining becomes black.
5. Then the white components act as connected components in Black and White image matrix then counted, which refers to the number of vehicles present in the lane.
6. The data of n number of lanes then calculated using the same above steps.
7. The average density of each lane is then calculated using

$$Avg.Density\ of\ a\ lane = \frac{Total\ Vehicles\ present\ in\ an\ image\ frame\ for\ lane}{\Sigma\ (Vehicles\ present\ in\ every\ lane)}$$

8. Then converted into percentages with taking floor values (Rounded off to nearest number).
9. Then the timings are set accordingly by multiplying the Avg Density to CONSTANT TIME*.
10. This is repeated for the whole day or done in a loop continuously.

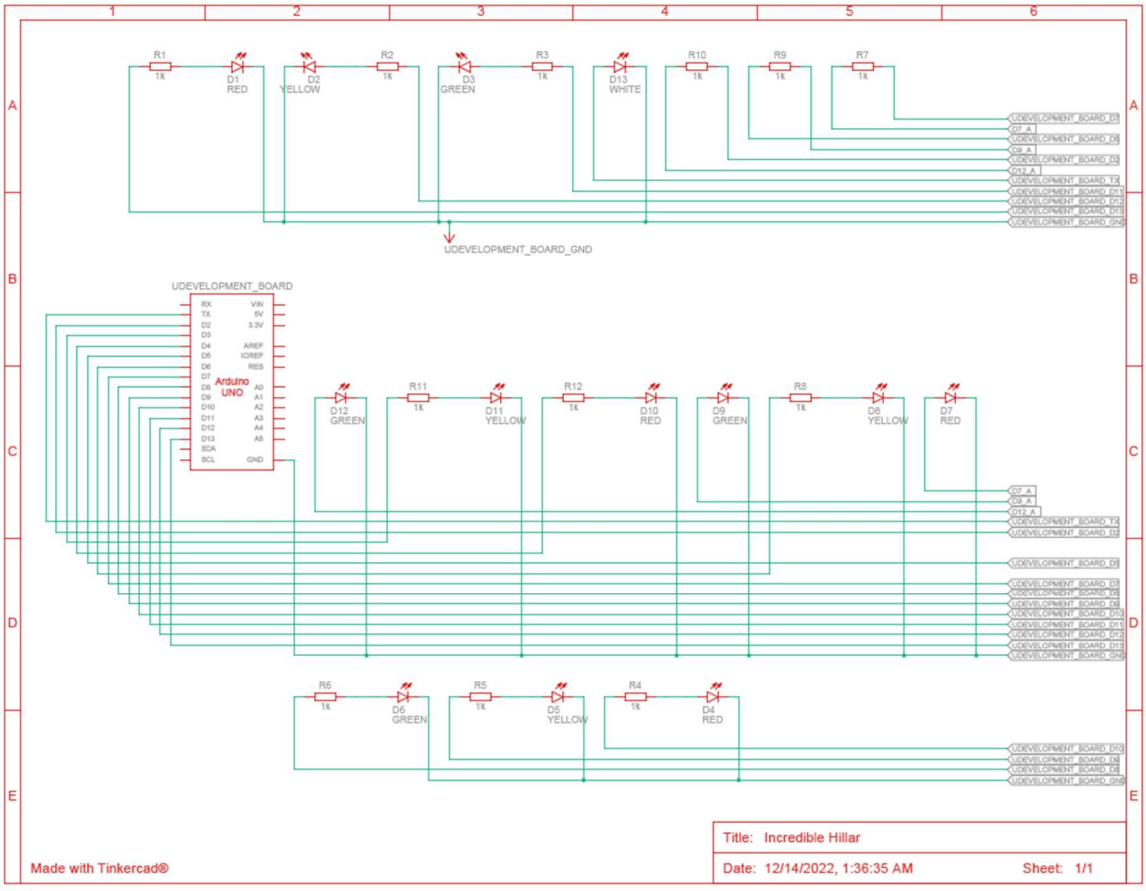
*: *The time which is preset for every lane to be open for that time MAXIMUM*

Chapter 5 - Implementation and Testing

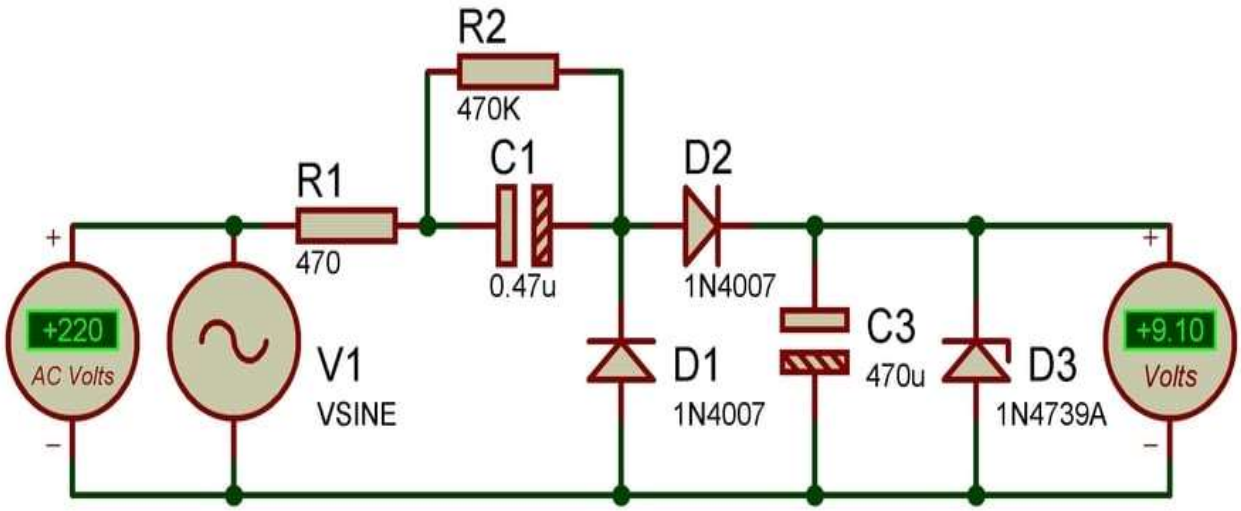
Chapter 5 Implementation and Testing

5.1 Circuit layout for different modules

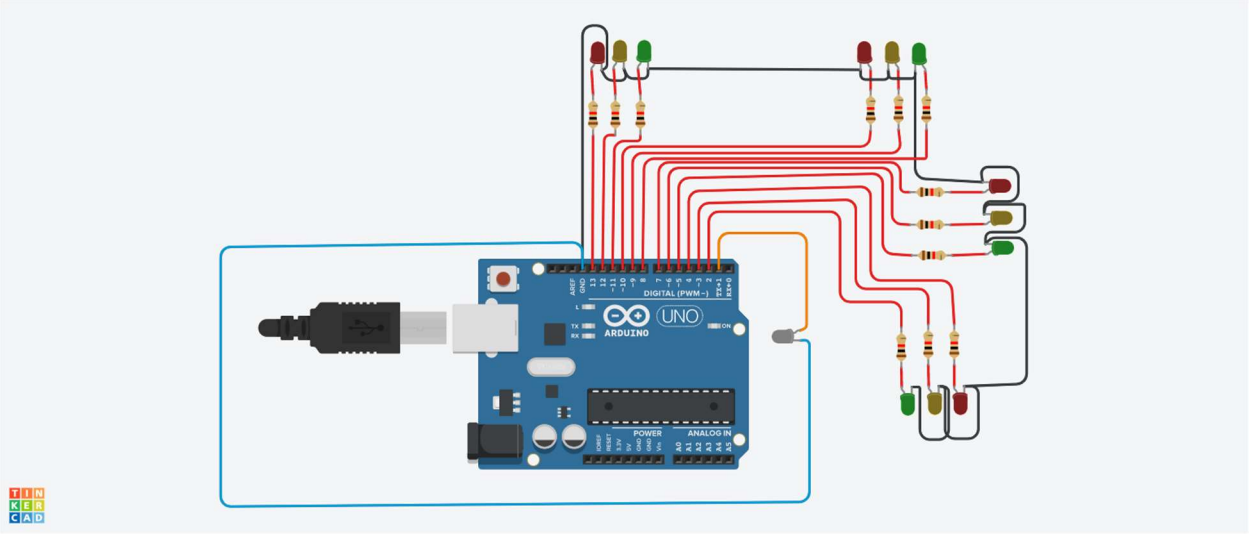
1. Arduino - Computer Module



2. DC power supply module (Not to be used in Prototype)



5.2 PCB layout of different sections



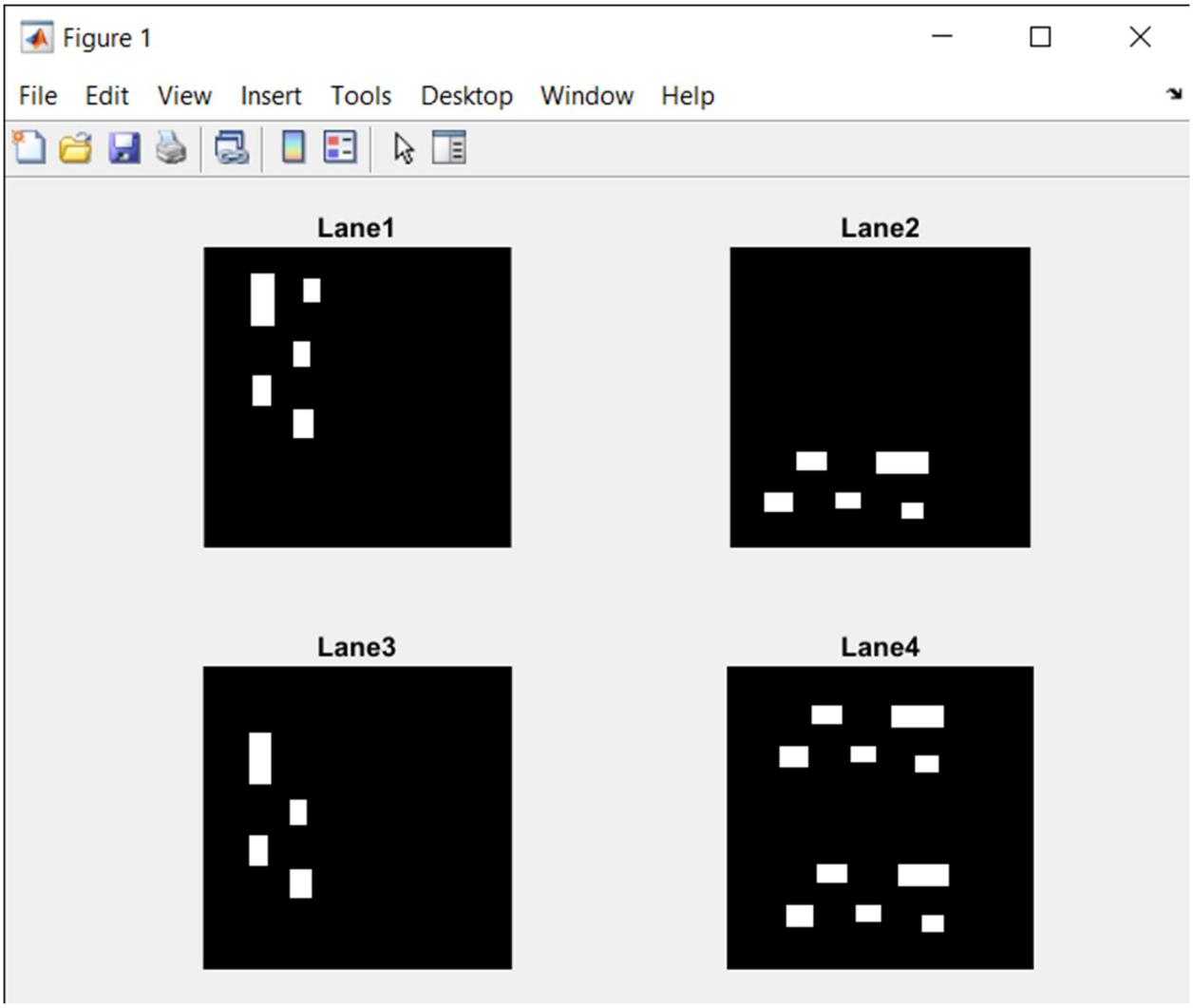
5.3 Test Cases

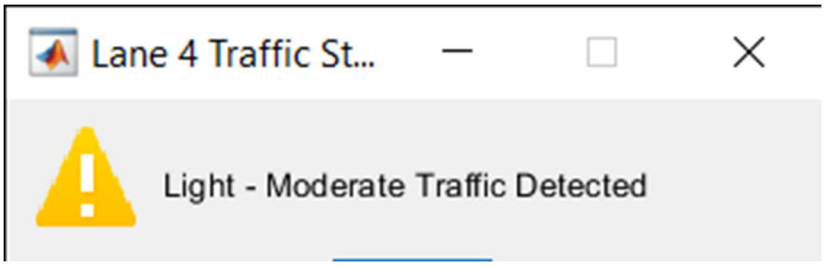
In four-lane square traffic, the number of vehicles present in each lane is counted as given below **lane 1** has **5** vehicles, **lane 2** has **5** vehicles, **lane 3** has **4** vehicles and **lane 4** has **10** vehicles hence the density of each lane calculated will be:

1. Calculating the Average density of each lane

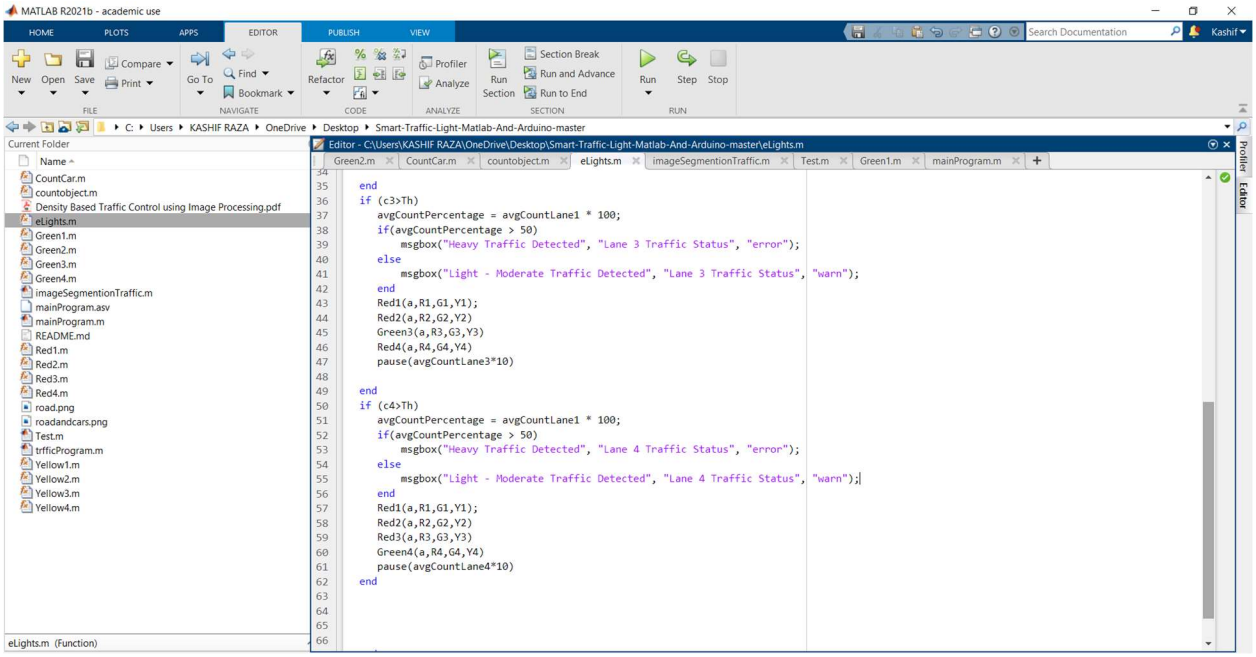
Avg Density of L1 = 5 / 24	=	0.208
Avg Density of L2 = 5 / 24	=	0.208
Avg Density of L3 = 4 / 24	=	0.166
Avg Density of L4 = 10 / 24	=	0.416
Density %age of L1	=	20% (3 seconds)
Density %age of L2	=	20% (3 seconds)
Density %age of L3	=	17% (3 seconds)
Density %age of L4	=	42% (3 seconds)

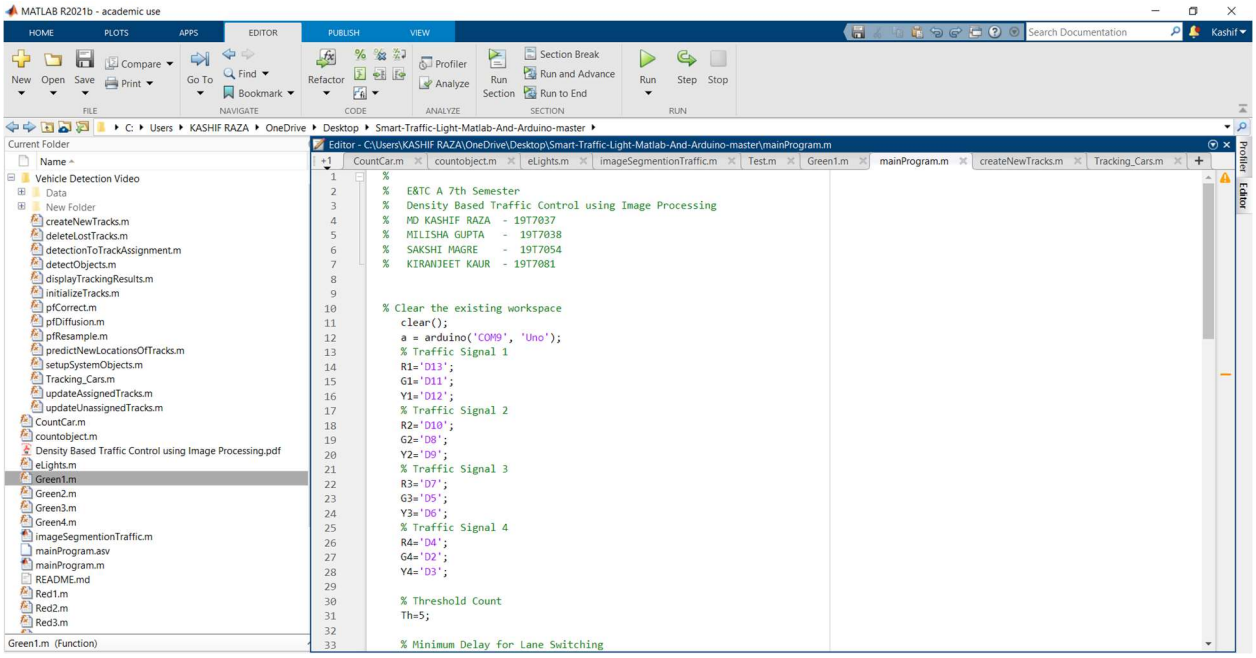
For this case, a signal will be green (open) for 3 seconds and then closed after a pause of 3 - 5 seconds during lane switching.





The images (snapshots above) depict the black and white objects detected as a vehicle using image processing with the description of Lane's traffic status.


MATLAB R2021b - academic use
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Kashif
New
Open
Save
Compare
Print
Go To
Find
Bookmark
Refactor
Profiler
Analyze
Run Section
Section Break
Run and Advance
Run to End
Run
Step
Stop
FILE
NAVIGATE
CODE
ANALYZE
SECTION
RUN
C: ▸ Users ▸ KASHIF RAZA ▸ OneDrive ▸ Desktop ▸ Smart-Traffic-Light-Matlab-And-Arduino-master
Current Folder
Name
CountCar.m
countobject.m
Density Based Traffic Control using Image Processing.pdf
eLights.m
Green1.m
Green2.m
Green3.m
Green4.m
imageSegmentionTraffic.m
mainProgram.asv
mainProgram.m
README.md
Red1.m
Red2.m
Red3.m
Red4.m
road.png
roadandcars.png
Test.m
trfficProgram.m
Yellow1.m
Yellow2.m
Yellow3.m
Yellow4.m
eLights.m (Function)
Editor - C:\Users\KASHIF RAZA\OneDrive\Desktop\Smart-Traffic-Light-Matlab-And-Arduino-master\eLights.m
Green2.m
CountCar.m
countobject.m
eLights.m
imageSegmentionTraffic.m
Test.m
Green1.m
mainProgram.m
end
if (c3>Th)
avgCountPercentage = avgCountLane1 * 100;
if(avgCountPercentage > 50)
msgbox("Heavy Traffic Detected", "Lane 3 Traffic Status", "error");
else
msgbox("Light - Moderate Traffic Detected", "Lane 3 Traffic Status", "warn");
end
Red1(a,R1,G1,Y1);
Red2(a,R2,G2,Y2)
Green3(a,R3,G3,Y3)
Red4(a,R4,G4,Y4)
pause(avgCountLane3*10)
end
if (c4>Th)
avgCountPercentage = avgCountLane1 * 100;
if(avgCountPercentage > 50)
msgbox("Heavy Traffic Detected", "Lane 4 Traffic Status", "error");
else
msgbox("Light - Moderate Traffic Detected", "Lane 4 Traffic Status", "warn");
end
Red1(a,R1,G1,Y1);
Red2(a,R2,G2,Y2)
Red3(a,R3,G3,Y3)
Green4(a,R4,G4,Y4)
pause(avgCountLane4*10)
end
Profiler
Editor


MATLAB R2021b - academic use
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Kashif
New
Open
Save
Compare
Print
Go To
Find
Bookmark
Refactor
Profiler
Analyze
Run Section
Section Break
Run and Advance
Run to End
Run
Step
Stop
FILE
NAVIGATE
CODE
ANALYZE
SECTION
RUN
C: ▸ Users ▸ KASHIF RAZA ▸ OneDrive ▸ Desktop ▸ Smart-Traffic-Light-Matlab-And-Arduino-master ▸
Current Folder
Name
Vehicle Detection Video
Data
New Folder
createNewTracks.m
deleteLostTracks.m
detectionToTrackAssignment.m
detectObjects.m
displayTrackingResults.m
initializeTracks.m
pfCorrect.m
pfDiffusion.m
pfResample.m
predictNewLocationsOfTracks.m
setupSystemObjects.m
Tracking_Cars.m
updateAssignedTracks.m
updateUnassignedTracks.m
CountCar.m
countobject.m
Density Based Traffic Control using Image Processing.pdf
eLights.m
Green1.m
Green2.m
Green3.m
Green4.m
imageSegmentionTraffic.m
mainProgram.asv
mainProgram.m
README.md
Red1.m
Red2.m
Red3.m
Green1.m (Function)
Editor - C:\Users\KASHIF RAZA\OneDrive\Desktop\Smart-Traffic-Light-Matlab-And-Arduino-master\mainProgram.m
CountCar.m
countobject.m
eLights.m
imageSegmentionTraffic.m
Test.m
Green1.m
mainProgram.m
createNewTracks.m
Tracking_Cars.m
%
% E&TC A 7th Semester
% Density Based Traffic Control using Image Processing
% MD KASHIF RAZA - 19T7037
% MILISHA GUPTA - 19T7038
% SAKSHI MAGRE - 19T7054
% KIRANJEET KAUR - 19T7081
% Clear the existing workspace
clear();
a = arduino('COM9', 'Uno');
% Traffic Signal 1
R1='D13';
G1='D11';
Y1='D12';
% Traffic Signal 2
R2='D10';
G2='D8';
Y2='D9';
% Traffic Signal 3
R3='D7';
G3='D5';
Y3='D6';
% Traffic Signal 4
R4='D4';
G4='D2';
Y4='D3';
% Threshold Count
Th=5;
% Minimum Delay for Lane Switching
Profiler
Editor

6. Conclusion

Conclusion

Thus, the system of Density based Traffic Control System using Image Processing has been successfully implemented. A microcontroller, ARDUINO MEGA was used to control the traffic signal system and two seven segment displays. Production costs are low while achieving high speed and accuracy. This density-based image processing technique removes limitations of traditional signal controlling systems.

An efficient density-based traffic control system is simulated and implemented which provides a good traffic control mechanism without time wastage. It is also a much better way of detecting the presence of vehicles on the road since it makes use of image data. So it surely operates much better than systems which rely on the metal content of the vehicles to detect their presence. Image processing techniques overcome the limitations of the all the traditional methods of traffic control. It eliminates the need for extra hardware and sensors. The use of multiple cameras will help to analyze and control traffic in a particular region. The proposed system outperforms the existing system in terms of accuracy and simplicity.

Learning

From this project we are introduced to various image processing techniques and using various MATLAB functions for processing the image i.e. taking the image, cropping it, converting to gray image, then to black and white, and enhancing it so that we can count the number of vehicles in a lane.

Future Scope

The weather conditions are not taken into account which may affect the image quality when it becomes foggy or in heavy rains. More advancements can be made to the proposed system to check identification of vehicles that pass through the system circle which could help in traffic surveillance.

Image processing technology extracts information from images and integrates it for wide range of applications. Here we have outlined one of the most prominent fields where image processing could bring significant benefits. A density-based image processing system will remove all the barriers which were placed by traditional timer or sensor-based systems.

REFERENCES

[1] Stormy Attaway, MATLAB: A Practical Introduction to Programming and Problem-Solving Paperback, Fifth Edition, Butterworth-Heinemann, 2018

[2] Ralph Gonzalez, Richard Woods, Steven Eddings, Digital Image Processing Using MATLAB, Second edition, McGraw Hill Education, 2017

[3] Ms. Pallavi Khedekar, Ms. Sayanti Banerjee, Prof. Makua, “Real Time Traffic Light Control Using Image Processing” UP,India.

[4] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis, "A survey of video processing techniques for traffic applications,"Image and Vision Computing, vol. 21, pp. 359-381, Apr 1,2003.

[5] Omkar Ramdas Gaikwad, Anil Vishwasrao, Prof.Kanchan Pujari, Tejas Talathi, “Image Processing Based Traffic Light Control”.

[6] Pejman Niksaz, "Automatic Traffic Estimation Using Image Processing" Azad University of Yazd, Iran.

[7] https://in.mathworks.com/ MATLAB

[8] https://in.mathworks.com/help/matlab/ , MATLAB documentation

[9] https://www.arduino.cc/ Arduino uno documentation

[10] https://www.tinkercad.com/ Tinker cad designing

[11] https://www.ti.com/tool/TINA-TI TINA software circuit designing

APPENDIX

APPENDIX

1. **AES** - Advanced Encryption Standard
2. **API** - Application Programming Interface
3. **GUI** - Graphical User Interface
4. **IDE** - Integrated Development Environment
5. **SIMULINK** - Simulation and Model-Based Design
6. **TBC** - To Be Confirmed
7. **MEX** - MATLAB Executable
8. **MCC** - MATLAB Compiler
9. **MLINT** - MATLAB Linter
10. **MLAPP** - MATLAB App Designer
11. **MIP** - Medical Image Processing
12. **DIP** - Digital Image Processing
13. **ROI** - Region of Interest
14. **GLCM** - Gray Level Co-occurrence Matrix
15. **SAR** - Synthetic Aperture Radar
16. **EKF** - Extended Kalman Filter
17. **PID** - Proportional-Integral-Derivative
18. **HMI** - Human Machine Interface
19. **RTOS** - Real-Time Operating System
20. **UART** - Universal Asynchronous Receiver/Transmitter
21. **PCB -** Printed Circuit Board
22. **FPS -** Frames per second