INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE

ISSN: 2057-5688

DENSITY BASED SMART TRAFFIC GREEN LIGHT TIMER ALLOCATION SYSTEM USING CANNY EDGE DETECTION ALGORITHM

[1] Mr. K. KOTESWARA CHARI, [2] A. PAVAN KUMAR, [3] P. GURVA REDDY, [4] M. AKSHAY REDDY

[1]Associate Professor, Dept. Of CSE, Teegala Krishna Reddy Engineering College, Meerpet, Hyderabad,

kkchari530@gmail.com

[2,3,4,]BTech Student, Dept. Of CSE, Teegala Krishna Reddy Engineering College, Meerpet, Hyderabad

pavanpontu@gmail.com, padigapatigurvareddy2000@gmail.com, akshayreddy1225@gmail.com

Abstract: *Urban traffic congestion is the focus of the project density-based smart traffic green light timer allocation system using clever edge detection algorithm. With more and more people living in metropolitan areas, there is a pressing need for cutting-edge technology and tools to better manage traffic flow. As the volume of traffic continues to grow, it has become clear that the timers and human controls now in use are inadequate. Using clever edge detection with digital image processing, we suggest a strategy for allocating time in this project in accordance with the density of vehicles. This revolutionary traffic management system is light-years ahead of the competition when it comes to speed of reaction, control of vehicles, automation, dependability, and overall efficiency. In order to use this method, we are sending the current traffic image to an application, which will then extract edges from the image, with more white edges indicating more traffic and fewer white edges indicating less traffic.*

I. INTRODUCTION

Congestion on the roads is a serious problem in any contemporary metropolis. A recent World Bank research found that in the previous decade, the average speed of vehicles in Dhaka has dropped from 21 km to 7 km/h. There is evidence from research comparing metropolitan areas that congestion decreases regional competitiveness and shifts economic activity by reducing the rise of county gross product or metropolitan area

ISSN: 2057-5688

employment. An entirely new traffic management system that makes full use of existing infrastructures is urgently needed as more and more cars enter service in a system that is already overburdened with congestion.

Because it takes so much time, money, and effort to construct new highways, flyovers, elevated motorways, etc., prioritising the effective and careful use of already existing infrastructures makes more sense. Discover information on the flow of traffic. There are some who compute the total number of automobiles and others who count the total number of pixels.

The outcomes of using these techniques to gather traffic statistics have been encouraging. However, if the distance between vehicles is extremely close, the count could be off. Additionally, rickshaws and auto-rickshaws, which are common forms of transportation in South Asian countries, might not be included in the total.

And counting pixels has the drawback of counting immaterial elements as vehicles, such as pavements and people. Time allocations based on traffic density alone have been suggested in several of the works. However, this might be inconvenient for people who drive in less-frequently-used lanes.

Some of the staff members count the number of automobiles, while others count the number of pixels. Some encouraging outcomes have been seen using these techniques to collect traffic statistics. Although rickshaws and auto rickshaws are the most common forms of transportation in South Asian countries, inaccurate results may be obtained by counting the number of vehicles if the inter-vehicular spacing is particularly small (two vehicles that are close to one another may be counted as one).

Furthermore, irrelevant items such as people and sidewalks are considered as automobiles when calculating the number of pixels. Some research has suggested scheduling appointments only around traffic flow. However, this may be an issue for those who often choose less-congested lanes. In this paper, we propose an image-processing-based intelligent traffic management system for estimating vehicular volumes.

It has been shown that the existing, practically antiquated traffic management system has flaws, and an alternative traffic control system has been developed and explored for its benefits. In order to do this,

ISSN: 2057-5688

four types of traffic scenarios were shot as samples. After the edges have been detected, the sample images are compared to the reference picture to evaluate how well they match. This closeness necessitated the implementation of the procedure for assigning time to each individual photograph.

II. LITERATURE SURVEY

Urban congestion is a growing concern, hence innovative approaches and technologies are needed to advance the existing state of the art in traffic management. The present methods, which may include timers or human control, are obviously insufficient to deal with the issue at hand. This article suggests a system that, utilising advanced edge detection and digital image processing, may control traffic flow by constantly keeping tabs on the number of vehicles on the road.

The system might achieve this goal with the help of sophisticated edge detection and digital image processing. When compared to current traffic control technology, this cutting-edge system boasts significant improvements in response time, vehicle management, automation, reliability, and overall efficacy. The full procedure of acquiring images, detecting edges, and allocating green signals is described with relevant diagrams, and the results are validated by means of hardware implementation. In addition, we present four photos as examples, each of which shows a different traffic scenario. These images serve as examples of process. We used the four pictures given to us to help us grasp this concept.

Most cities throughout the world have traffic congestion, and this is despite the fact that many subpar traffic control systems are already in place. This is because of the growth in population, which has led to a corresponding rise in the number of automobiles on the road. Since the present traffic light system operates on the principle of a fixed period, the duration of each side's green light does not vary in response to variations in traffic volume. This method has been in practise for some time now. In an effort to resolve this problem, we launched a project to dynamically control traffic flow.

In order to achieve this goal, we are employing cameras, image processing software, and Arduino to manage the traffic lights. After importing photos of the

ISSN: 2057-5688

intersection roadways into image processing software, a computation is made to determine the vehicle density. Once the data has been processed, the system will decide on its own how long each traffic light should be based on the volume of traffic along each route. As a result of population concentrations, image processing has made feasible an advanced traffic light control system that can also distinguish between different types of emergency vehicles.

For a country to prosper, its transport and public transport systems must be well-developed. Congestion and poor management waste time and money, hence an effective, timely, and inexpensive traffic management system should be put in place. Traffic management has become an urgent issue in the modern world.

Since the number of users is growing every day, a sophisticated traffic control system is necessary for efficient operation. Several methods are at one's disposal for efficient traffic control. Given the dynamic nature of real-world events, however, there is no ideal strategy and no system that can automatically adjust to new conditions as they arise. There are two types of common traffic management systems, and they are:

Controlling traffic manually requires the use of personnel. He'll have a signboard, a sign light, and a whistle, and he'll be part of a team of traffic cops stationed there.

Electronic sensors and timers power the automatic controls. When a sensor detects that a vehicle is available, it adjusts the schedule accordingly.

However, it is neither an efficient or adaptable system due to its many shortcomings. In this project, we aimed to accomplish the following by presenting a way for administering a smart traffic light management system based on population density:

Identify whether cars are present or absent in the road picture.

If the road is empty, the light should change to red.

It is important to make sure that the traffic signal turns green for an appropriate amount of time dependent on the quantity of cars in the area.

The proposed system may be implemented using Mat Lab software, and its major goal and focus is on effective traffic control.

ISSN: 2057-5688

The camera's viewing point, just above the intersection's traffic light, allowed for a comprehensive overview of the whole thoroughfare. Each frame of the video is compared to the original, which provides information on the vehicle density, and the video is presented as a series of still images. The total amount of cars is also shown on the screen.

Time slots are shown as they should be using today's traffic management methods. Because of this, the green light suddenly became yellow. Then, an emergency vehicle detection system is utilised to help identify an ambulance and other emergency vehicles, directing additional focus to the affected lane. Then, they're sent to the hardware, which includes a USART module for sending control data to the microcontroller and an ATMEGA 8 microprocessor for actually controlling the traffic light. A USART module serves to link the two devices together.

Someone must be in charge of the traffic signal, since this is the only reasonable explanation. Using the traffic density data provided by the Android app, the user may choose a location that best suits his needs. Current events and related data are included.

information on traffic conditions and other data available on several websites. Depending on the current situation, the location you choose to go to might be different. Due to the low complexity of this application, no extra charges apply.

III SYSTEM DESIGN

SYSTEM ARCHITECTURE

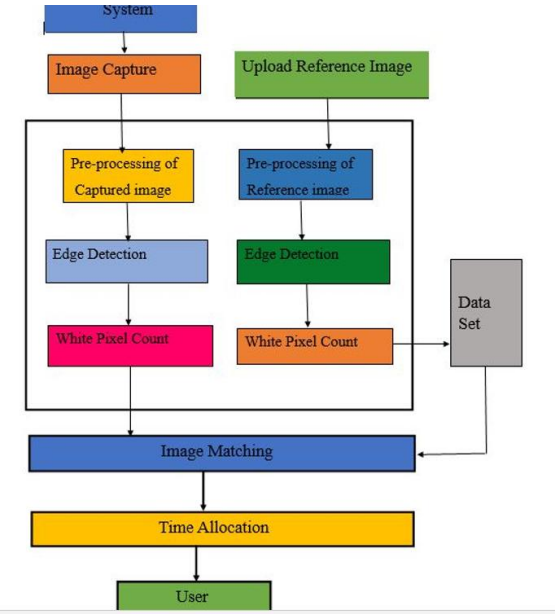


FIGURE 4.1: SYSTEM ARCHITECTURE

DATA FLOW DIAGRAM:

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

The data flow diagram (DFD) is one of the most important modelling tools. It is used

ISSN: 2057-5688

to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

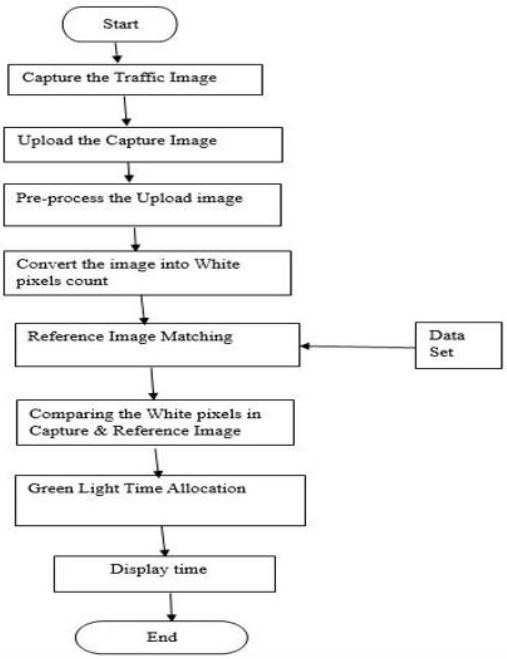


FIGURE 4.2 DATA FLOW DIAGRAM

UML DIAGRAMS

UML stands for Unified modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

ISSN: 2057-5688

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modelling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

IV IMPLEMENTATION

Guido Van Rossum published the first version of Python code (version 0.9.0) at outsources in February 1991. This release included already exception handling, functions, and the core data types of lists, dict, str and others. It was also object oriented and had a module system.

Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked. Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting Unicode. Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it. "Some changes in Python 7.3:

- Print is now a function
- Views and iterators instead of lists
- The rules for ordering comparisons have been simplified. E.g., a

ISSN: 2057-5688

heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.

- There is only one integer type left, i.e., int. long is int as well.
- The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
- Text Vs. Data Instead of Unicode Vs. 8-bit

Purpose :-

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

Python

Guido Van Rossum published the first version of Python code (version 0.9.0) at outsources in February 1991. This release included already exception handling, functions, and the core data types of lists, dict, str and others. It was also object oriented and had a module system.

Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked. Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting Unicode. Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it. "Some changes in Python 7.3:

- Print is now a function
- Views and iterators instead of lists
- The rules for ordering comparisons have been simplified. E.g., a heterogeneous list cannot be sorted, because all the elements of a

ISSN: 2057-5688

list must be comparable to each other.

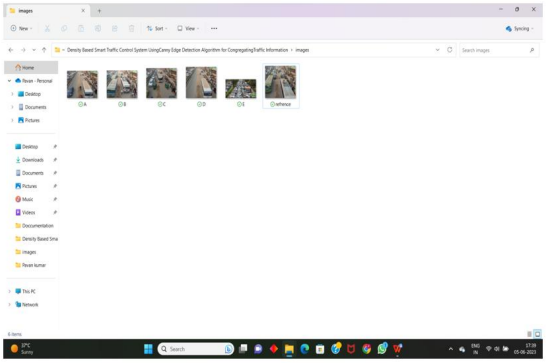
- There is only one integer type left, i.e., int. long is int as well.
- The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
- Text Vs. Data Instead of Unicode Vs. 8-bit

Purpose :-

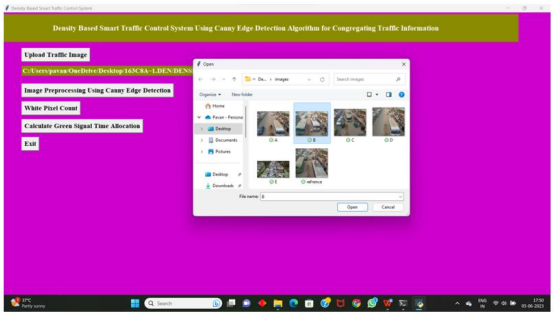
We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

V RESULTS

To implement this project, we are using input images given in paper and one reference image. Below are the images screen shots saved inside images folder.

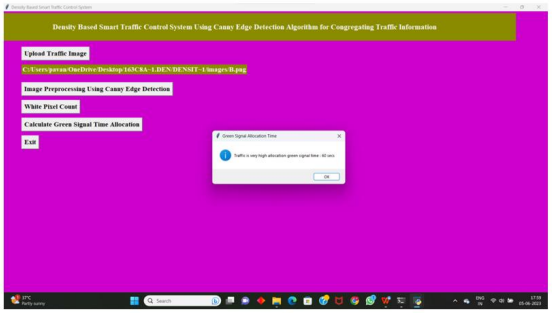


We can upload above images to application to calculate traffic signal time.

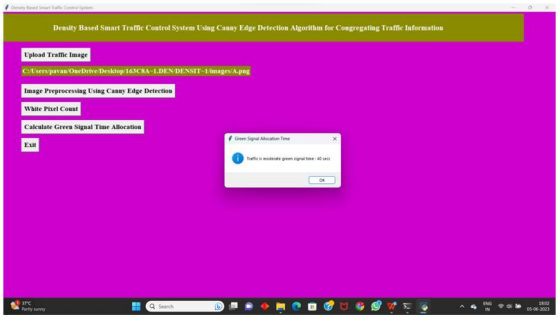


In above screen I am uploading image B and now click on 'Open' button to load image.

Now click on 'Calculate Green Signal Time Allocation' button to get signal time.



For the uploaded image we got message as it contains high traffic and signal time must be 60 seconds. Similarly, you can upload any image and get output. Below is the output for image A.



ISSN: 2057-5688

VI CONCLUSION

The system introduces a fresh strategy for building a traffic management system by using the Canny Edge Detection (CED) algorithm, with the ultimate goal of enhancing current traffic patterns. The suggested system takes pictures at crossroads, then runs them via the CED algorithm. After then, the number of white points is used to determine traffic density, and the timing of the signals is adjusted on the fly to account for variations in volume. Therefore, the system regulates traffic without any human input or additional infrastructure costs. This model is an effort to identify traffic density in a given area at any given moment. The suggested approach, if implemented, would aid in achieving high precision. By eschewing conventional edge detection methods that are inefficient in attaining adequate traffic management, the tested dataset's accuracy will greatly improve. It has been suggested to implement a smart traffic management system that uses image processing as a tool for gauging traffic density. The benefits of the proposed traffic management system have been shown, while the drawbacks of the present, almost antiquated method have been explained. Four representative photos of varying traffic conditions were collected for this purpose. After edge detection is complete, the degree to which test pictures resemble the reference image may be determined. Based on these shared characteristics, the time allocation method has been applied to each picture. The Python programming language has also been used to highlight the percentage and time allocation similarities between the four-example graphics.

VII FUTURE ENHANCEMENT

The OpenCV software is already pre-installed on the Raspberry Pi micro-controller, thus there will be no need to do any further setup in future projects. With the aid of a Raspberry Pi, we can send live traffic data to the control centre, ensuring that drivers in the right areas get the green light for as long as possible at each light.

The camera or imager may be used. It accomplishes its goal by simulating street scenes. It does this by transforming the light's varying attenuation into a signal that may be interpreted as an image. Digital and analogue electrical equipment were employed in imagers.

ISSN: 2057-5688

Using this information, we can create a profile of the flow of traffic and adjust the timing of the lights accordingly. Traffic studies and the daily, weekly, monthly, and yearly fluctuations in traffic volumes may also benefit from profiling. In addition, we may tweak the settings such that emergency vehicles like ambulances benefit most from it. The gathered traffic data may be utilised to identify other routes for a certain daily vehicle to help alleviate the congestion issue.

More long-range cameras will soon be available, allowing for more precise traffic data collecting over greater distances.

Using the cameras installed at signals, we can eventually implement an automated system for issuing fines for infractions like reckless driving, failing to wear a helmet, or skipping a red light.

REFERENCES

[1] Y. Sri Lalitha, et al. “Analysing histopathological images by using machine learning techniques”. Apl Nanosci 13, 2507–2513 (2023).

[2] K. Katiyar and P.V. Arun, “Comparative analysis of common edge detection techniques in context of object extraction, ”IEEE Transactions on Geoscience and Remote Sensing, Vol.50 no.11b (2022).

[3] Partha Narayan Chowdhury, Tommy Chandra Ray, Jia Uddin “A Vehicle Detection Technique for Traffic Management using Image Processing” BRAC University Dhaka, Bangladesh - April 2020.

[4] Badura, S.; Lieskovsky, A., "Intelligent Traffic System: Cooperation of MANET and Image Processing," Integrated Intelligent Computing (ICIIC), 2010 First International Conference on , vol., no., pp.119,123, 5-7 Aug. 2020.

[5] S. Yuan, S. E. Venegas-Andraca, C. Zhu, Y. Wang, X. Mao, and Y. Luo, “Fast Laplacian of Gaussian Edge Detection Algorithm for Quantum Images,” in 2019 IEEE International Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (Smart CNS), 2019, pp. 798–802.

ISSN: 2057-5688

[6] A. Frank, Y. S. K. Al Aamri, and A. Zayegh, "IoT based smart traffic density control using image processing," in 2019 4th MEC International Conference on Big Data and Smart City (ICBDSC), 2019, pp. 1–4.

[7] Shubham Sahu, Dip Anjan Paul, S Senthil Murugan "DENSITY BASED TRAFFIC SIGNAL CONTROL USING ARDUINO AND IR SENSORS", SRM Institute of Science & Technology, Chennai – April 2018 (International Journal of Novel Research and Development - (IJNRD) Shahebgouda Halladamani, Radha R C "Development of Closed Loop Traffic Control System using Image Processing" International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS-2017), 978-1-5386-1887 5/17/©2017 IEEE.

[8] Busarin Eamthanakul, Mahasak Ketcham, Narumol Chumuang "The Traffic Congestion Investigating System by Image Processing from CCTV Camera", 978-1-5090-5210-3/17/©2017 IEEE.

[9] Aman Dubey, Aksdeep, Sagar Rane "Implementation of an Intelligent Traffic Control System and Real Time Traffic Statistics Broadcasting" International Conference on Electronics, Communication and Aerospace Technology (ICECA 2017), 978-1-5090 5686-6/17/ ©2017 IEEE.

[10] Elizabeth Basil, S D Sawant "IoT based Traffic Light Control System using Raspberry Pi" International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS-2017), 978-1-5386-1887-5/17/ ©2017 IEEE.