Smart Control of Traffic Lights Using Artificial Intelligence

CH RAMYA SRI[1], B. MURALI[2]
[1]*PG Student, Quba College of Engineering & Technology*
[2]*Assistant professor, Quba College of Engineering & Technology*

***Abstract*—Traffic congestion is becoming one of the critical issues with increasing population and automobiles in cities. Traffic jams not only cause extra delay and stress for the drivers, but also increase fuel consumption and air pollution. Although it seems to pervade everywhere, megacities are the ones most affected by it. And its ever increasing nature makes it necessary to calculate the road traffic density in real-time for better signal control and effective traffic management. The traffic controller is one of the critical factors affecting traffic flow. Therefore, the need for optimizing traffic control to better accommodate this increasing demand arises. Our proposed system aims to utilize live images from the cameras at traffic junctions for traffic density calculation using image processing and AI. It also focuses on the algorithm for switching the traffic lights based on the vehicle density to reduce congestion, thereby providing faster transit to people and reducing**

***Index Terms*— Traffic Congestion, Urbanization, Air Pollution, Fuel Consumption, Megacities, Road Traffic Density, Real-time Traffic Management, Traffic Control Optimization, Image Processing**

I. INTRODUCTION

With the increasing number of vehicles in urban areas, many road networks are facing problems with the capacity drop of roads and the corresponding Level of Service. Many traffic related issues occur because of traffic control systems on intersections that use fixed signal timers. They repeat the same phase sequence and its duration with no changes. Increased demand for road capacity also increases the need for new solutions for traffic control that can be found in the field of Intelligent Transport Systems. Let us take the case study of Mumbai and Bangalore. Traffic flow in Bangalore is the worst in the world while Mumbai is close behind in fourth position, according to a report detailing the traffic situation in 416 cities across 57 countries. In Bangalore, a journey during rush-hour takes 71% longer. In Mumbai, it is 65% longer.

These conventional methods face certain drawbacks. The manual controlling system requires a large amount of manpower. As there is poor strength of traffic police, we cannot have them controlling traffic manually in all areas of a city or town. So a better system to control the traffic is needed. Static traffic controlling uses a traffic light with a timer for every phase, which is fixed and does not adapt according to the real-time traffic on that road. While using electronic sensors i.e., proximity sensors or loop detectors, the accuracy and coverage are often in conflict because the collection of high-quality information is usually based on sophisticated and expensive technologies, and thus limited budget will reduce the number of facilities.

II.LITERATURE SURVEY

2.1 W. L. Ou, M. H. Shih, C. W. Chang, X. H. Yu, C. P. Fan, "Intelligent Video-Based Drowsy Driver Detection System under Various Illuminations and Embedded Software Implementation" An intelligent video-based drowsy driver detection system, which is unaffected by various illuminations, is developed in this study. Even if a driver wears glasses, the proposed system detects the drowsy conditions effectively. By a near-infrared-ray (NIR) camera, the proposed system is divided into two cascaded computational procedures: the driver eyes detection and the drowsy driver detection. The average open/closed eyes detection rates without/with glasses are 94% and 78%, respectively, and the accuracy of the drowsy status detection is up to 91%. By implementing on the FPGA-based embedded platform, the processing speed with the 640×480 format video is up to 16 frames per second (fps) after software optimizations.

2.2 W. B. Horng, C. Y. Chen, Y. Chang, C. H. Fan, "Driver Fatigue Detection based on Eye Tracking and Dynamic Template Matching" A vision-based real-time driver fatigue detection system is proposed for driving safely. The driver's face is located, from color images captured in a car, by using the characteristic of skin colors. Then, edge detection is used to locate the regions of eyes. In addition to being used as the dynamic templates for eye tracking in the next frame, the obtained eyes' images are also used for fatigue detection in order to generate some warning alarms for driving safety. The system is tested on a Pentium III 550 CPU with 128 MB RAM. The experiment results seem quite encouraging and promising. The system can reach 20 frames per second for eye tracking, and the average correct rate for eye location and tracking can achieve 99.1% on four test videos. The correct rate for fatigue detection is l00%, but the average precision rate is 88.9% on the test videos.

III. SYSTEM ANALYSIS

3.1 EXISTING SYSYEM: Manual Controlling: As the name suggests, it requires manpower to control the traffic. The traffic police are allotted for a required area to control traffic. The traffic police carry signboard, a sign light, and whistle to control the traffic. Conventional traffic lights with static timers: These are controlled by fixed timers. A constant numerical value is loaded in the timer. The lights are automatically switching to red and green based on the timer value. Electronic Sensors: Another advanced method is placing some loop detectors or proximity sensors on the road. This sensor gives data about the traffic on the road. According to the sensor data, the traffic signals are controlled.

3.1.1.DISADVANTAGES: These conventional methods face certain drawbacks. The manual controlling system requires a large amount of manpower. As there is poor strength of traffic police, we cannot have them controlling traffic manually in all areas of a city or town. So a better system to control the traffic is needed. Static traffic controlling uses a traffic light with a timer for every phase, which is fixed and does not adapt according to the real-time traffic on that road

3.2.PROPOSED SYSTEM: Now-a-days due to increasing number of vehicles it's becoming difficult to manage traffic efficiently which leads to longer duration journey and maximum petrol consumption and to avoid this problem standard techniques was introduce such as manual traffic control which require more number of traffic person, static time traffic control which is not effective as it will use same timer for all lanes with heavy and light traffic and sensor based traffic management but this require heavy budget of sensor deployment to sense and manage traffic based on density.

3.2.1.ADVANTAGES:

➢ It will give better accuracy

➢ Better accuracy

➢ Better prediction

3.3.SYSTEM REQUIREMENTS

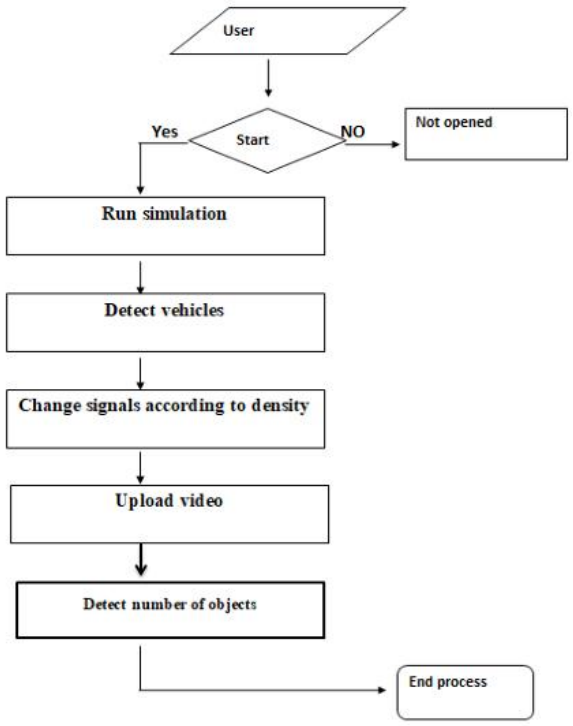
3.3.1.HARDWARE REQUIREMENTS(minimum):

- System : Pentium IV 2.4 GHz
- Hard Disk : 40 GB
- Ram : 512 Mb. 3.3.2.

SOFTWARE REQUIREMENTS:

- Operating System: Windows
- Coding Language: Python 3.7

IV. SYSTEM ARCHITECTURE



V. SYSTEM DESIGN

UML: Introduction to UML: The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling 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. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

Goals of UML The primary goals in the design of the UML were:

• Provide users with a ready-to-use, expressive visual modeling language so they and exchange meaningful models.

• Provide extensibility and specialization mechanisms to extend the core concepts.

• Be independent of particular programming languages and development processes.

• Provide a formal basis for understanding the modeling language.

• Encourage the growth of the OO tools market. can develop

• Support higher-level development concepts such as collaborations, frameworks, patterns and components.

• Integrate best practices.

VI. SOFTWARE ENVIRONMENT

Python: Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Getting Python The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python https://www.python.org. Windows Installation Here are the steps to install Python on Windows machine. • Open a Web browser and go to https://www.python.org/downloads.

- Follow the link for the Windows installer python-XYZ. msifile where XYZ is the version you need to install.
- To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

Interactive Mode

Programming Invoking the interpreter without passing a script file as a parameter brings up the following

prompt−$python
Python2.4.3(#1,Nov112010,13:34:43)

VII. SYSTEM IMPLEMENTATION

System Implementation of Smart Traffic Light Control Using Artificial Intelligence:

1. Data Collection and Preprocessing: - Collect real-time traffic data from sensors, cameras, and other sources at intersections. - Preprocess the data to remove noise, handle missing values, and normalize features.

2. Model Selection: - Choose appropriate AI techniques such as reinforcement learning, deep learning, or a combination based on the specific requirements and complexity of the traffic environment.

3. Model Training: - Train the selected AI model using historical traffic data to learn patterns and optimize traffic light timings. - For reinforcement learning-based approaches, define states, actions, rewards, and the learning algorithm (e.g., Q-learning, Deep Q-Networks).

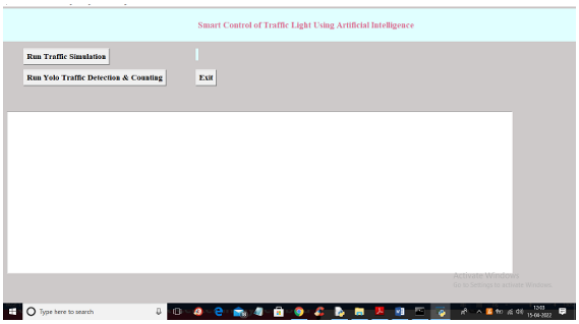
4. Real-Time Traffic Control: - Deploy the trained model to make real-time decisions on traffic light timings. - Continuously monitor traffic conditions and update the model's parameters as needed.

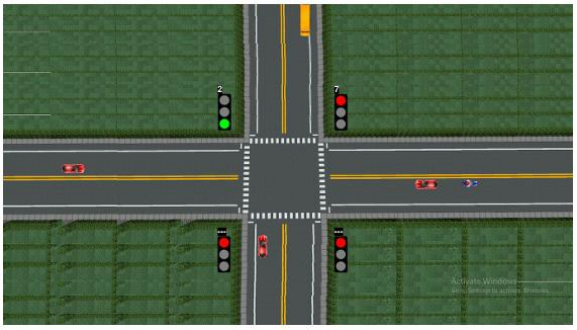
5. Integration with Traffic Infrastructure: - Interface the AI-based traffic light control system with the existing traffic infrastructure, including traffic lights, controllers, and communication networks.

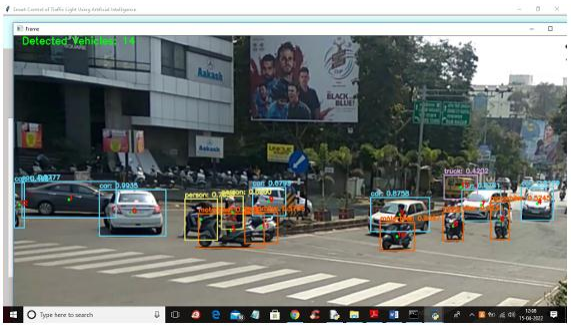
VIII SYSTEM TESTING

Testing: During the software development process, unit testing and coding may be accomplished at the same time. It is also possible to segregate unit testing from other testing. It is imperative that strategies and tactics be put to the test. Detailed functional tests will be written up for future reference during field testing. The test's

IX SCREENSHOTS







X.CONCLUSION

In conclusion, the proposed system sets the green signal time adaptively according to the traffic density at the signal and ensures that the direction with more traffic is allotted a green signal for a longer duration of time as compared to the direction with lesser traffic. This will lower the unwanted delays and reduce congestion and waiting time, which in turn will reduce fuel consumption and pollution. According to simulation results, the system shows about 23% improvement over the current system in terms of the number of vehicles crossing the intersection, which is a significant improvement. With further calibration using reallife CCTV data for training the model, this system can be improved to perform even better. Moreover, the proposed system possesses certain advantages over the existing intelligent traffic control systems prevalent such as Pressure Mats and Infrared Sensors. The cost required to deploy the system is negligible as footage from CCTV cameras from traffic signals is used, which requires no additional hardware in most cases, as intersections with heavy traffic are already equipped with such cameras. Only minor alignment may need to be performed. The maintenance cost also goes down as compared to other traffic monitoring systems such as pressure mats that normally suffer wear and tear due to their placement on roads where they are subjected to immense pressure constantly. Thus, the proposed system can thus be integrated with the CCTV cameras in major cities in order to facilitate better management of traffic

REFERENCES

[1]. TomTom.com, 'Tom Tom World Traffic Index', 2019. [Online]. Available: https://www.tomtom.com/en_gb/traffic-index/ranking/

[2]. Khushi, "Smart Control of Traffic Light System using Image Processing," 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), Mysore, 2017, pp. 99-103, doi: 10.1109/CTCEEC.2017.8454966.

[3]. A. Vogel, I. Oremović, R. Šimić and E. Ivanjko, "Improving Traffic Light Control by Means of Fuzzy Logic," 2018 International Symposium ELMAR, Zadar, 2018, pp. 51-56, doi: 10.23919/ELMAR.2018.8534692.

[4]. A. A. Zaid, Y. Suhweil and M. A. Yaman, "Smart controlling for traffic light time," 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), Aqaba, 2017, pp. 1-5, doi: 10.1109/AEECT.2017.8257768.

[5]. Renjith Soman "Traffic Light Control and Violation Detection Using Image Processing”.” IOSR Journal of Engineering (IOSRJEN), vol. 08, no. 4, 2018, pp. 23-27

[6]. A. Kanungo, A. Sharma and C. Singla, "Smart traffic lights switching and traffic density calculation using video processing," 2014 Recent Advances in Engineering and Computational Sciences (RAECS), Chandigarh, 2014, pp. 1-6, doi: 10.1109/RAECS.2014.6799542.

[7]. Siddharth Srivastava, Subhadeep Chakraborty, Raj Kamal, Rahil, Minocha, “Adaptive traffic light timer controller” , IIT KANPUR, NERD MAGAZINE

[8]. Ms. Saili Shinde, Prof. Sheetal Jagtap, Vishwakarma Institute Of Technology, Intelligent traffic management system:a Review, IJIRST 2016

[9]. Open Data Science, ‘Overview of the YOLO Object Detection Algorithm’, 2018. [Online]. Available: https://medium.com/@ODSC/ overview-of-the-yolo-object-detection-algorithm 7b52a745d3e0

[10]. J. Hui, ‘Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3’, 2018. [Online]. Available: https://medium.com/ @jonathan _hui/ real-time-object-detection-with-yolo yolov2-28b1b93e2088