MULUNGUSHI UNIVERSITY

SCHOOL OF SCIENCE, ENGINEERING, AND TECHNOLOGY

(SSET)

DEPARTMENT OF COMPUTER SCIENCE AND IT

PROJECT REPORT TITLE: DENSITY BASED TRAFFIC CONTROL SYSTEM

NAME: MAZUBA MALAMBO

STUDENT ID: 201701671

COURSE: ICT431 CAPSTONE PROJECT

PROGRAMME: BScCSIV

SUPERVISOR: MR. L . SIMUKONDA

THIS REPORT IS SUBMITTED IN PARTIAL FULFILMENT FOR THE AWARD OF BACHELOR OF COMPUTER SCIENCE FOR THE 2020/2021 ACADEMIC YEAR

s

Table of Contents

DECLARATION

I, Mazuba Malambo the undersigned here, declare that the Density Based Traffic Control System is my own work and to that end, this work has not been submitted to any other University, be it Local or International, for the purpose of attaining a degree or for an examination. In instances where I have used or quoted external material, these sources have been accredited and indicated in the complete reference list. I fully understand that the use of external material without giving due credit to the source or author may and will be considered plagiarism.

Author: Mazuba Malambo
Date:
Signature……………………….

Supervisor: Mr. Luckson Simukonda. School of Science Engineering and Technology.
Date:
Signature…………………………

DEDICATION

I would like to thank my supervisor (Mr L. Simukonda) for the knowledge patience and guidance conveyed to me through the conception of the project and the time taken to help me clearly bring my document to completion.

To my parents Mr. Marianos M. Malambo and Mrs. Patricia K. Masule-Malambo, for all the love, care, financial, mental, physical and spiritual support, and dedication rendered to me in my upbringing and in supporting and encouraging me to peruse greatness. Thank you both for giving me the strength to reach for the stars and pursue my dreams. There are no words to express how grateful I am to you both.

To all my friends, thank you for your motivation, understanding and encouragement in my many, many moments of crisis. Your invaluable friendship makes my life a wonderful experience.

Above all, I thank God Almighty for the charitable time, strength, inspiration, and always being there for me.

ACKNOWLEDGEMENT

I would like to acknowledge my dissertation Supervisor Mr. Simukonda, who has been an invaluable mentor. His enduring encouragement and practical advice have been an inestimable source of support for me during this process. I would also like to thank all lecturers in the School of Science Engineering and Technology for the excellent support and Excellent knowledge delivery that they have rendered unto me.

ABSTRACT

Traffic congestion especially at road intersections is becoming an issue for which road traffic users contend daily. The conventional traffic light applies a fixed logic of allocating equal “GO” time to lanes of traffic at road intersections irrespective of the density of traffic on each lane. This is a serious problem in most cities and therefore an automated system with decision-making capabilities would work best to help replace the manual mode of a fixed timer. The present-day traffic signaling system is based on fixed time which may render inefficient when one lane is operational than the others. To optimize this problem a density-based traffic control system can be implemented. Sometimes higher traffic density at one side of the junction demands longer green time as compared to the standard allotted time. Therefore, a system in which the period of green light and red light is assigned based on the density of the traffic present at that time can be implemented.

The proposed system will use cameras, which will detect the traffic density via image processing and change the junction timing automatically to accommodate the movement of vehicles smoothly avoiding unnecessary waiting time at the junction. The density of the vehicles is measured in three zones i.e., low, medium, high based on which zone is detected, the timings are allotted accordingly. Once the density is calculated, the glowing time of green lights assigned accordingly. The cameras which are present on the road will detect the presence of the vehicles and send the information to the system where it will decide how long a flank will be open or when to change over the signal lights.

Keywords: Traffic Density, Traffic Light, ‘+’ Road Intersection.

LIST OF FIGURES

LIST OF TABLES

ACRONYMS AND ABBREVIATIONS

Table 1 List of Acronyms and Abbrieviations

AQL	Average Queue Length
AWT	Average Wait Time
BS	Base Station
COCOMO	Constructive Costing Model
I/O	Input/Output
IR	InfraRed
IT	Information Technology
KLOC	Kilo Lines Of Code
MATLAB	Matrix Laboratory
MCU	Microcontroller Unit
RFID	Radio Frequency Identification
RSU	Road Side Unit
SDLC	Software Development Life Cycle
SUMO	Simulation of Urban Mobility
TSN	Traffic Sensor Nodes
WSN	Wireless Sensory Networks

CHAPTER 1 - INTRODUCTION

1.1 Introduction

The constant drive of people moving from rural to urban areas in search of greener pastures has brought about an urban population explosion and over-stretched infrastructure. An example of an over-stretched infrastructure is the road, a situation that has culminated in increased traffic. Although traffic lights have always been used for controlling the flow of traffic, traffic management in major cities around the world has continued to be a subject of concern. Quick transportation systems inclusive of rapid transit systems are the nerves of economic development.

Traffic mismanagement, as well as traffic congestion, results in long waiting times, loss of fuel, and money. Traffic congestion or traffic snarl-up is now considered to be one of the largest problems in urban settlements (Udoakah & Okure, 2018). Traffic problems are widely increasing as an expected result of the growing number of transportations means and current low-quality infrastructure of the roads.

With the drastic increase in the population of our cities as well as the busyness of our roads and the need to utilize one maximum time, there arises a need for a more dynamic system that seeks to reduce unnecessary congestion as well as unnecessary wait times at traffic intersections. It would be best that a new design is developed to improve the flow and control of traffic based on the density of vehicles on a particular lane by automatically assigning the left-over time in an idle lane to the lane with more traffic density, hence, reducing traffic congestion in the lanes. Studies have been carried out to improve the current traffic control system which is the fixed timer because it is rendered inaccurate or inefficient when the is more traffic in one lane compared to the other (Panjwani et al., 2014).

Traffic congestion may come about due to heavy traffic at an intersection. To avoid traffic congestion, there are so many traffic management techniques available. But there is no technique that is perfect by itself as the real-time situations are constantly changing and the system has to adapt itself to change in the continuously changing circumstances.

The main aim of designing and developing the Density-based traffic control system is to reduce the waiting time of each lane of the cars and also to maximize the total number of cars that can cross an intersection. In this system, the time assigned to the traffic light of a particular lane is based on the current traffic density on the road.

1.2 Problem Statement

The present issue is that the current system of traffic lights uses a fixed allotted time and as such regardless of the traffic condition at an intersection, each lane is given equal time. The current system in place also causes unnecessary wait times at traffic intersections in an event that there is only one car and the rest of the lanes are empty. This disrupts the smooth flow of traffic and adds to the already existing issue of traffic congestion.

1.3 Aim

To develop a Density Based Traffic Control System that uses image processing to help reduce traffic congestion

1.4 Objectives

The main objectives of the system are:

- Use image processing to detect the density of traffic.
- Control traffic lights timer based on the density of traffic.
- Reduce wait times at traffic intersections

1.5 Project Scope

The scope of this project is to develop a density based traffic control system with the use of image processing. In this project, a density-based traffic system will be developed to reduce traffic congestion at intersections. The system will be able to analyze traffic at intersections and then make a decision based on the amount of traffic present.

The value of traffic density measures only the ratio between the density of the vehicles and the total density of the road. So based on this measure, the traffic control system will compare different roads in the intersection and decide on the traffic light and the time interval given.

1.6 Project Justification

This proposed system is aimed at reducing traffic congestion at intersections. The current system uses a fixed time allocating equal “GO” time to lanes of traffic at road intersections regardless of the density of traffic on each lane. The current system isn’t accurate because sometimes higher traffic density at one side of the junction demands longer green time as compared to the standard allotted time. The proposed system will use a mechanism in which the period of green light and red light is assigned based on the density of the traffic present at that time and not that standard fixed allotted time.

The system will be very useful because it will also eliminate the unnecessary wait time that occurs when there is only one car at an intersection that is forced to wait for a green light instead of just proceeding because there is no impeding traffic. The proposed system will use image processing technology that will be used to verify the density of traffic at the intersection.

1.7 Conclusion

The contribution presented in this proposed system will help combat traffic congestion at road intersections. It will employ the use of image processing which will be more effective at controlling the flow of traffic as compared to the standard fixed timer system. Therefore, this system has to be developed because it will help eliminate unnecessary wait times at traffic lights as well as help improve the flow of traffic.

CHAPTER 2 – LITERATURE REVIEW

2.1 Introduction

Various Traffic Systems are used around the world to try and enhance and improve the flow of traffic. These systems are somewhat similar in the sense that they have the same aim of enhancing the flow of traffic but they also differ in the mechanisms used to control traffic. This chapter simply gives a definitive review of the components of the system and related literature. Firstly, the literature intuitively outlines the fields of study subject to the project research, then secondly, it reviews the systems that are used in the mentioned technologies and study areas, and thereafter it compares the reviewed systems and discusses the proposed system.

2.1.1 History of Traffic Lights

Traffic control began to appear in London, England in the early 18th century(Michels, 2014). As early as 1722, traffic control measures were taken to ensure the rapid movement of horse-drawn carriages, buggies, carts, and pedestrians across the London Bridge. At the time, crossing the bridge was seen as a challenge or problem because of the disorderly nature of the traffic movements. The Lord Mayor organized a coalition of three men and appointed them as public servants to monitor and regulate individuals crossing on the bridge.

Traffic control in the United Kingdom goes back to the 1860s when the city's Police Department was tasked with managing the reckless driving of horse-drawn buses within the city. This was in response to public outcry over the deaths of several pedestrians trampled by the horse-drawn buses. The Police Department assigned the tallest officers on the force to the new squad to ensure that the officers could see over carriages and pedestrians. The officers were told to point, wave, and shout to move traffic on the busy streets (Michels, 2014).

2.2 Related Literature

2.2.1 Intelligent Traffic Light Control System Using IR Sensors

This system aims to eliminate the delay on roads by reducing traffic on roads automatically via the use of an embedded system that determines traffic on each road by using sensors (Jadhav et al., 2015). IR sensors are placed on each road which will then detect the vehicle and give current traffic information on each road. The timing of the signal of the traffic light is adjusted according to the traffic level on each road. The road which has the highest level of traffic is assigned a green signal and for others, red is assigned. Shinde & Powar (2017) also proposed a similar system but instead of there being only one set of sensors, they proposed that there should be three. The three sensors range from low to medium to high-density zones. They are placed facing each other by putting an IR transmitter on one end and an IR receiver on the opposite end. Once vehicles are filled and they cross the first pair of sensors, then there will be an obstacle between transmitter and receiver and this sends a digital signal (low or high) to the microcontroller, assuming that there is low-density traffic. A microcontroller also referred to as a Microcontroller Unit (MCU) is simply an integrated circuit that is designed to control a specific operation in an embedded system. Microcontrollers work by interpreting data that it receives from I/O peripherals such as sensors using its central processor. The temporal information received is then stored in the data memory where the processor then accesses it and uses algorithms that are stored in program memory to decipher and use the incoming information. Finally, it then uses I/O peripherals in this case which are the traffic lights to communicate and enact the appropriate or required action. In this scenario, when the vehicle crosses the second sensor then it assumes medium density and for the third sensor pair, it assumes that there is high-density traffic respectively. Once the density is calculated, the Road Side Unit (RSU) makes a decision based on the time to be allotted.

2.2.2 Intelligent Traffic Density Monitoring Using RFID

Varambally, Samhita & Shahi, Prachi & Rajesh (2018) suggested that each vehicle is to be fitted with a Radio Frequency Identification tag (RFID) and RFID readers that will also be installed at the start and end of each road between two junctions. Each of these RFID readers communicates the main system. When a vehicle crosses a reader, the reader picks up the vehicle’s unique ID, along with a time-stamp from the particular RFID tag. The RFID tag contains an in-built antenna to transmit data to the RFID reader which then converts the received radio waves to a usable form of data. The received tag information is then stored in a database and can be further analyzed, processed, and used to determine the road traffic density and allot time-based on the traffic.

2.2.3 Intelligent Traffic Light Scheduling Technique Using Calendar-Based History Information

Yousef et al. (2019) Proposed that the system to be designed is to implement history-based traffic management that relies on the previous all-year traffic information to predict the flow of traffic. Hence, they referred to it as a calendar-based traffic congestion management system. The system was to record traffic history information and use it to compute the green or red times for a given direction at an intersection. The system was aimed at reducing average wait times (AWT) as well as average queue length (AQL). MATLAB and SUMO simulator was used as the development and simulation environment. MATLAB (Matrix Laboratory) is simply a proprietary programming language as well as a numerical computing environment developed by MathWorks. It allows matrix manipulations, plotting of functions and data, algorithms implementation, user interface designs, and the interfacing of programs that are written in other languages. Simulation of Urban Mobility (SUMO) is simply a non-proprietary, portable, and continuous multi-modal traffic simulation package that is used for research purposes such as traffic forecasting, traffic lights valuation, route selection, or in the field of vehicular communication systems.

2.2.4 Intelligent Traffic Light Flow Control System Using Wireless Sensors Networks

K. M. Yousef et al(2010) proposed that a traffic light control system that is based on Wireless Sensory Networks is to be designed. The system is composed of two parts: the Wireless sensory network (WSN) and a Base Station (BS) running control algorithms. The Wireless Sensory Networks consist of a group of traffic sensor nodes (TSNs) that are designed to provide the traffic communication infrastructure and to mediate easy and large deployment of traffic systems. Each traffic sensory node (TSN) is to collect and generate traffic data such as vehicle speed and vehicle length which will then be sent in real-time to the base station. The Traffic Sensory Nodes are to be installed in the roadbeds.

2.3 Review of Existing Systems

2.3.1 Manual Traffic Control Managements

Manual traffic control is one of the common intersection control strategies in which trained personnel, typically police law enforcement officers, allocate intersection right-of-way to approaching vehicles (S. Parr & Wolshon, 2015). This is the simplest form of traffic management, which mainly includes humans in the method. An example of a traffic control mechanism that is manual is a traffic policeman is standing at cross-sections of roads controlling the flow of traffic by using signs as shown in the figure below

Figure 2. 1: Traffic Officer controlling traffic flow (Kerala Police, 2019).

Another means of manual traffic control is via the use of road traffic signs and road markings. Traffic signs and road markings are used to control and provide a smooth and orderly flow of traffic. Apart from guiding traffic, Traffic Road signs and road markings also warn drivers. Below are images of some road signs and markings.

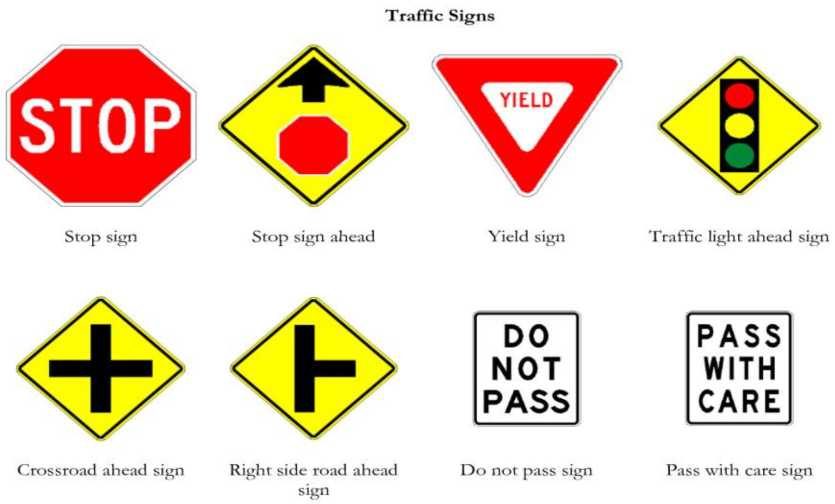


Figure 2. 2: Examples of road signs (Slinn et al., 2005)

Figure 2. 3: Road Markings (Slinn et al., 2005).

Traffic signals, road signs, and road markings can be combined or used individually to convey the maximum information available to a road user.

A. STRENGTHS

The advantage of using manual traffic control management is that as traffic density varies randomly, a traffic officer can allocate the best time appropriately at any given instance (S. A. Parr et al., 2020). Another is that traffic officers can use common sense judgments in varying situations and pedestrians can cross the road safely.

B. LIMITATIONS

Having an officer control traffic is much more expensive than an automated system. Traffic officers are prone to get tired thus having so many humans do the job. Lack of trained traffic police officers makes this problem worse in many cities. In bad weather conditions, it is next to impossible to be able to manually manage traffic successfully. Below is an image depicting traffic control in bad weather.

Figure 2. 4: Traffic Officer controlling traffic in bad weather (Malambo and Chomba, 2018)

If the weather gets to an extreme condition, the traffic officer is at risk of getting an injury thus it being a disadvantage.

2.3.2 Automatic Traffic Management Technique

Traffic Lights, Stoplights, or Robots are simply signaling devices positioned at road intersections, pedestrian crossings, and other locations to help control flows of traffic. As a way of trying to eliminate most weaknesses of the manual traffic control system, an automatic traffic management method is implemented (S. Parr & Wolshon, 2015). This system includes a simple three-color traffic signal which is Red: Tell traffic to stop or prohibits any traffic from proceeding, Amber: This warns that the signal is about to change to red, requiring drivers to stop if it is safe to do so, and others allowing some other drivers to go through the intersection if safe to do so or the signal is about to change to green allowing drivers to get ready to proceed and Green: Allows traffic to proceed in the direction that is denoted. Below is an image depicting a traffic light.

Figure 2. 5: Traffic Light

A traffic light is composed of the main display which has red, orange, and green lights. The lights on the main display are made of Led lights which stand a test of time before they require replacement or servicing. Traffic lights also have a signal cabinet or control box where the traffic controller is stored. The signal cabinet houses a power panel in charge of electrical distribution, the controller itself which is the brain of the whole system. And lastly supports such as the body of the traffic light systems.

A. STRENGTHS

Traffic lights provide authority to drivers to move with confidence as they know that the light has permitted them to move. They also help or make traffic move in an orderly manner as well as reduce the number of road accidents such as pedestrian accidents. Traffic light indications are also visible in foggy weather as compared to the conventional manual system.

B. LIMITATIONS

Traffic lights slow down the flow of traffic at peak hours. Also, when there is a signal breakdown there is a widespread of traffic difficulties. Rear-end collisions are prone to happen at traffic lights. Also, unjustified traffic signals can cause excessive delays, disobedience of signals, and diversion of traffic to inadequate alternate routes.

2.3.3 Traffic Control System Using Inductive Loop Detection

An Inductive loop is simply a detection system that uses a moving magnet or an alternating current to induce an electric current in a nearby wire. They are made of wire "coiled" to form a loop that usually is a square, circle, or rectangle shape and it is installed into or under the surface of the roadway for transmission and reception of communication signals, or for detection of metal objects in metal detectors or vehicle presence indicators. Inductive loops work like metal detectors in the sense that they measure the change in the field when objects pass over them e.g. when a car drives over a loop sensor the loop field changes which allows the detection device to detect the presence of an object (car). Below is an image depicting an inductive loop.

Figure 2. 6: Inductive Loop at Traffic Intersection

Bhaskar et al., (2016) proposed that the system is to make use of two inductive loops in each lane to measure traffic density. The traffic density is to be classified into three levels namely normal, medium congested, and extremely congested. Suppose that there is a red signal in LANE-1 at an intersection. The vehicles will come and stop causing the loop-1of LANE-1 to go high. This means that the vehicles halted extend up to the point where the first loop has been installed. And as such the traffic can be said to be medium congested in this lane. In case of detection in both circuits of both the loops of a lane, the signal simultaneously goes high and this indicates that there are halted vehicles extending up to or beyond the second loop. If the loops don't detect any presence of a vehicle at a red signal, it would mean that there is normal traffic in that lane. Thus, it can be concluded that loop-1 is a measure of medium congestion while loop-2 will be measuring extreme congestion. The proposed system was to only allow one lane to be functional at an intersection while the other lanes await a green light. Below is an image depicting the proposed system with a timer.

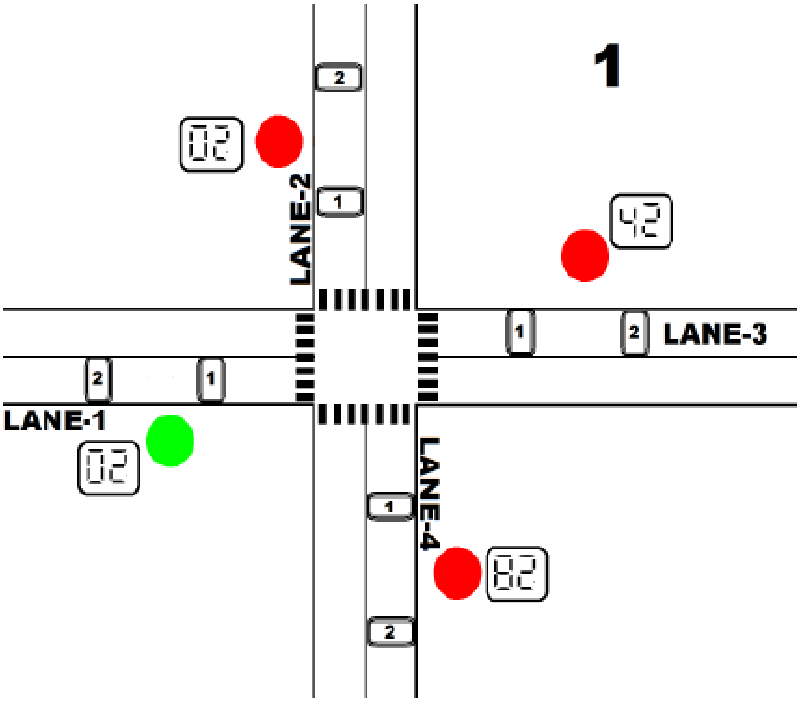


Figure 2. 7: Traffic lights with inductive loops with a timer of 40sec (Bhaskar et al., 2016).

A. STRENGTHS

Inductive loops are accurate and reliable at detecting the density of traffic. They are inexpensive and they also consume little power. They are also convenient and hidden since they are installed under the road.

B. LIMITATIONS

As much as inductive loop detections are highly accurate, they have challenges in detecting small cars. They are sensitive to temperature fluctuations and they also get damaged easily as the road begins to deteriorate because of heavy vehicle/portholes. Inductive loop detection is also affected by the metallic particles in road construction material and the is a high risk of theft of the loop and feeder cable.

2.4 Comparison of Reviewed Systems

Below is a table that shows the comparisons of the reviewed systems.

Table 2. 1 Comparison of Systems

		Features				
		Fixed Time	Bad Weather Performance	Vehicle Detection Accuracy	Real-Time Congestion Resolution	Working Time (24/7)
System	Manual Traffic Control Managements (S. Parr & Wolshon, 2015)	NO	VERY POOR	FAIR	GOOD	NO
	Automatic Traffic Management Technique (Pashupatimath, 2018)	YES	GOOD	N/A	N/A	YES
	Traffic Control System Using Inductive Loop Detection (V.S & Shaithya.V, 2015)	NO	FAIR	FAIR	FAIR	YES
	Proposed System	NO	FAIR	VERY ACCURATE	GOOD	YES

2.5 Proposed System

The proposed system is aimed at making the traffic signal to be more efficient and feasible. Therefore, reducing traffic as well as utilizing the time. The proposed system will be density-based which means it will give priority to the lane which has comparatively a greater number of vehicles. Image processing is the method that will be used for calculating the density. Python will be used as a simulation environment.

2.6 Summary

From the above, it can be concluded that Traffic signal optimization is dependant on several factors such as real-time traffic conditions, volume, congestion, and route choice. To optimize traffic signals of urban roads all these variables are to be considered and a best-fitted method to solve the problem can be formulated. This chapter provides a definitive review of the components of the system and related literature was given. Firstly, the literature outlined the fields of study subject to the project research, a review of systems similar, and thereafter a discussion on the details following the functional and non-functional requirements of the proposed system.

CHAPTER 3 – RESEARCH METHODOLOGY

3.1 Introduction

A software is a collection of data or instructions that tell a computer what to do. For software to be developed to has to go through a Software Development life cycle. A software Development Life Cycle (SDLC) is simply a systematic process for building software that ensures the quality and correctness of the software built. Its main aim is to produce high-quality software that meets customer expectations. For any IT project to be developed, it involves a software development methodology which is important because if the framework is chosen correctly it ensures the success of the project (Zima, 2015). A software development method is comprised of activities like analysis, planning, development, testing, deployment, maintenance, and retirement. The purpose of this chapter is to choose a software development methodology that will be used to effectively implement the density-based traffic control system.

3.2 Selected Methodology

The methodology used for this project is the waterfall model. Each phase in the Waterfall model has to be completed fully before the next phase can commence and thus, phases don't overlap. The Waterfall software development model is used for a small project where no uncertain requirements are present. After each phase is completed, a review is done to determine if the project is on the right path and whether to continue or discard the project. This model software tests the software only after the development is complete.

Below is an image of the waterfall model.

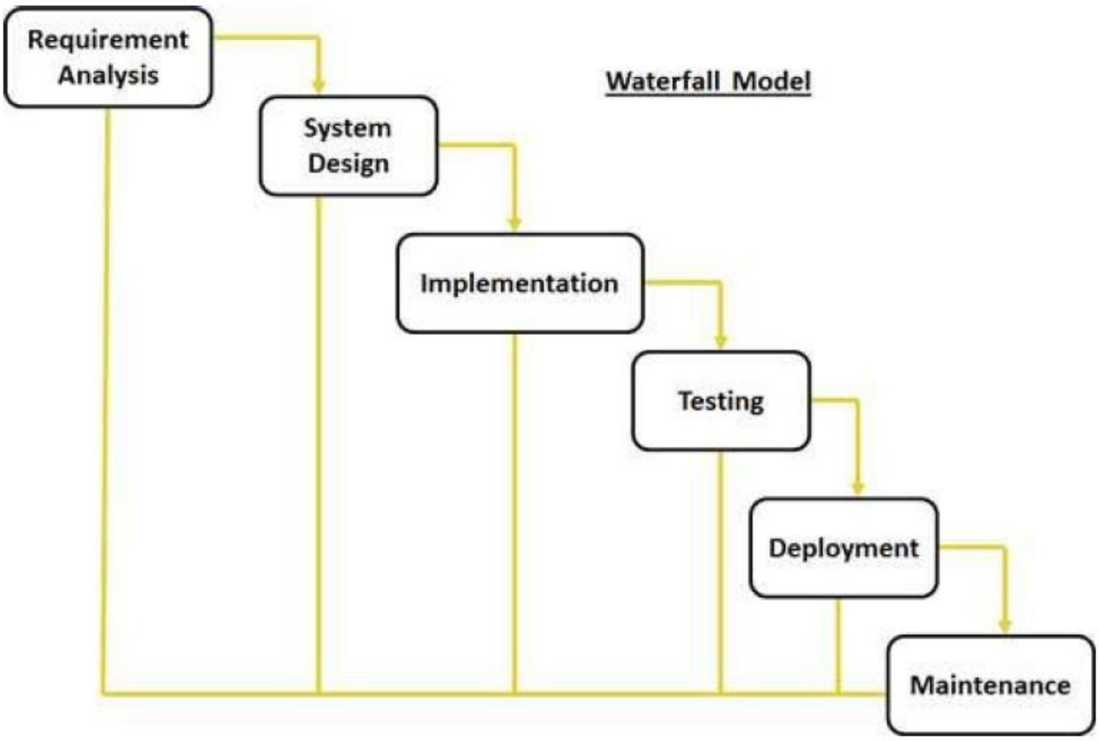


Figure 3. 1: Waterfall Model Phases (Bulman, 2017)

The phases of the waterfall model are:

3.2.1 Requirements Analysis

This is the first and initial stage of the waterfall model where all the requirements for the system to be developed are captured. Requirements are captured or collected from the end-user of the system. At this phase of the project requirements such as traffic data at intersections were gathered. Other requirements such as how the system should behave when there is traffic and when there isn't.

3.2.2 System Design

This is the second phase of the Waterfall model. Its function is to study the requirements that are listed in phase one in preparation for system design. This process allocates the required resources to either hardware or software by establishing an overall architecture. The software design process involves the identification and description of fundamental system relationships and their abstractions. During this phase, the proposed system was designed in terms of what software and frameworks to be used such as OpenCV and Python programming language.

3.3.3 Implementation and Unit Testing.

In this phase, the coding of the system is commenced and the system is developed into programs called units. The individual program units are tested to verify that each of them meets its specification. At this stage, individual parts of the system are designed such as image retrieval, image conversion as well as the traffic timer after the density has been assessed.

3.3.4 Integration and System Testing

The various program units are integrated to form a complete system and it is tested for proper coordination among system modules as well as its behavior. Once this phase is completed, the software is delivered to the customer. At this phase, all the individual units are bundled into one whole system and it's tested for its functionality between the individual units.

3.3.5 Operation and Maintenance

This is the longest phase in the Waterfall model. The system is installed and placed in a running environment. Once put to practical use, problems arise and thus maintenance comes in. Maintenance helps to solve issues that are related to the system that arise only after deployment of the system. The system is then deployed and if problems arise, maintenance is carried out.

3.3 Justification

The waterfall model was used in this system because it allows for departmentalization and control of the software development life cycle of the project. In the waterfall model, schedules can also be set with deadlines to review each phase of the project and decide whether to continue or whether to discard it. It is also appropriate for this project because it is small and every phase or stage of the project is planned and scheduled before the work on the phase commences.

3.4 Technologies and framework used

3.4.1 Python Programming Language

Python is a high-level, interpreted, and general-purpose programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming (van Rossum & Python development team, 2018). According to Nosrati, (2011) some of the characteristics of python are that:

- It is open source
- It is fast and powerful
- It is portable
- It supports other technologies

Nosrati (2011) also stipulated that python can be used to write programs types such as:

- System Programs
- Graphical User Interface (GUI)
- Numerical Programming
- Database Programming
- Components Integrity and many others

3.4.2 OpenCV

Open Source Computer Vision Library commonly known as OpenCV is a library of programming functions mainly aimed at real-time computer vision. It is used for all kinds of image and video processing and analysis. It is capable of processing images and videos to identify objects, faces, or even handwriting (Scherfgen & Sc, 2015). OpenCV is a library of programming functions mainly used for image processing and some of its image processing features according to Naveenkumar, (2016) are

- Image Filtering
- Image Transformation
- Object Tracking
- Feature Detection

3.5 Summary

Various methodologies are used in the software development process. This chapter highlighted the chosen methodology and why it was chosen. The selected methodology is suitable because requirements are not likely to change and they are well defined. An overview of the methodology justification was given as well as the technologies and frameworks used.

CHAPTER 4 – PROJECT MANAGEMENT

4.1 Introduction

Project management is simply the application of processes, methods, skills, knowledge, and experience to a project's activities. Good management does not guarantee project success, however, bad management normally results in project failure in cases such as the software may be delivered after the due date, cost more than originally estimated, or fail to meet the expectations of customers (Sommerville, 1977).

This chapter focuses on project risk analysis and quality management as well as providing a review of the risks that are associated with the proposed system. Other sections of this chapter provide a description of the cost model that was used for the project and the estimated costs calculated as well as outlining the schedule and work plan estimated for the system designed.

4.2 Risk and Quality Management

Currently, software controls more tools and machines which are part of the daily life in modern industrialized cities. Thus, the existence of risks which are directly linked to the non-functioning of software. Because of this, sufficient quality of software systems must be ensured. During software development, the project time and budget don't allow complete coverage of the whole software functionality through software testing. Risk and the quality of the software are to be analyzed to prepare for uncertainties that might occur.

Project success is evaluated via the triple constant as shown in figure 4. 1 below:

Figure 4. 1 Triple Constant Model (Van Wyngaard et al., 2012)

Without the effective management of the triple constraint as an interrelated system, projects run the risk of becoming separated from the purpose (Van Wyngaard et al., 2012).

4.2.1 Risk Management

A risk is defined as a bundle of future uncertain events with a probability of occurrence and potential for loss. Risk management is defined as a planned and structured process aimed at helping the project team make the right decision at the right time to identify, classify, quantify the risks and then to manage and control them (Srinivas, 2018). It should begin at the earliest stages of program planning and continue throughout the total life-cycle of the project to help ensure project success or account for uncertainties.

4.3 Risk Register

Dunović et al., (2013) defines a risk register as the main tool for an integrated analysis of risks relating to time, costs, and technical issues. Risk Analysis project involves a series of steps to quantify the impact of uncertainty on a project. These activities are risk identification, assessment of the likelihood and impact of the project estimate. The estimated risks and planned responses are shown in the table below.

4.3.1 Proposed System Risks

The following are the identified risks that are associated with the proposed system.

- *Ambiguity in Requirements*

  This comes about when the requirements have not been clearly defined and captured. Hence, attention must be paid to requirements gathering, and requirements must be clearly defined and outlined.

- *Loss of Data and Source Code*

  For the fact that a computer system to be used to develop the proposed system and the fact that computer systems are prone to failure, the risk of losing data and the source code is very high. Proper backups of important files should be observed.

- *Failure to Complete the System on Time*

  Gantt chart schedule is to be followed to ensure that project is on the right track.

- *Poor or No Wireless Internet Connection*

Due to the nature of the proposed system, connection to the internet should be readily available during the development process as a lack of internet access will delay the completion of the proposed system.

- *No Electricity (ZESCO)*

For the reason that a computer will be used to develop this project, electricity is needed to power it up. The absence of electricity would delay the completion of the project.

- *Steep Learning Curve Due to Use of Image Processing Technology*

The development of the proposed system will require the learning of how to implement image processing technology, this may take a considerable amount of time as the technologies used are involving.

Table 4. 1 shows the risk register of the proposed system

Ref	Risk Identification	Category	Risk Evaluation (1, 2, or 3)			Risk Result	Risk Owner	Risk Response
			Probability (a)	Impact (b)	Score (a * b)			
1	Ambiguity in requirements	Scope	2	2	4	Medium	Student	Reanalyze requirements and ensure ambiguity is resolved
2	Loss of Data and Source Code	Technical	3	3	9	Very High	Student	Employ the use of cloud storage as well as local frequent backups.

3	Failure to complete the system on time	Scope	1	2	2	Low	Student	Follow Gantt chart with its schedule to ensure that project is on time
4	Poor or no wireless internet connection	Technical	2	2	4	Medium	Student	Purchase enough data bundles and employing the use of different ISPs
5	No electricity (ZESCO)	Technical	3	2	6	High	Student	Having a backup power supp;y such as a generator in place
6	Steep learning curve due to the complexity of image processing technology	Technical	3	3	9	Very High	Student	Do an extensive study on how technology is implemented. And view tutorial videos showing how image processing is used

4.4 Effort Costing Model

The Constructive Cost Model, also referred to as COCOMO, is a well-documented and widely accepted algorithmic model for effort estimation. COCOMO was introduced by Boehm in 1981. This model consists of mathematical equations that identify the developed time, the effort, and the maintenance effort (F. & Aljahdali, 2013). The constructive Cost Model has three variants. These are basic, intermediate, and detailed. The main parameter used for computations is the size of the project. The size is represented in terms of lines of code (LOC) or a thousand lines of code (KLOC).

$$E = a(KLOC)^b$$

Where E represents the software effort computed in person-month, KLOC denotes the size of the code (kilo-lines of code), and A, B represent the COCOMO model parameters. The value of A and B depends on the model of a software project. COCOMO model has three types, depending upon project size. These models are Organic, Semi-detached, and Embedded model (Sachan et al., 2016).

4.5 Effort Calculations for Project

Information Domain Values

Measurement Parameter	Count		Simple ◉	Average ○	Complex ○		Total
Number of user inputs	1	X	3	4	6	=	3.00
Number of user outputs	1	X	4	5	7	=	4.00
Number of user inquiries	0	X	3	4	6	=	.0
Number of files	0	X	7	10	15	=	.0
Number of external interfaces	0	X	5	7	10	=	.0
Count=Total							7.00

Figure 4. 2 Information domain calculation for function points

Complexity Weighting Factors
// heading of the second table Rate each factor on a scale of 0 to 5:
(0 = No influence, 1 = Incidental, 2 = Moderate, 3 = Average, 4 = Significant, 5 = Essential):

Question	0	1	2	3	4	5
1. Does the system require reliable backup and recovery?	◉	○	○	○	○	○
2. Are data communications required?	○	◉	○	○	○	○
3. Are there distributed processing functions?	○	◉	○	○	○	○
4. Is performance critical?	○	○	○	○	○	◉
5. Will the system run in an existing, heavily utilized operational environment?	○	○	○	○	◉	○
6. Does the system require on-line data entry?	◉	○	○	○	○	○
7. Does the on-line data entry require the input transaction to be built over multiple screens or operations?	◉	○	○	○	○	○
8. Are the master file updated on-line?	○	◉	○	○	○	○
9. Are the inputs, outputs, files, or inquiries complex?	○	○	◉	○	○	○
10. Is the internal processing complex?	○	○	○	◉	○	○
11. In the code designed to be reusable?	○	◉	○	○	○	○
12. Are conversion and installation included in the design?	◉	○	○	○	○	○
13. Is the system designed for multiple installations in different organizations?	◉	○	○	○	○	○
14. Is the application designed to facilitate change and ease of use by the user?	○	○	◉	○	○	○
Total 20.00						

Figure 4. 3 Complexity weighting factors calculation

Programming Language	LOC/FP (average)	Select
Assembly Language	320	○
C	128	○
COBOL	105	○
Fortran	105	○
Pascal	90	○
Ada	70	○
Object-Oriented Languages	30	○
Fourth Generation Languages (4GLs)	20	◉
Code Generators	15	○
Spreadsheets	6	○
Graphical Languages (icons)	4	○

LOC/F P: Show LOC/FP 119.00

Figure 4. 4 Selection of programming language to be used

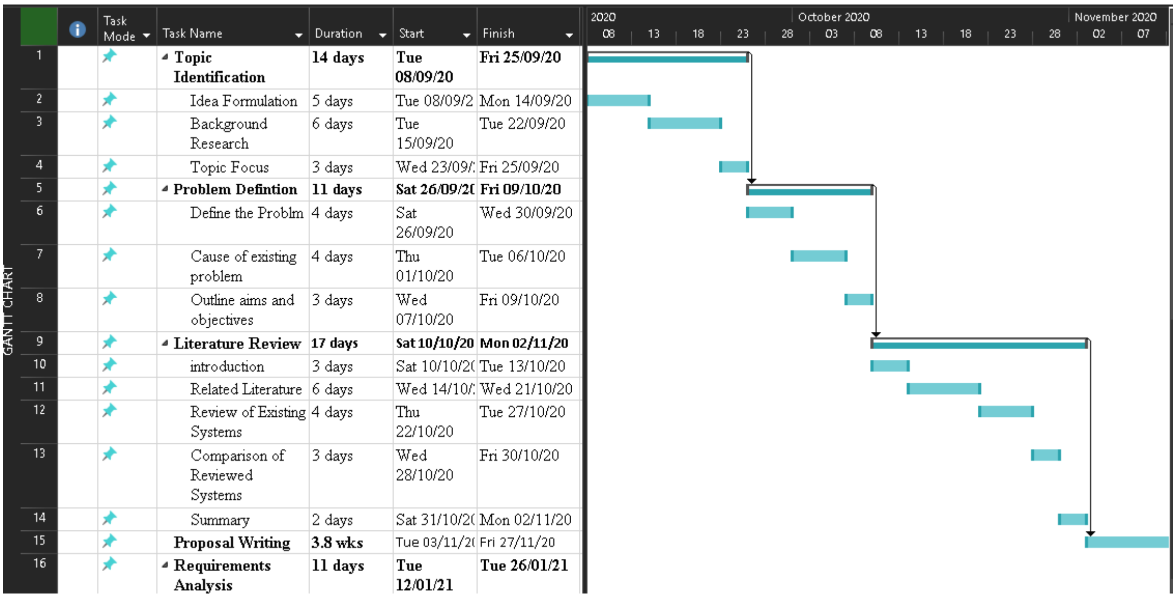
Software Project	a_b	b_b	c_b	d_b	Select
Organic	2.4	1.05	2.5	0.38	○
Semi-detached	3.0	1.12	2.5	0.35	◉
Embedded	3.6	1.20	2.5	0.32	○

Calculate Effort and Duration

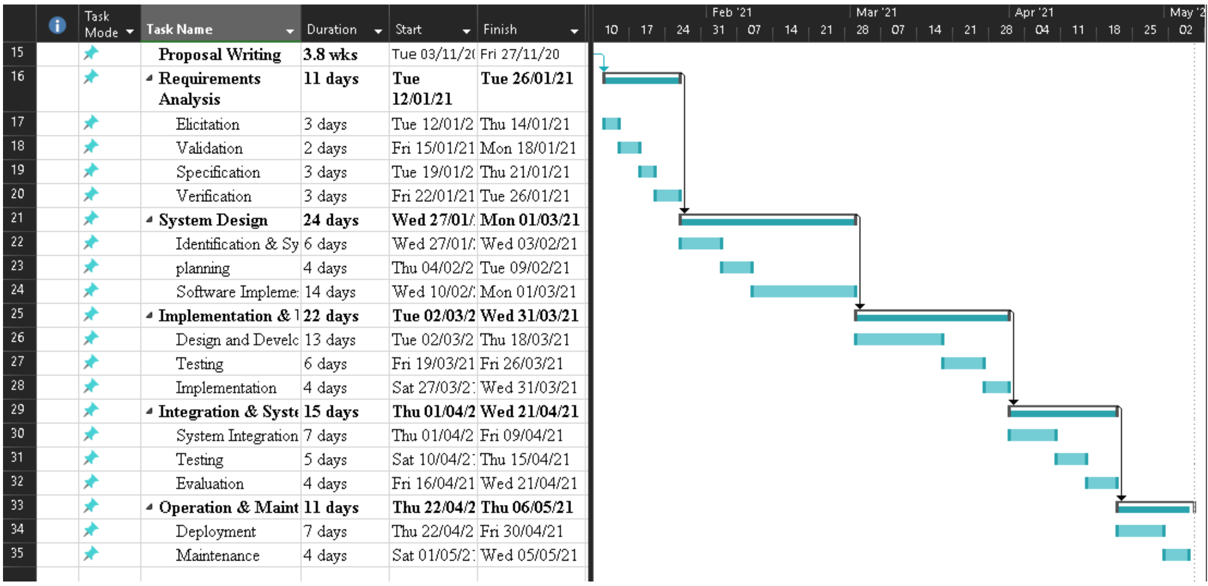
Effort (E) = $a_b(KLOC)^{b}{}_{b}$ = 28 **Duration (D) = $c_b(E)^{d}{}_{b}$ =** 1.59

Figure 4. 5 Effort and Duration estimated (without costing model)

4.6 Scheduling and Work plan



	Task Mode	Task Name	Duration	Start	Finish
1		**Topic Identification**	**14 days**	**Tue 08/09/20**	**Fri 25/09/20**
2		Idea Formulation	5 days	Tue 08/09/2	Mon 14/09/20
3		Background Research	6 days	Tue 15/09/20	Tue 22/09/20
4		Topic Focus	3 days	Wed 23/09/:	Fri 25/09/20
5		**Problem Defintion**	**11 days**	**Sat 26/09/2(**	**Fri 09/10/20**
6		Define the Problm	4 days	Sat 26/09/20	Wed 30/09/20
7		Cause of existing problem	4 days	Thu 01/10/20	Tue 06/10/20
8		Outline aims and objectives	3 days	Wed 07/10/20	Fri 09/10/20
9		**Literature Review**	**17 days**	**Sat 10/10/20**	**Mon 02/11/20**
10		introduction	3 days	Sat 10/10/2(	Tue 13/10/20
11		Related Literature	6 days	Wed 14/10/:	Wed 21/10/20
12		Review of Existing Systems	4 days	Thu 22/10/20	Tue 27/10/20
13		Comparison of Reviewed Systems	3 days	Wed 28/10/20	Fri 30/10/20
14		Summary	2 days	Sat 31/10/2(	Mon 02/11/20
15		**Proposal Writing**	**3.8 wks**	Tue 03/11/2(	Fri 27/11/20
16		**Requirements Analysis**	**11 days**	**Tue 12/01/21**	**Tue 26/01/21**



	Task Mode	Task Name	Duration	Start	Finish
15		**Proposal Writing**	**3.8 wks**	Tue 03/11/2(	Fri 27/11/20
16		**Requirements Analysis**	**11 days**	**Tue 12/01/21**	**Tue 26/01/21**
17		Elicitation	3 days	Tue 12/01/2	Thu 14/01/21
18		Validation	2 days	Fri 15/01/21	Mon 18/01/21
19		Specification	3 days	Tue 19/01/2	Thu 21/01/21
20		Verification	3 days	Fri 22/01/21	Tue 26/01/21
21		**System Design**	**24 days**	**Wed 27/01/**	**Mon 01/03/21**
22		Identification & Sy	6 days	Wed 27/01/:	Wed 03/02/21
23		planning	4 days	Thu 04/02/2	Tue 09/02/21
24		Software Impleme:	14 days	Wed 10/02/:	Mon 01/03/21
25		**Implementation & T**	**22 days**	**Tue 02/03/2**	**Wed 31/03/21**
26		Design and Develc	13 days	Tue 02/03/2	Thu 18/03/21
27		Testing	6 days	Fri 19/03/21	Fri 26/03/21
28		Implementation	4 days	Sat 27/03/2:	Wed 31/03/21
29		**Integration & Syste**	**15 days**	**Thu 01/04/2**	**Wed 21/04/21**
30		System Integration	7 days	Thu 01/04/2	Fri 09/04/21
31		Testing	5 days	Sat 10/04/2:	Thu 15/04/21
32		Evaluation	4 days	Fri 16/04/21	Wed 21/04/21
33		**Operation & Maint**	**11 days**	**Thu 22/04/2**	**Thu 06/05/21**
34		Deployment	7 days	Thu 22/04/2	Fri 30/04/21
35		Maintenance	4 days	Sat 01/05/2:	Wed 05/05/21

Table 4. 6 Schedule and work plan for the proposed system

Table 4. 2 Budget

Name	Amount (K)
Wages	3000/month
PC with all required software pre-installed	7000
Internet	350/month
Power	70/month
Printer	4000

It is assumed that the monthly income of a programmer is k15000 but there are also projects in the other 4 courses. This means that we only acquire 1/5 of the k15000 income and the rest goes to the 4 other courses.

Total Wage for Project:

*K15,000 * 1/5 * 8 months =* ***k24,000***

Total Budget for the Project:

K24,000 + k7000 + k4000 + k2800 + k560 = ***k38,360***

The project duration is 8 months which is split into 2 semesters of 4 months duration each.

4.7 Summary/ Conclusion

This chapter is focused on effort cost as well as risk and quality management whose function is to assist in the assessment of risks that might be encountered and means of avoiding them and effort costs which help to determine the time and effort that would be required to complete the project according to the schedule. COCOMO help to give an estimate of the software costs while the Gantt chart helps outline the documentation and schedule for the development methodologies outlined.

CHAPTER 5 – CONCLUSION

5.1 Introduction

In conclusion, the Density Based Traffic Control System will be developed to help replace the existing system which is the Traffic Lights with a standard allotted time. The proposed system will be designed to help reduce traffic congestions as well as improve traffic flow. The implementation of image processing technology helps in the identification of traffic density and decision making of time to be allotted for a particular lane.

This project is aimed at getting rid of the problems that were identified in the initial analysis of the proposal. Some of the problems identified were traffic congestion, unnecessary wait times at traffic intersections when there is no traffic and eliminating the standard allotted time for traffic lights. These problems are faced by most if not all road users.

Therefore, the proposed system will be designed to eliminate these problems. Some solutions that will be applied to eliminate these problems include the use of image processing technology as well as eliminating the standard allotted time. The proposed solutions help improve traffic flow as well as get rid of traffic congestion which is the main aim of the project.

5.2 Research contributions

In the reviewed, it has been outlined that most traffic control systems aim to eliminate traffic congestion but still face challenges because with time traffic congestion reemerges. We introduce a density-based traffic control system that will employ the use of image processing to detect the density of traffic and help make a decision of the time to be allotted to a particular lane.

REFERENCES

Bhaskar, L., Sahai, A., Sinha, D., Varshney, G., & Jain, T. (2016). Intelligent traffic light controller using inductive loops for vehicle detection. *Proceedings on 2015 1st International Conference on Next Generation Computing Technologies, NGCT 2015*, *September*, 518–522. https://doi.org/10.1109/NGCT.2015.7375173

Dunović, I. B., Radujković, M., & Vukomanović, M. (2013). Risk register developement and implementation for construction projects. *Gradjevinar*, *65*(1), 23–35. https://doi.org/10.14256/jce.722.2012

F., A., & Aljahdali, S. (2013). Software Effort Estimation Inspired by COCOMO and FP Models: A Fuzzy Logic Approach. *International Journal of Advanced Computer Science and Applications*, *4*(11). https://doi.org/10.14569/ijacsa.2013.041127

Jadhav, A. D., Madhuri, B., & Ketan, T. (2015). *Intelligent Traffic Light Control System Based Image Intensity*. *September*, 84–87.

Kerala Police, 2019. *Traffic Officer Manually Controlling Traffic*. [image] Available at: <https://old.keralapolice.gov.in/wings/specialized-wings/traffic-police> [Accessed 1 November 2020].

Malambo, C. & Chomba, G., 2018. *Traffic Officer Controlling Traffic In Bad Weather*. [image] Available at: <https://zambiareports.com/2018/05/17/police-officer-rains-traffic-control/cop-rains/> [Accessed 1 November 2020].

Michels, B. (2014). Traffic Light History Abstracts. *Journal of Chemical Information and Modeling*, *53*(9), 1689–1699.

Naveenkumar, M. (2016). *OpenCV for Computer Vision Applications*. *March 2015*, 52–56.

Nosrati, M. (2011). Python: An appropriate language for real world programming. *World Applied Programming*, *12*, 110–117.

Panjwani, M., Tyagi, N., Venkata Lakshmi Narayana, K., & Shalini, D. (2014). Smart traffic control using image processing. *International Journal of Applied Engineering Research*, *9*(16), 3429–3442.

Parr, S. A., Herrera, N., Wolshon, B., & Smith, T. (2020). Effect of manual traffic control on evacuation time estimates. *Transportation Research Record*, *2674*(9), 809–819.

https://doi.org/10.1177/0361198120932165

Parr, S., & Wolshon, B. (2015). *Manual traffic control for planned special events and emergencies 7*. *7*(2).

Pashupatimath, S. G. (2018). Different Techniques Used in Traffic Control System : An Introduction. *International Journal of Engineering Research & Technology (IJERT)*, *6*(03), 1–4.

Sachan, R. K., Nigam, A., Singh, A., Singh, S., Choudhary, M., Tiwari, A., & Kushwaha, D. S. (2016). Optimizing Basic COCOMO Model Using Simplified Genetic Algorithm. *Procedia Computer Science*, *89*, 492–498. https://doi.org/10.1016/j.procs.2016.06.107

Scherfgen, D., & Sc, M. (2015). *Introduction to OpenCV (Part 1)*. *Part 1*.

Shinde, Y., & Powar, H. (2017). *Intelligent Traffic Light Controller Using IR Sensors for Vehicle Detection*. *Ncetete*, 2393–2395. https://doi.org/10.17148/IARJSET/NCETETE.2017.28

Sommerville, I. (1977). Software engineering. In *Science* (Vol. 195, Issue 4283). https://doi.org/10.1126/science.195.4283.1199

Srinivas, K. (2018). Process of Risk Management K. Srinivas. *IntechOpen*, 1–16. https://www.intechopen.com/books/advanced-biometric-technologies/liveness-detection-in-biometrics

Udoakah, Y. N., & Okure, I. G. (2018). Design and implementation of a density-based traffic light control with surveillance system. *Nigerian Journal of Technology*, *36*(4), 1239. https://doi.org/10.4314/njt.v36i4.34

V.S, R., & Shaithya.V. (2015). A Traffic Control System Using Inductive Loop Detector. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, *Vol. 4*(Issue 5), 4590–4593.

van Rossum, G., & Python development team, the. (2018). *Python Tutorial Release 3.7.0 Guido van Rossum and the Python development team*.

Van Wyngaard, C. J., Pretorius, J. H. C., & Pretorius, L. (2012). Theory of the triple constraint -A conceptual review. *IEEE International Conference on Industrial Engineering and Engineering Management*, *December*, 1991–1997.

https://doi.org/10.1109/IEEM.2012.6838095

Varambally, Samhita & Shahi, Prachi & Rajesh, A. (2018). *Intelligent Traffic Density Monitoring Using RFID*. *March*.

Yousef, A., Shatnawi, A., & Latayfeh, M. (2019). Intelligent traffic light scheduling technique using calendar-based history information. *Future Generation Computer Systems*, *91*(August), 124–135. https://doi.org/10.1016/j.future.2018.08.037

Yousef, K. M., Al-Karaki, J. N., & Shatnawi, A. M. (2010). Intelligent traffic light flow control system using wireless sensors networks. *Journal of Information Science and Engineering*, *26*(3), 753–768. https://doi.org/10.6688/JISE.2010.26.3.2

Zima, D. (2015). Modern Methods of Software Development. *Task Quarterly*, *19*(4), 481–493.