International Journal of Engineering Sciences & Emerging Technologies, Oct. 2023.
ISSN: 22316604 Volume 11, Issue 2, pp: 171-176 ©IJESET

DESIGNING TRAFFIC MANAGEMENT SIMULATOR USING ARTIFICIAL INTELLIGENCE ALGORITHM

Richa Gupta, Sherin Zafar, Imran Hussain

Department of CSE, SEST, Jamia Hamdard, New Delhi, India
richagupta@jamiahamdard.ac.in
sherin.zafar@jamiahamdard.ac.in
ihussain@jamiahamdard.ac.in

ABSTRACT

Traffic congestion is a major problem in urban areas and has become a significant challenge for the transportation industry. Fuel consumption, air pollution and longer travel times are a result of the growth in the number of vehicles on the road. The traditional traffic management systems have become outdated and are not efficient in managing the traffic flow. A smart traffic control system that can instantly adjust to shifting traffic circumstances is required. The work presented here attempts to create a real-time, AI-based traffic control system that can recognise cars. Techniques related to image processing are used to detect and count the number of vehicles in a given area and then uses AI algorithms to manage the traffic flow. Additionally, the system has a user interface with a display of the current traffic situation and controls for the traffic lights.

KEYWORDS: *Artificial Intelligence, Traffic signals, simulation, image processing, traffic management.*

1. INTRODUCTION

In urban places, traffic congestion is a significant issue and has become a significant challenge for the transportation industry. The traditional traffic management systems have become outdated and are not efficient in managing the traffic flow. An intelligent traffic management system is necessary that can adapt to changing traffic conditions in real-time. The current traffic management systems are based on fixed-time traffic signals and **does** not take into account real-time traffic conditions. The fixed-time traffic signals are not efficient in managing traffic flow and can cause traffic congestion during peak hours. Traffic congestion increases travel times, fuel consumption and air pollution.

Traditional traffic management systems does not prioritize emergency vehicles such as ambulances and fire engines resulting in fatalities. Existing vehicle detection systems are also inefficient at detecting vehicles of different sizes and shapes. The current vehicle detection systems use sensors or cameras to detect vehicles, which can be expensive and require regular maintenance. The current vehicle detection systems are also not accurate and can miss vehicles, leading to inaccurate traffic flow management. Therefore, there is a need for an intelligent traffic management system that can detect vehicles accurately and manage traffic flow efficiently in real-time. The system should be able to adapt as per the change in conditions of traffic. The system should also be efficient in managing traffic flow and reduce traffic congestion. The system should be cost-effective and require minimal maintenance. The development of artificial intelligence (AI) and image processing techniques has provided an opportunity to create an intelligent traffic management system. The system can detect and count the number of vehicles in a given area and then use AI algorithms to manage the traffic flow. The use of AI algorithms can improve the efficiency of traffic management systems by adapting to changing traffic conditions in real-time.

In the upcoming sections, Section 2 describes methodology of the proposed work, followed by results and simulations in Section 3 and conclusion and future work in Section 4. References are listed at the end of the paper.

DOI: 10.5281/zenodo.10441509

2. METHODOLOGY

The section describes the series of algorithms used to develop a technique to manage the traffic in real time.

• **Image Acquisition:** The simulator captures the frames of the traffic intersection using a camera or pre-recorded video. The camera is positioned strategically to cover the desired area of the intersection.

• **Image Pre-processing**: The acquired images undergo pre-processing to enhance the quality and extract relevant features. This typically involves the following steps:

1. **Resizing**: If necessary, the images are resized to a standardized resolution to ensure consistency in the processing pipeline.
2. **Cropping:** If necessary, the images are cropped to focus only on the region of interest, which typically includes the lanes and areas where vehicles are expected to appear.
3. **Color Conversion:** The color space of the image may be converted to enhance the visibility of vehicles against the background.

• **Object Detection:** The pre-processed images are fed into an object detection algorithm, such as YOLO (You Only Look Once). This algorithm is trained on large datasets to recognize and localize objects in images. The detection algorithm analyzes the images and identifies bounding boxes around the vehicles present in the scene.

• **Vehicle Classification:** Once the vehicles are detected, they are classified into different classes. This classification step helps in distinguishing between different types of vehicles, such as cars, buses, trucks, rickshaws, or bikes. Classification can be achieved using machine learning techniques like convolutional neural networks (CNNs) trained on labelled datasets.

• **Vehicle Tracking:** To track the detected vehicles across frames, a tracking algorithm is employed. Common tracking algorithms include the Kalman filter, particle filter, or correlation-based tracking methods. These algorithms assign unique identifiers to each vehicle and track their movement by predicting the position of the vehicle in subsequent frames based on its previous trajectory and the detected bounding boxes.
Integration with the Simulator: The simulator is used to display the behaviour of the vehicles, such as their movement, and adherence to traffic rules. The simulator incorporates the tracked positions and velocities of the vehicles to simulate their realistic behaviour.

• **Simulation Feedback**: The rendered vehicles move and interact with the simulated environment, including other vehicles, and traffic signals. This feedback allows users to observe the interaction of vehicles and the impact on traffic flow.
By following this detailed methodology, the simulator combines image acquisition, pre-processing, object detection, vehicle classification, tracking, integration, and evaluation to simulate realistic vehicle behaviour at the intersection. Figure 1 explains the flow of data among various processes.

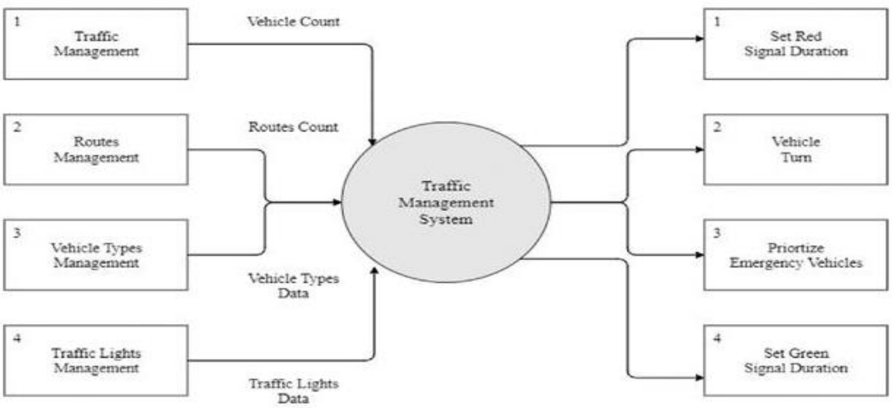

Figure 1: Flow of data in real time traffic management.

DOI: 10.5281/zenodo.10441509

3. RESULTS AND SIMULATION

Code has been developed using Python and its components as it is that is widely used for developing various applications including AI and machine learning-based

A. Detecting Traffic

1. **Image Acquisition**: The code reads input images from a specified directory using the cv2.imread() function. Each image file in the directory is loaded and processed sequentially as shown in Figure 2.

Figure 2: capturing image in real time.

2. **Image Preprocessing**: The code performs color conversion as a form of image preprocessing. Specifically, it converts the input image from the BGR color space to grayscale using the cv2.cvtColor() function. This transformation reduced additional processing steps by reducing the image to a single channel representing the intensity values.

3. **Object Detection:** The code utilizes a vehicle classifier, loaded from the "cars.xml" file using the cv2.CascadeClassifier() function. It employs the classifier to detect vehicles in the preprocessed grayscale image. The vehicle_classifier.detectMultiScale() function is used, which employs a sliding window approach and applies the classifier to various image regions, identifying potential vehicle locations.

4. **Vehicle Classification:** Once the vehicles are detected, the code marks their positions by drawing rectangles around them using the cv2.rectangle() function. It iterates over the detected vehicle regions, retrieves the coordinates, and draws rectangles on the original color image. Additionally, the code displays the count of detected vehicles on each image using the cv2.putText() function.

Figure 3 sows the object detection and classification for step 3 and 4.

DOI: 10.5281/zenodo.10441509

International Journal of Engineering Sciences & Emerging Technologies, Oct. 2023.
ISSN: 22316604 Volume 11, Issue 2, pp: 171-176 ©IJESET

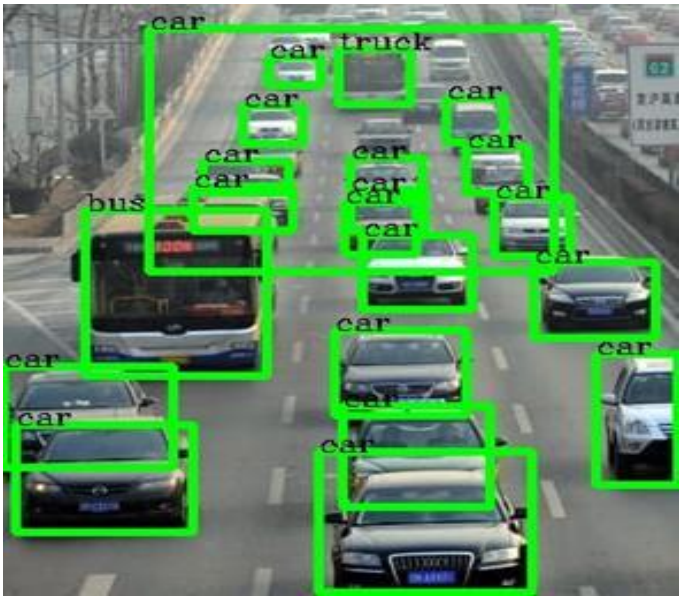

Figure 3: shows each vehicle detected as object and put in box. Also, code marks are placed on each box to identify them as detected.

B. Designing Simulation to manage traffic

Once traffic has been identified, it is important to manage it so that the chaos on roads can be reduced and smooth movement of traffic takes place. Here, how simulation has been made using AI based algorithm is explained.

1. Switching Signals: The algorithm in real time manages the timer of the green signal depending on the traffic density as given by the vehicle detection module. It utilizes the information about the detected vehicles, which is received in JSON format containing object labels, confidence scores, and coordinates. The algorithm classifies the provided data to find the count of the of vehicles for each class. Using this information, it calculates and assigns the appropriate green signal time for the current signal. Additionally, the red signal timers of other signals are updated accordingly. The algorithm operates in a cyclic way, switching between signals based on the assigned timers, thus resulting in to harbour any number of signals at an intersection.

2. Working: The default clock shall be set to the first cycle signal of the algorithm upon its initial execution. The algorithm is then apt to find and correct the timing for all additional signals recorded in an initial run, as well as those observed during subsequent cycles. In order to ensure the proper operation, a separate vehicle detection thread shall be triggered in each direction and control of the timer for this signal will be carried out by the primary thread. The detection threads will capture the next direction photo when the green light timer on the current signal approaches zero. For each green signal, the resulting values shall be analysed in order to determine a timer. This process is running in the background, which will make it easy for your timer to be assigned and prevent delays. The next signal will be green for the duration specified by this algorithm as soon as the Green Timer of an existing signal has reached zero. A photograph is taken as soon as the time value of a signal that will turn green reaches zero. This provides a time frame of 5 seconds (same as to the duration of the yellow signal timer) to process the image, detect the number of vehicles for each class, calculate the green signal time, and for the current signal and red signal timers signal to set the next signal.

3. **Simulation:** A custom simulation was created using Pygame to emulate real-life traffic scenarios. The simulation is capable of showing the system as a graphical representation and allows for comparison with current traffic systems. It has a four way intersection with four traffic signals,

DOI: 10.5281/zenodo.10441509

each equipped with a timer indicating the remaining time for signal changeovers from green to yellow, yellow to red, or red to green. In addition to each signal, the number of vehicles that have come through an intersection shall be displayed. Different types of vehicles are coming into the city from different directions, such as cars, bicycles, buses, trucks and rickshaws. Some vehicles in the rightmost lane are programmed to turn at a crossroads, so as to make it more realistic. On the basis of the vehicle's generation, the decision whether to turn or continue straight shall be made on a random basis. In addition, the elapsed time since simulation started shall be recorded using a timer. A sampling of the simulation carried out to manage traffic is shown in Figure 4-5.

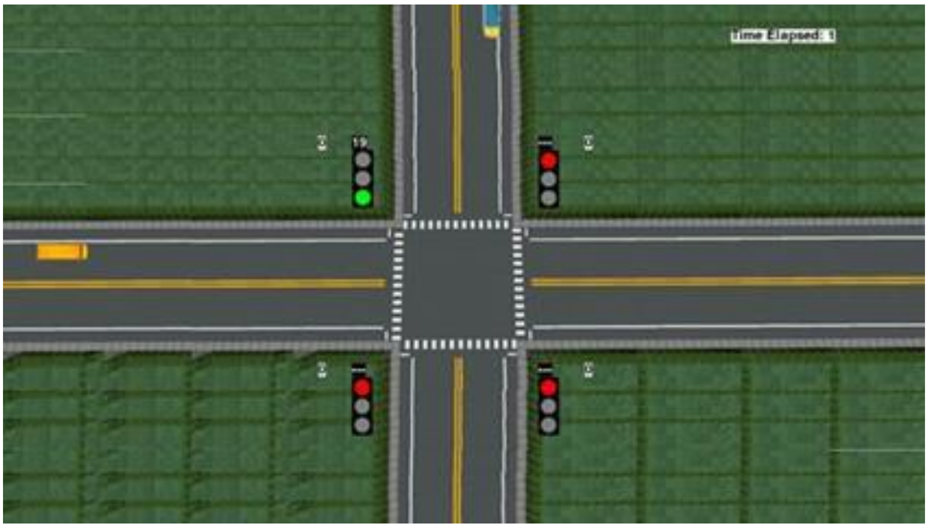

Figure 4: shows when normal working when no traffic is there.

Figure 5: shows simulation to manage traffic during peak time.

4. CONCLUSION

Managing traffic using AI is a promising application that utilizes the latest technologies to improve the efficiency of managing the traffic. With the increasing urbanization, traffic congestion has become a critical issue worldwide, and this project aims to address this challenge by implementing an intelligent system that can manage traffic flow. The project has a clear objective of detecting vehicles using AI and displaying the results in real-time. It also aims to provide a simulation that shows how the traffic will be managed in real-life scenarios. The system requirements specification is well-defined, and the project uses a range of technologies, including Python, OpenCV, TensorFlow, and Pygame, to achieve its objectives.

While there are limitations to the project, such as the need for clear images for accurate detection, the

DOI: 10.5281/zenodo.10441509

future scope of the project is vast, with the potential for further development and improvements. These developments can enable more efficient and effective traffic management, reduce traffic congestion, and improve road safety. It is essential to continue exploring and implementing these improvements to ensure that the project remains relevant and effective in the ever-changing world of traffic management.

REFERENCES

[1] Javaid, S., Sufian, A., Pervaiz, S., & Tanveer, M. (2018). Smart traffic management system using Internet of Things. In 2018 20th International Conference on Advanced Communication Technology (ICACT). https://doi.org/10.23919/icact.2018.8323770

[2] Singh, V., Unadkat, V., & Kanani, P. (2019). Intelligent Traffic Management System. International Journal of Recent Technology and Engineering, 8(3), 7592–7597. https://doi.org/10.35940/ijrte.c6168.098319

[3] User, S. (n.d.). International Research Journal of Engineering and Technology (IRJET) - Volume 4, Issue 01 Jan-2017. https://irjet.org/volume-4-issue-01-jan-2017

[4] Ouallane, A. A., Bahnasse, A., Bakali, A., & Talea, M. (2022). Overview of road traffic management solutions based on IoT and AI. Procedia Computer Science, 198, 518-523.

[5] Modi, Y., Teli, R., Mehta, A., Shah, K., & Shah, M. (2022). A comprehensive review on intelligent traffic management using machine learning algorithms. Innovative infrastructure solutions, 7(1), 128.

[6] Xu, J., & Wu, K. (2018). Living with artificial intelligence: A paradigm shift toward future network traffic control. Ieee Network, 32(6), 92-99.

[7] Bielli, M., Ambrosino, G., Boero, M., & Mastretta, M. (1991). Artificial intelligence techniques for urban traffic control. Transportation Research Part A: General, 25(5), 319-325.

[8] Misbahuddin, S., Zubairi, J. A., Saggaf, A., Basuni, J., Sulaiman, A., & Al-Sofi, A. (2015, December). IoT based dynamic road traffic management for smart cities. In 2015 12th International conference on high-capacity optical networks and enabling/emerging technologies (HONET) (pp. 1-5). IEEE.

[9] Ahmad, I., Noor, R. M., Ali, I., & Qureshi, M. A. (2017). The role of vehicular cloud computing in road traffic management: A survey. In Future Intelligent Vehicular Technologies: First International Conference, Future 5V 2016, Porto, Portugal, September 15, 2016, Revised Selected Papers 1 (pp. 123-131). Springer International Publishing.

[10] Ravish, R., & Swamy, S. R. (2021). Intelligent traffic management: A review of challenges, solutions, and future perspectives. Transport and Telecommunication Journal, 22(2), 163-182.

DOI: 10.5281/zenodo.10441509