A Deep Learning-based Model for Traffic Signal Control using the YOLO Algorithm

Oluwole Ayodeji Ayegbusi
Department of Computer Science, Adekunle Ajasin University, Akungba Akoko, Nigeria

Ayodeji Olusegun Akinwumi
Department of Computer Science, Adekunle Ajasin University, Akungba Akoko, Nigeria

OluwatosinLara Ogbeide
Department of Computer Science, Adekunle Ajasin University, Akungba Akoko, Nigeria

Alaba Olu Akingbesote
Department of Computer Science, Adekunle Ajasin University, Akungba Akoko, Nigeria

Joy Rotimi Obafemi
Department of Computer Science, Adekunle Ajasin University, Akungba Akoko, Nigeria.

Olatunde David Akinrolabu
Department of Computer Science, Adekunle Ajasin University, Akungba Akoko, Nigeria

Odunayo Rotiba
Department of Computer Science, Adekunle Ajasin University, Akungba Akoko, Nigeria

ABSTRACT
With increasing urbanization and population growth, traffic congestion has become a significant challenge in cities, resulting in delays, excessive fuel consumption, pollution, and stress-related health issues. This problem stems from the imbalance between transportation demand and road infrastructure supply. Conventional methods, such as using the manual or fixed-time control systems, have proven inadequate as they fail to adapt to real-time traffic conditions. This research therefore introduces a deep learning-based smart traffic signal control model utilizing the YOLO (You Only Look Once) algorithm to mitigate traffic congestion in real time. The model captures live images from traffic junction cameras, detects vehicles, calculates traffic density, and dynamically adjusts signal timers, prioritizing lanes with higher traffic. The system's architecture includes five phases: data acquisition, data pre-processing, training, signal control, and smart traffic control. The Open Images Dataset was utilized for the prototype demonstration, where the dataset was labeled and converted into a YOLO compatible format during the preprocessing phase to prepare it for training. The model was implemented using Python programming language. Evaluation revealed a 27% increase in traffic flow and a 50% reduction in vehicle waiting time. The research concluded that the YOLO based traffic control system provides a more effective solution to urban traffic congestion and is recommended for integration with CCTV cameras to facilitate efficient traffic management in cities.

General Terms
Deep Learning, Traffic Management, Object Detection, Artificial Intelligence, Pattern Recognition, Computer Vision

Keywords
YOLO Algorithm, Traffic Signal Control, Real-Time Traffic Management, Deep Learning, Smart Traffic Systems, Vehicle Detection

1. INTRODUCTION
Over the past two centuries, the global population has experienced significant growth, reaching over 8.2 billion as of 2024 [1]. This rapid population increase has led to a corresponding surge in the number of vehicles, particularly in urban areas worldwide. As cities expand to accommodate the growing population, the demand for transportation has also risen drastically [2][3]. Furthermore, recent technological advancements have enabled the automobile industry to scale up production, resulting in nearly 253 million cars being produced and sold annually [4], and statistical predictions indicate that the number of automobiles on the road will exceed two billion by 2030 [5]. This rapid growth of the human population, along with urbanization and the surge in vehicle production, has placed significant pressure on existing transportation infrastructure [6]. This has led to numerous challenges, with traffic congestion being one of the most critical issues [7][8].

Traffic congestion occurs when the desired speed of vehicles are reduced and the safe time headway, which is the time gap drivers maintain to ensure a safe distance from the vehicle ahead, increases [9][10]. It also occurs when the volume of vehicles on a road exceeds its capacity to accommodate them at an acceptable level of service [11]. As a result, vehicle speeds slow down, travel times increase, and traffic queues grow longer [12]. This issue has become a significant problem in both developed and developing countries, affecting individuals who rely on both private and public transportation [13]. Beyond the inconvenience of delays, exposure to traffic congestion also leads to a host of negative consequences, such as elevated noise and air pollution levels [14][15][16][17]. The increased pressure on passengers' time and the stress associated with traffic can exacerbate mental health issues, contributing to conditions like traffic stress syndrome. This syndrome arises when individuals are confined in a small space for long periods in congested traffic [18].

Traffic congestion continues to worsen globally, despite extensive research aimed at alleviating delays. Unfortunately, this issue shows no sign of slowing down, becoming a persistent challenge that undermines urban living standards [19][20]. In developing countries like Nigeria, poor infrastructure and limited road networks contribute to traffic congestion. For instance, inadequate or damaged roads, along with a lack of proper maintenance, create bottlenecks and slow traffic. In a bid to avoid these bad roads, drivers often crowd onto the few good roads available, which leads to overcrowding and worsens congestion.

The growing issue of traffic congestion in Nigeria's urban centers has reached critical levels, highlighting the need for urgent attention. A report cited in the work of Kanabe [21] quantifies the economic consequences of the traffic problem. According to the report, Lagos State, the commercial capital of Nigeria, stands to lose as much as 21 billion dollars each month if traffic congestion continues unaddressed by 2030. Other major cities in Nigeria, such as Ibadan, Benin City, Port Harcourt, Abuja, Kano, and Kaduna are also facing similar challenges. Traffic congestion is now a widespread issue in nearly every state capital in Nigeria [20][22][23][24]. Significant hours are lost due to traffic delays, which not only waste time but also leave individuals stressed and fatigued by the time they reach their destination [25][26]. Additionally, the harmful emissions from idling vehicles in traffic contribute to air pollution, causing various health problems, including respiratory issues, lung cancer, headaches, dizziness, and other related health issues [27]. Given these impacts, there is an urgent need to address and mitigate the effects of traffic congestion.

Existing methods for managing traffic congestion in Nigeria, similar to some other nations around the world, are the use of manual and conventional static traffic signal control systems [28] [29]. These systems have been in place for decades to regulate traffic flow at intersections and key locations, but they have proven to be limited in their effectiveness in addressing growing congestion. The manual method involves the use of traffic wardens to manage traffic. Traffic wardens are personnel who manage traffic at intersections, roadways, or busy areas, using hand signals, whistles, or verbal commands to ensure smooth flow and reduce congestion [30]. However, with the increasing number of vehicles, relying solely on human capacity has become impractical. Fatigue, inefficiency, and adverse weather conditions can further hinder their effectiveness, sometimes causing them to leave their posts [31]. A study highlighted the limitations of this manual system, revealing that traffic wardens often struggle to manage traffic effectively, particularly in urban cities [32]. The conventional traffic control system, otherwise called traffic lights [33], is another approach being used. This approach relies on fixed timing for the three colored signals: red, yellow, and green, which alternate at set intervals. The red light signals vehicles to stop, yellow warns that the light is about to change, and green allows vehicles to proceed [33][34][35]. The timing for these signals can range from 30 to 120 seconds; however, the system is not automated and does not adapt to real-time traffic on the roads [28]. This results in long wait times at intersections, which means the issue of congestion persists and continues to worsen.

The current approaches to traffic control indicate that existing efforts to alleviate congestion have not been sufficiently effective. To address these challenges, there is a need for more effective traffic control strategies that can reduce congestion, minimize vehicle wait times, and enhance the overall efficiency of transportation systems [36]. One promising solution is the application of advanced technologies, such as deep learning, in the development of intelligent traffic management systems [37].

The emergence of deep learning algorithms offers powerful solutions to complex classification and pattern recognition problems. These algorithms often deliver results comparable to or even surpassing human expertise [38][39][40][41]. Among the most widely used deep learning architectures is the Convolutional Neural Network (CNN) [42]. CNN have been extensively applied in tasks such as image recognition [43][44] object detection [45], speech analysis [46][47], traffic monitoring [48], biometric authentication [49][50], and pattern recognition [51][52][53] with considerable success. Their ability to analyze intricate patterns and process large amounts of data efficiently has made the algorithm integral to advancements in various domains, including traffic management. YOLO (You Only Look Once), is a real-time object detection algorithm [54][55], which is built on the principles of CNN and takes their capabilities a step further [56]. YOLO leverages the speed and accuracy of CNN to identify and localize objects in real time, making it ideal for dynamic applications like traffic management [57][58]. It is considered one of the most advanced object detection techniques, offering better generalization than other models for real-time use [56].

Building on these advancements, a deep learning-based model for traffic signal control using the YOLO algorithm is proposed. This model utilizes YOLO real-time object detection to dynamically adapt to traffic conditions, optimizing traffic flow and reducing vehicle wait times. An assessment of some developed systems revealed that, despite their strong results, a few problems still needed to be resolved. One such issue is the excessive vehicle wait time, which occurs when lanes with fewer vehicles are allocated more green time, leading to inefficiency. The proposed model addresses these challenges by dynamically adjusting signal timings based on real-time traffic data, ensuring efficient traffic flow and alleviating congestion, which will ultimately help in reducing congestion in Nigeria and other parts of the world.

2. RELATED WORKS

In recent years, several researchers have focused on traffic congestion, leading to significant contributions to the topic. A review of some of these works is presented as follows:

The work of Salama et al. [59] presented an Intelligent Cross Road Traffic Management System. In order to monitor the moving vehicles, the traffic management department installs these sensors at the proper distance. The traffic control cabinet receives this data and uses it to install software that determines the relative weight of each road. The system will enable a longer signal time for the congested road based on the assessed relative weight. A whole road is opened for these vehicles to pass first in order to manage emergency scenarios (such as the passing of ambulances, ministries, and other VIPs). RFID tags are installed on various vehicles (ambulance, ministry, VIP car) to manage emergency circumstances. The RFID reader next to the sensor picks up this RFID tag, and the information it collects is then forwarded to the traffic control cabinet where a decision is made. The flow of the traffic is ensured in this way.

Another study [60] proposed a system based on Radio Frequency Identification Devices (RFID). The work's main concept is to place the RFID on the booth side of the street. This aids in finding stolen vehicles by helping to track them and determine their location. Based on the amount of traffic that the RFID sensors installed on either side can detect, the traffic signal lights are adjusted. During busy hours, various vehicles are given priority. For instance, ambulances are available around the clock, but college buses are given preference in the morning and evening. However, as the car with the higher priority is allowed, which is an unfair practice, this worsens congestion. Vehicle turning points are not identified by this system. The vehicles in the lane with less traffic must wait longer.

The work Yadav et al. [61] on Adaptive traffic management system using IoT and Machine learning was also studied.

Camera sensors capture live lane details, which are then passed to a controller board that differentiates vehicles using TensorFlow. This board adjusts traffic signals and congestion lights based on the count, using the Min-Max Fairness algorithm for low average waiting time and the Round Robin algorithm for low traffic congestion.

Ogbeide et al. [62] introduced the Deep SARSA Replay model for real-time traffic control. The model leverages deep reinforcement learning, specifically the Deep SARSA (DSARSA) algorithm, integrated with experience replay. The system captures traffic data from all traffic lanes using a multilayer perceptron with two hidden layers, producing an approximated Q-value updated via the SARSA algorithm. By sampling random experiences from a replay buffer, the model stabilizes training and addresses diverse traffic conditions effectively. Experiments were conducted using the SUMO traffic simulator with 1000 episodes under low, medium, and heavy traffic flows. The results obtained revealed that the DSARSA replay model outperformed baseline DRL models in terms of learning stability, reward maximization, and traffic optimization. There were reductions in vehicle waiting times and an improved traffic flow.

A traffic control system that reacts to current inputs and outputs to determine the best choice to make in view of the present situation at a traffic junction was developed [37]. These decisions are which traffic lights to turn red and which turns green. Further review of the developed model shows that it analyzes the number of stopped and moving vehicles in each arm of a four-way intersection using an image analysis work flow. The state of the entire intersection is then managed using the information gathered to maintain the optimum possible traffic flow for all stationary and moving cars.

The work of [29] was also reviewed. The research presented a hybrid methodology for designing and implementing an intelligent traffic light control system. The methodology combines the Structured System Analysis and Design Methodology (SSADM) and Fuzzy Based Design Methodology, replacing the first step of the Fuzzy Based Design Methodology. The Fuzzy Logic-based methodology is chosen for its ability to develop both linear and non-linear systems for embedded control. The hybrid approach aims to examine existing systems, classify intersections, and design traffic control systems using fuzzy rules and simulation to eliminate traffic deadlock and improve road junction efficiency.

Miyim & Muhammed [63] developed a Smart traffic management system using Arduino and RFID tags in order to provide a smart way to monitor and control traffic congestion on roads and emergency service vehicles. The proposed system uses two Arduino circuit boards, the Uno board for radio frequency readers and the Mega board for light-emitting diodes and servo motors. It detects and scans emergency vehicles from left lane roads using built-in functions. However, this model was developed for emergency vehicles only, leaving other vehicles stranded in traffic.

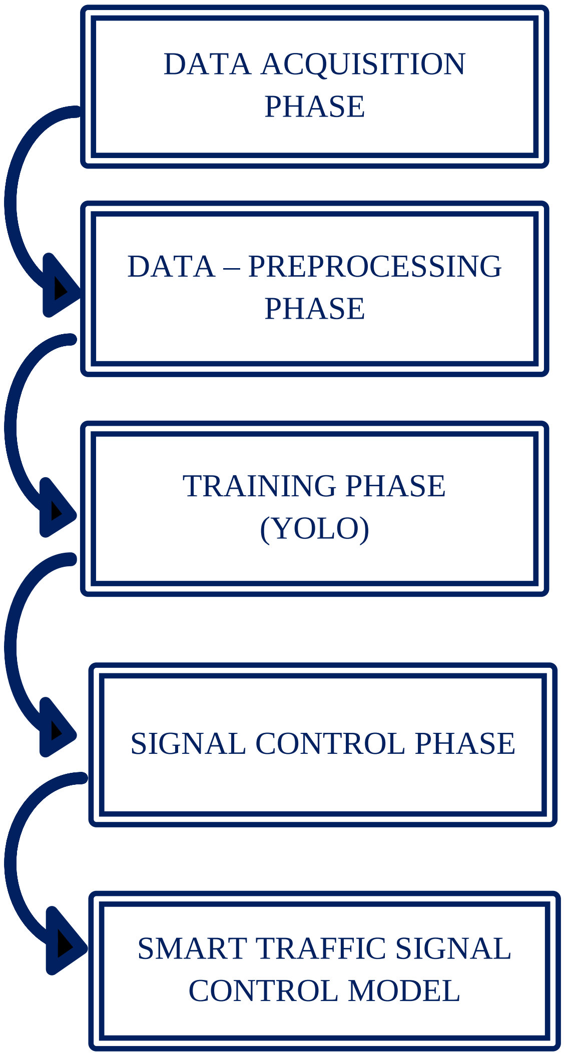
This research conducted by Khan et al. [64] introduced a lightweight neural network capable of accurately recognizing traffic signs with fewer trainable parameters. The model was trained using the German Traffic Sign Recognition Benchmark (GTSRB) and Belgium Traffic Sign (BelgiumTS) datasets. Experimental results demonstrated that the proposed model achieved an impressive accuracy of 98.41% on the GTSRB dataset and 92.06% on the BelgiumTS dataset. This performance surpassed several state-of-the-art models, including GoogleNet, AlexNet, VGG16, VGG19, MobileNetv2, and ResNetv2. Furthermore, the model outperformed these models on the GTSRB dataset by a margin of 0.1 to 4.20 percentage points and on the BelgiumTS dataset by a margin of 9.33 to 33.18 percentage points.

Building on existing research on traffic congestion, this study aims to contribute further to the ongoing efforts to optimize traffic management systems. The approach involves the application of a deep learning algorithm, specifically the YOLO algorithm, to enhance traffic signal control. Additionally, the study aims to improve the efficiency of traffic flow and reduce vehicle wait times by optimizing the system's ability to adapt to dynamic traffic conditions.

3. METHODOLOGY

This research utilizes the YOLO real-time object detection algorithm, a one-stage detector, in developing the smart traffic signal control system. As shown in Figure 1, the model's architecture consists of five key phases: (1) Data acquisition, (2) Data preprocessing, (3) Training, (4) Signal control, and (5) Development of the smart traffic signal control model. The performance of the developed system will be evaluated to assess its effectiveness. A detailed discussion of these phases follows in the next section.

Figure 1: The overview of the proposed Methodology



3.1 Data Acquisition

To train and test the model, real-time image datasets and pre-trained weights were utilized. The dataset was sourced from the Open Images Dataset, available at [https://storage.googleapis.com/openimages/web/index.html]. This dataset, widely used in various research studies, contains over 9 million annotated images, including those with traffic-related objects. To extract the relevant data for this research, the OIDv4 Toolkit was employed. Using the toolkit, 1,000 traffic-related images were selected from the Open Images Dataset, as shown in Figure 2. The frequency distribution of the extracted dataset is detailed in Table 1 below.

Table 1: Frequency of the extracted dataset.

S/n	IMAGE TYPE	FREQUENCY
1	Car	400
2	Bus	400
3	Truck	80
4	Rickshaw	50
5	Bike	50
6	Traffic Light	10
7	Fire Hydrant	10
	TOTAL	**1000**

3.2 Data Pre-processing

The following steps were followed in the data preprocessing phase:

Step 1: Labeling the Dataset: During data collection from the Open Images dataset, the labeled data was downloaded to facilitate the training of the YOLO algorithm. The labels were provided in the form of bounding boxes that identified the objects in the images and their locations. Each bounding box was annotated with a label corresponding to the class of the object. Labeling the dataset is crucial for enabling YOLO to recognize and classify objects in images based on patterns. It also ensures the algorithm can generalize to new, unseen images and accurately identify objects in real-world scenarios.

Step 2: Conversion of the Labeled Files to YOLO Format: The annotated files from the Open Images dataset were initially in Pascal VOC (Visual Object Classes) XML format, a widely used format for image annotation in object detection tasks.

These files contained information such as the image size (width, height), object class names, and the bounding box coordinates (xmin, xmax, ymin, ymax) for each object in the image. To train the YOLO algorithm, the Pascal VOC XML files were converted into YOLO format. The conversion process involved extracting the bounding box information (center coordinates, width, and height of the box) and class labels (represented by integer indices). The final YOLO format files included the object class labels and their bounding box coordinates. Two files, *7_class_test.txt* and *7_class_test_classes.txt*, were created, containing the bounding box coordinates and class labels. These files were used to train the YOLO algorithm, as illustrated in Figure 3.

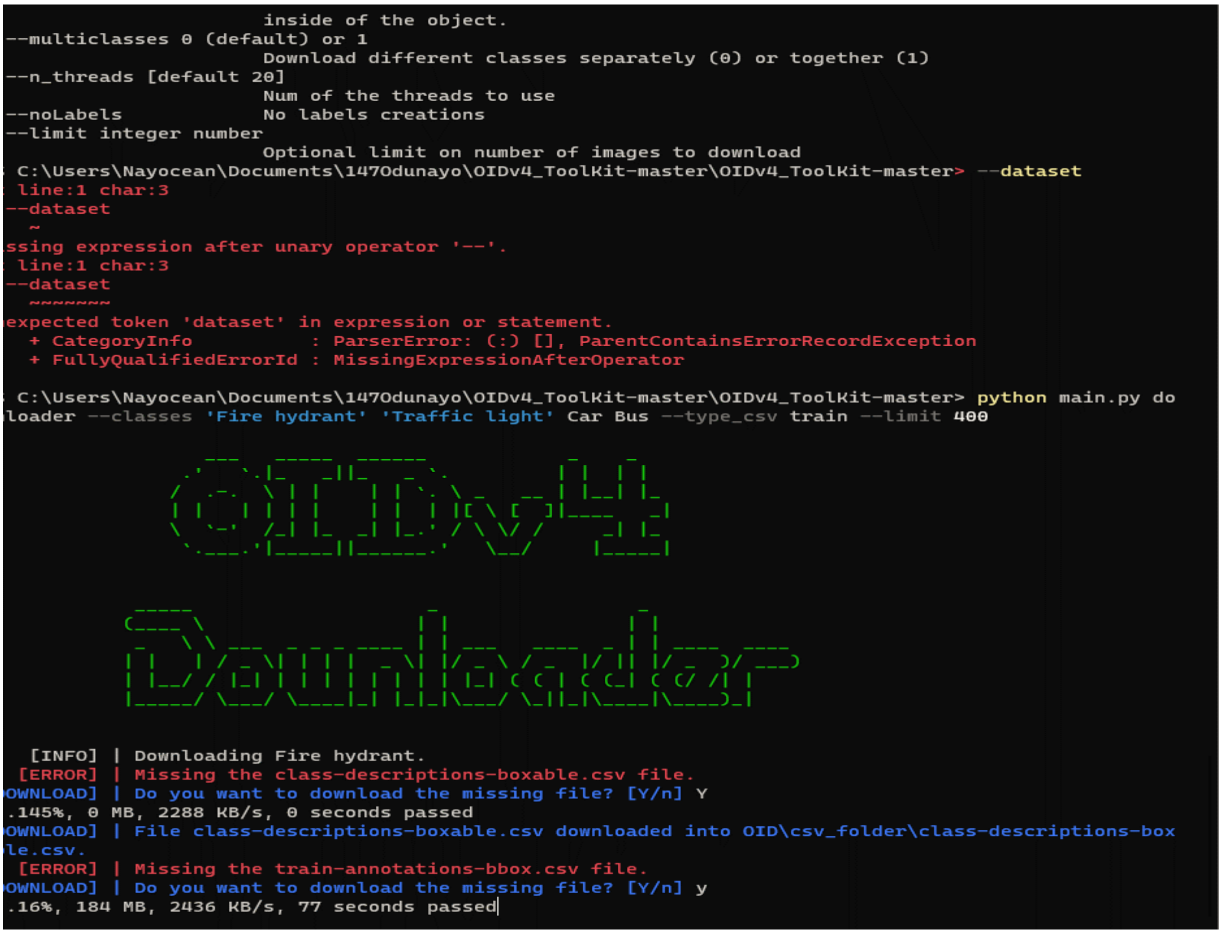


Figure 2: The Process in which traffic related images are being Extracted using the OIDV4 ToolKit

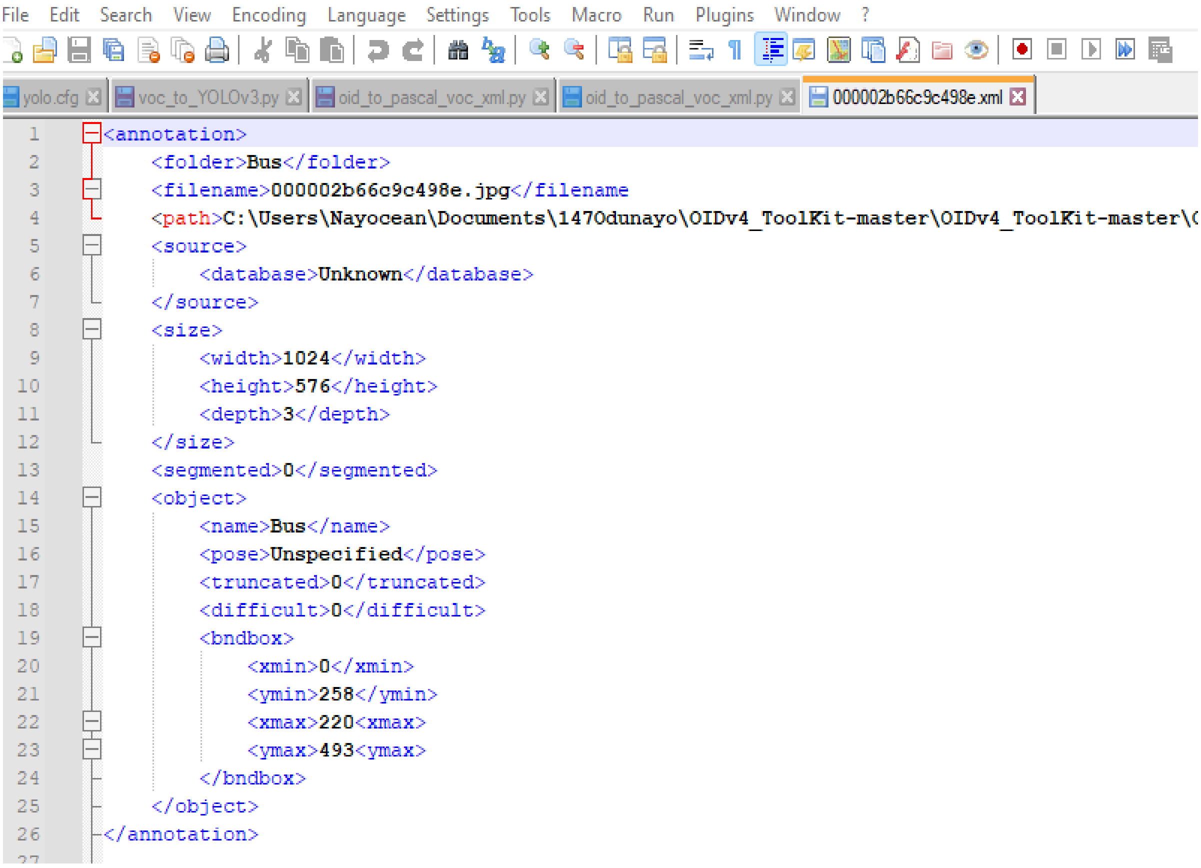


Figure 3: Pascal VOC Format

3.3 Training

The YOLO pre-trained weights used for training were obtained online from http://kaggle.com/datasets/valentynsichkar/yolo-coco-data. The dataset includes three key files: the YOLOv3 configuration file *(yolov3.cfg)*, the COCO dataset class file *(coco.txt)*, and the pre-trained weights file *(yolov3.weights)*.

The configuration file *(yolov3.cfg)* and the class labels file *(7_class_test_classes.txt)* were modified to meet the specific requirements of this model. Key modifications include:

i. **Output Neurons**: The number of output neurons in the last layer was adjusted to correspond with the number of classes the model was designed to detect. In this case, there were originally 7 classes (car, bus, traffic light, fire hydrant, truck, rickshaw, bike), but the model was customized to detect only 5 classes: 'Car, Bike, Bus, Truck, Rickshaw'.

ii. **Filters**: The number of filters in the convolutional layers was updated using the formula: *5 * (5 + number of classes) = 50*. This adjustment helped optimize the detection for the specific classes.

iii. **Image Size**: The input image size was reduced to 320x320 pixels, allowing the model to detect objects more quickly and with better accuracy.

Once the configuration changes were made, the model was trained using Darknet, an open-source neural network framework optimized for YOLO. Darknet enabled faster training by utilizing efficient GPU processing, as shown in Figure 3. The training process resulted in updated weights tailored to the custom configuration and classes.After training, the *yolov3.weights* file was imported into the code and used for vehicle detection. The model's output was processed using libraries such as Matplotlib and OpenCV. A confidence threshold was set to determine the minimum level of certainty required for successful object detection. Once the model was loaded and an image was inputted, the results were generated in JSON format, containing the detected objects along with their class labels and bounding box coordinates.

3.4 Signal Control

The signal control phase is responsible for managing the green and red signal timings based on the traffic density detected in the training phase. The green signal time for each signal is dynamically set based on the vehicle count, while the red signal times of the other signals are adjusted accordingly. The system operates cyclically, switching between signals based on their timers.

The YOLO algorithm, as described in the previous section, provides input to the signal control system in the form of vehicle detection data. This data, returned in JSON format, includes the detected object labels, confidence scores, and coordinates. The algorithm processes this information to calculate the total number of vehicles for each class of vehicle at the intersection. Using this data, the green signal time is determined for each signal, while the red signal times of the other signals are recalibrated.

The signal control algorithm can scale to any number of signals at an intersection. Initially, a default green signal time is set for the first signal in the first cycle. The model then calculates and assigns the green signal times for all subsequent signals in the first cycle, and similarly for all subsequent cycles. The system runs multiple threads: one for vehicle detection in each direction and another for managing the current signal's timer.

When the green light timer for the current signal reaches 5 seconds, the detection threads capture an image of the next direction. This image is then processed in the background to detect the number of vehicles in each class and to compute the green signal time for the next signal. During this process, the main thread continues counting down the current signal's timer. This ensures that the signal switching process is seamless and without delay.

Once the green timer of the current signal reaches zero, the next signal turns green for the duration determined by the model. At this point, the image capture process for the next green signal begins, giving the system a total of 10 seconds to process the image, detect the vehicles, calculate the green signal time, and adjust the timers for the next signal accordingly.

The green signal time (GST) for each signal is calculated using:

$$\text{GST} = \frac{\sum(\text{noOfVehicles} \times \text{AverageTime})}{noOfLanes + 1} \quad (1)$$

Where:

- GST is the green signal time
- *noOfVehicles* is the number of vehicles of each class of vehicle at the signal as detected by YOLO algorithm
- *averageTime* is the average time the vehicles of that class take to cross an intersection, and
- *noOfLanes* is the number of lanes at the intersection

3.5 Smart Traffic Signal Control Model

The model was built using Pygame to mimic real-life traffic and compare it with the current static system. It shows a 4-way intersection with four traffic lights. Each light has a timer that counts down the time until it changes from green to yellow, yellow to red, or red to green. The model also keeps track of how many vehicles pass through each intersection. Different types of vehicles, like cars, bikes, buses, trucks, and tricycles, come from all directions. To make the simulation more realistic, some vehicles in the rightmost lane randomly turn at the intersection. There is also a timer that tracks how much time has passed since the simulation started. Figure 4 shows a flowchart of how the model works.

3.6 Evaluation

To evaluate the performance of the Smart Traffic Signal Control model compared to the existing static system, certain conditions were measured, such as the waiting time of vehicles and the total number of vehicles that passed through the intersection. The model was run over an extended period, and the results demonstrated that the Smart Traffic Signal Control model outperformed the existing system in both reducing vehicle wait times and increasing the number of vehicles that passed through the intersection.

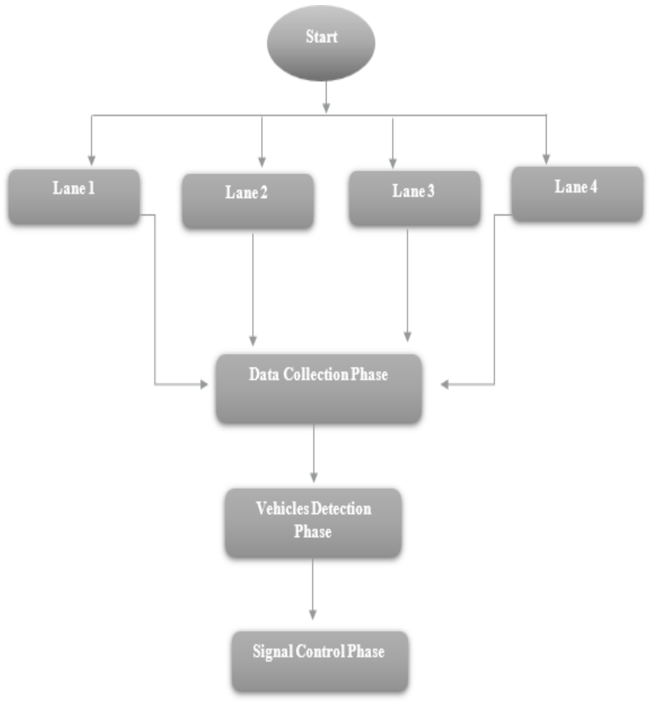


Figure 4: Flowchart of smart traffic signal control model

4. RESULTS AND DISCUSSION

The implementation of the deep learning-based traffic signal control model was carried out using Python 3.7, with essential libraries such as OpenCV, Matplotlib, TensorFlow, and Pygame. The development took place on a system running Windows 11, with hardware specifications including an Intel Core i9 processor, 32GB DDR4 RAM, and 1TB of disk space. Python was chosen due to its flexibility, stability, and strong support for object detection frameworks like YOLO, Fast R-CNN, and CNN, which are crucial for the task. Python's vast library ecosystem, including OpenCV for image processing and TensorFlow for model development, further aided in building the system.

For testing, 1,000 traffic images were downloaded from the Open Images dataset and formatted into YOLO format, which includes object class labels and bounding boxes.

The **signal control phase** of the model operates as follows:

Initial Setup: At the start, all traffic signals are assigned default values. The red signal time for the second signal (TS 2) is calculated based on the green and yellow times of the first signal (TS 1), as shown in Figure 5.

Signal Status and Timers: The leftmost column displays the status of each signal (red, yellow, or green), followed by the signal number and its respective red, yellow, and green timers (Figure 6) Initially, **TS 1** transitions from green to yellow. While the yellow timer counts down, vehicle detection results are processed, and a green time of 9 seconds is calculated for **TS 2**. Since this value is less than the minimum green time of 10 seconds, **TS 2** is assigned the minimum value of 10 seconds. When the yellow timer for **TS 1** reaches 0, **TS 1** turns red, and **TS 2** turns green, starting its countdown. The red signal time for **TS 3** is then updated as the sum of the yellow and green times of **TS 2**, which equals 15 seconds (5+10).

Subsequent Cycles: In the next cycle, **TS 1** again transitions from green to yellow. As the yellow timer counts down, new vehicle detection results are processed, and a green time of 25 seconds is calculated for **TS 2**. Since this value is within the allowed range (greater than the minimum green time and less than the maximum), **TS 2** is assigned a green time of 25 seconds, as shown in Figure 7. Once the yellow timer for **TS 1** reaches 0, **TS 1** turns red, and **TS 2** turns green, starting its countdown. The red signal time for **TS 3** is updated as the sum of the yellow and green times of **TS 2**, resulting in 30 seconds (5+25).

4.1 Smart Traffic Signal Control Model: Simulation Results

The Smart Traffic Signal Control system was tested using a simulation built in Pygame to mimic real-world traffic. The setup includes a four-way intersection with traffic signals, timers showing signal changes, and counters displaying the number of vehicles that have crossed. Different vehicle types approach from all directions, and some vehicles in the rightmost lane are programmed to turn at the intersection. A timer on the interface tracks the total elapsed time, creating a realistic environment to evaluate the system's performance. Below are the results from the simulation.

4.1.1 Initial State of the Model

As shown in Figure 8, at the start of the simulation, the traffic lights display both red and green signals. The green signal timer begins its countdown from a default value of 20 seconds. For the next signal, which is red, the timer remains blank initially until it reaches 10 seconds, at which point it becomes visible and starts counting down. Beside each signal, the number of vehicles that have crossed is displayed, all of which start at 0. Additionally, the total elapsed time since the simulation began is shown at the top-right corner of the screen.

4.1.2 Transition to Yellow Light

In Figure 9, as the signal transitions from green to yellow, the yellow light is displayed. During this transition, the red timer for the next signal becomes visible. When the red timer drops below 10 seconds, a countdown timer is displayed to alert vehicles, enabling them to prepare to move once the signal turns green.

4.1.3 Dynamic Green Signal Adjustments Based on Traffic

The model dynamically adjusts the green signal duration based on the traffic volume in each lane, as demonstrated in Figure 10. For example, when the number of vehicles in a lane is low, the model reduces the green signal time accordingly. In this instance, the green time is set to 10 seconds for a lightly trafficked lane, in contrast to the 30 seconds used in static systems. This prevents wasted time and ensures traffic flows efficiently.

4.1.4 Summary of Simulation Results

The simulation results are illustrated in Figure 11, where the model was run for a total of 5 minutes. During this period, the total number of vehicles passing through each lane and the entire intersection was recorded. This data shows the model's capability to improve traffic management by adapting to real-time conditions.

5. PERFORMANCE EVALUATION

To evaluate the performance of the developed Smart Traffic Signal Control Model, 10 simulations were conducted, each running for 5 minutes. These simulations were designed to compare the proposed model with the conventional static traffic lighting system. In the conventional system, a junction with four sides was considered. The default green light remained active for 60 seconds, while the red light had a waiting time of 180 seconds. Each side received a green signal for a fixed duration of 60 seconds and a red signal for 120 seconds, sequentially. The results presented in Table 2 and Table 3 compare the performance of the current static traffic system and the developed Smart Traffic Signal Control Model. Table 2 shows the performance metrics of the static system, while Table 3 shows the corresponding results achieved by the developed smart traffic signal model.

The performance of both the conventional traffic system and the developed model was assessed by analyzing the total number of vehicles that passed through each lane and the average waiting time for vehicles. The smart traffic model demonstrated significant improvements, passing 2,527 vehicles compared to 1,983 vehicles in the static system over a period of 1 hour and 15 minutes. This represents an additional 544 vehicles processed, reducing idle green signal time and waiting time.

To quantify the overall performance of the model, two metrics were utilized: traffic flow rate and waiting time.

1. **Traffic Flow Rate:**
   (a) The conventional system: 1,983 vehicles per hour
   (b) The Smart Traffic Signal Control Model: 2,527 vehicles per hour.
2. **Waiting Time:**
   (a) The conventional system: 120 seconds per vehicle.
   (b) The Smart Traffic Signal Control Model: 60 seconds per vehicle.

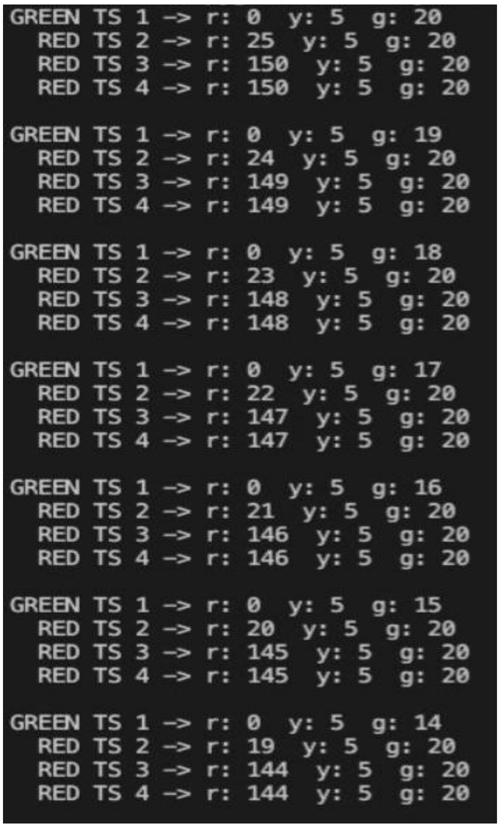
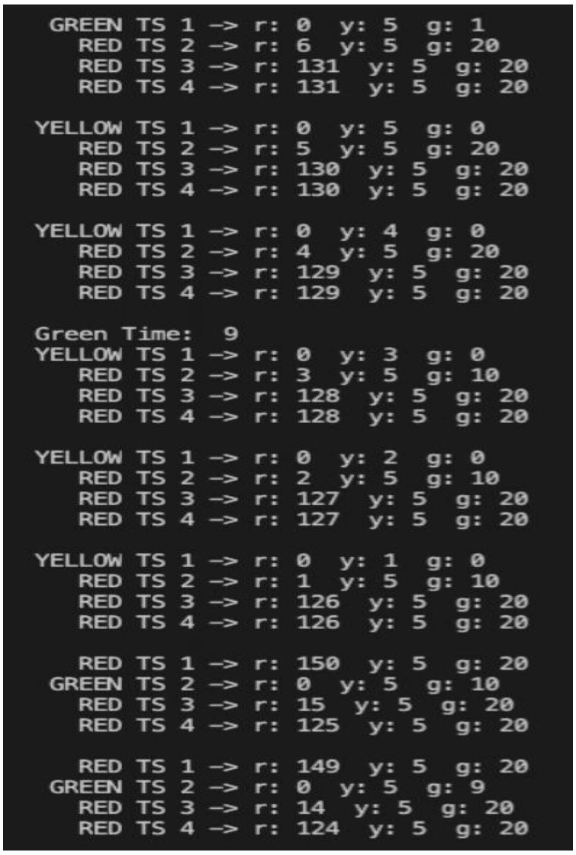
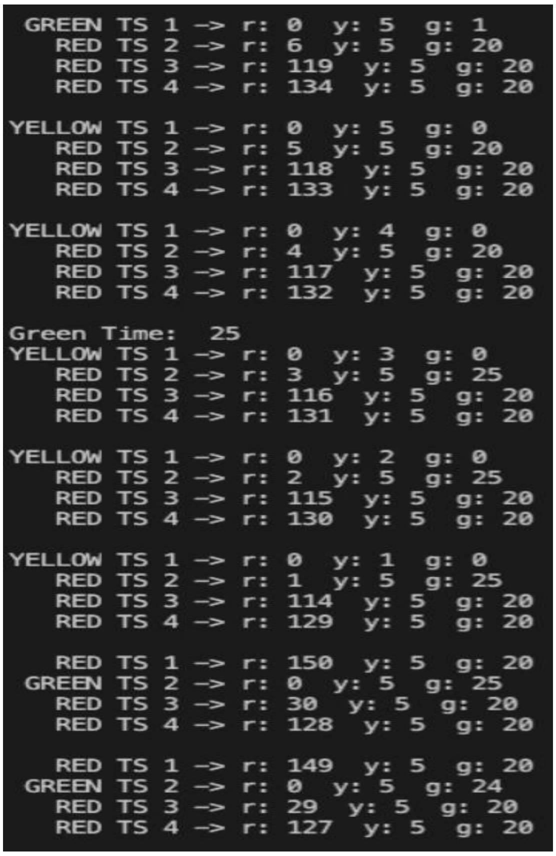


Figure 5: Default values for the first duration of all signals



```
  GREEN TS 1 -> r: 0  y: 5  g: 1
    RED TS 2 -> r: 6  y: 5  g: 20
    RED TS 3 -> r: 131  y: 5  g: 20
    RED TS 4 -> r: 131  y: 5  g: 20

 YELLOW TS 1 -> r: 0  y: 5  g: 0
    RED TS 2 -> r: 5  y: 5  g: 20
    RED TS 3 -> r: 130  y: 5  g: 20
    RED TS 4 -> r: 130  y: 5  g: 20

 YELLOW TS 1 -> r: 0  y: 4  g: 0
    RED TS 2 -> r: 4  y: 5  g: 20
    RED TS 3 -> r: 129  y: 5  g: 20
    RED TS 4 -> r: 129  y: 5  g: 20

 Green Time:  9
 YELLOW TS 1 -> r: 0  y: 3  g: 0
    RED TS 2 -> r: 3  y: 5  g: 10
    RED TS 3 -> r: 128  y: 5  g: 20
    RED TS 4 -> r: 128  y: 5  g: 20

 YELLOW TS 1 -> r: 0  y: 2  g: 0
    RED TS 2 -> r: 2  y: 5  g: 10
    RED TS 3 -> r: 127  y: 5  g: 20
    RED TS 4 -> r: 127  y: 5  g: 20

 YELLOW TS 1 -> r: 0  y: 1  g: 0
    RED TS 2 -> r: 1  y: 5  g: 10
    RED TS 3 -> r: 126  y: 5  g: 20
    RED TS 4 -> r: 126  y: 5  g: 20

    RED TS 1 -> r: 150  y: 5  g: 20
  GREEN TS 2 -> r: 0  y: 5  g: 10
    RED TS 3 -> r: 15  y: 5  g: 20
    RED TS 4 -> r: 125  y: 5  g: 20

    RED TS 1 -> r: 149  y: 5  g: 20
  GREEN TS 2 -> r: 0  y: 5  g: 9
    RED TS 3 -> r: 14  y: 5  g: 20
    RED TS 4 -> r: 124  y: 5  g: 20
```

Figure 6: Status of all signals



```
  GREEN TS 1 -> r: 0  y: 5  g: 1
    RED TS 2 -> r: 6  y: 5  g: 20
    RED TS 3 -> r: 119  y: 5  g: 20
    RED TS 4 -> r: 134  y: 5  g: 20

 YELLOW TS 1 -> r: 0  y: 5  g: 0
    RED TS 2 -> r: 5  y: 5  g: 20
    RED TS 3 -> r: 118  y: 5  g: 20
    RED TS 4 -> r: 133  y: 5  g: 20

 YELLOW TS 1 -> r: 0  y: 4  g: 0
    RED TS 2 -> r: 4  y: 5  g: 20
    RED TS 3 -> r: 117  y: 5  g: 20
    RED TS 4 -> r: 132  y: 5  g: 20

 Green Time:  25
 YELLOW TS 1 -> r: 0  y: 3  g: 0
    RED TS 2 -> r: 3  y: 5  g: 25
    RED TS 3 -> r: 116  y: 5  g: 20
    RED TS 4 -> r: 131  y: 5  g: 20

 YELLOW TS 1 -> r: 0  y: 2  g: 0
    RED TS 2 -> r: 2  y: 5  g: 25
    RED TS 3 -> r: 115  y: 5  g: 20
    RED TS 4 -> r: 130  y: 5  g: 20

 YELLOW TS 1 -> r: 0  y: 1  g: 0
    RED TS 2 -> r: 1  y: 5  g: 25
    RED TS 3 -> r: 114  y: 5  g: 20
    RED TS 4 -> r: 129  y: 5  g: 20

    RED TS 1 -> r: 150  y: 5  g: 20
  GREEN TS 2 -> r: 0  y: 5  g: 25
    RED TS 3 -> r: 30  y: 5  g: 20
    RED TS 4 -> r: 128  y: 5  g: 20

    RED TS 1 -> r: 149  y: 5  g: 20
  GREEN TS 2 -> r: 0  y: 5  g: 24
    RED TS 3 -> r: 29  y: 5  g: 20
    RED TS 4 -> r: 127  y: 5  g: 20
```

Figure 7: Allocation of green light timer

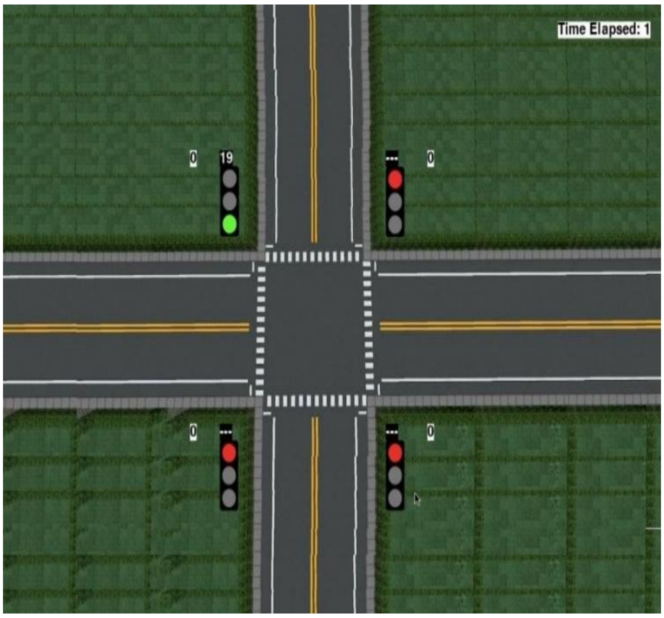


Figure 8: Start-up for smart traffic control signal model

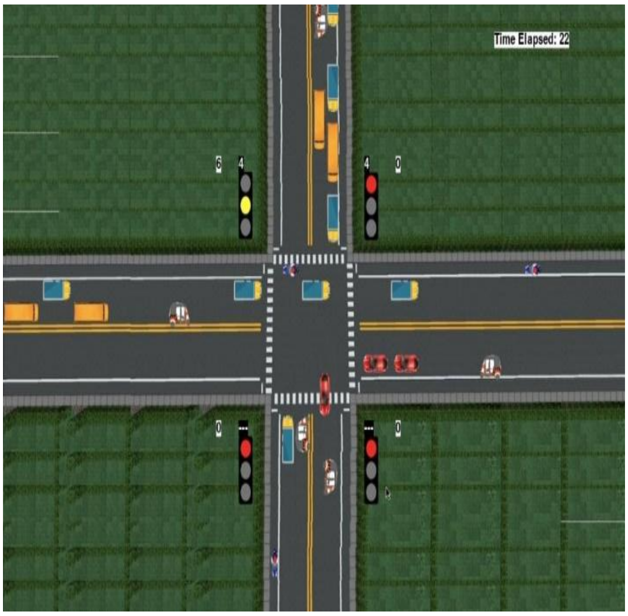


Figure 9: Countdown and numbers of vehicles

Figure 10: Dynamic adjustment of green signal duration based on traffic volume

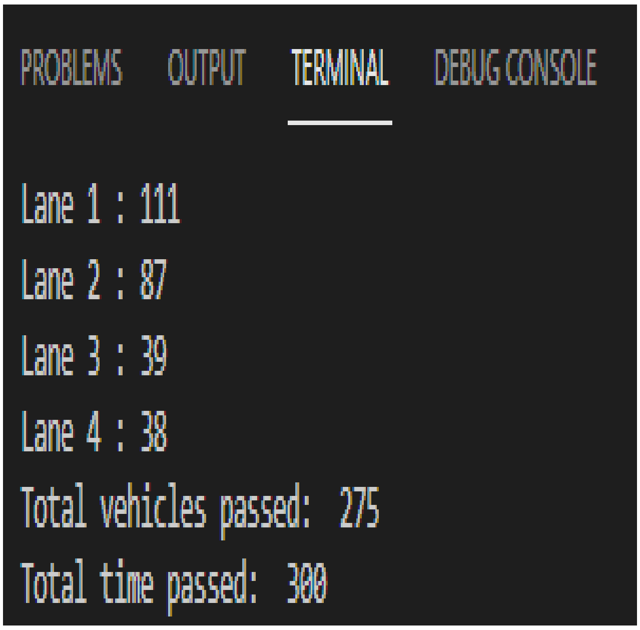


Figure 11: Result of the Smart Traffic Signal Control Model

Table 2: Results of the current static traffic system

Simulation Number	Vehicles in Lane 1	Vehicles in Lane 2	Vehicles in Lane 3	Vehicles in Lane 4	Total number of vehicles
1	70	52	20	12	154
2	56	48	49	25	178
3	73	53	63	62	251
4	74	44	65	71	254
5	90	32	25	41	188
6	95	71	15	14	195
7	73	63	43	24	203
8	54	47	10	67	178
9	81	29	88	37	235
10	39	52	34	22	147

Table 3: Results of the smart traffic signal control model

Simulation Number	Vehicles in Lane 1	Vehicles in Lane 2	Vehicles in Lane 3	Vehicles in Lane 4	Total number of vehicles
1	91	107	42	26	266
2	128	55	49	25	257
3	126	70	42	34	272
4	87	109	44	47	287
5	64	82	40	48	236
6	94	118	11	16	239
7	87	68	70	33	258
8	56	108	19	50	233
9	60	53	84	47	244
10	42	75	101	17	235

Furthermore, the Traffic Flow Improvement (TFI) of the smart traffic control model is therefore calculated using the equation below:

$$\textbf{\textit{TFI (\%)}} = \frac{\textit{Difference in vehicles passed}}{\textit{Initial NUmber of Vehicles}} \times \mathbf{100} \qquad \textbf{(2)}$$

$$\textit{TFI (\%)} = \frac{544}{1983} \times 100 = \textit{27\%}$$

Therefore, the Traffic Flow Improvement is 27%.

In addition, to calculate the Waiting Time Reduction (WTR), the equation below is utilized:

$$\textbf{WTR (\%)} = \frac{\textit{Difference in waiting time}}{\textit{Initial waiting time}} \times \mathbf{100} \qquad \textbf{(3)}$$

$$\textit{TFI (\%)} = \frac{60}{120} \times 100 = \textit{50\%}$$

Therefore, the Waiting Time Reduction is by 50%.

The Overall Improvement of the model is given by:

Overall Improvement (%) =TFI (%) + WTR (%) **(4)**

Substituting the values:

Overall Improvement (%) = 27% + 50% = 77%

Thus, the overall improvement of the model is 77%, demonstrating a significant enhancement in traffic flow efficiency and vehicle waiting time.

When comparing the results with other adaptive traffic control systems proposed in previous studies, the Smart Traffic Signal Control Model demonstrated superior performance. For instance, the work of Khusi [65] which introduced a smart traffic light system using image processing, showed comparatively lower effectiveness in optimizing traffic flow than the developed Smart Traffic Signal Control Model.

6. CONCLUSION

Traffic congestion is a significant challenge in urban areas worldwide, resulting in increased travel times, economic losses, environmental degradation, and social costs. This prompted the research to propose a Smart Traffic Signal Control Model using the YOLO real-time Object Detection algorithm. The developed model dynamically adjusts green signal durations based on traffic density, ensuring that heavily trafficked lanes are allotted more green time than less congested lanes. This adaptive approach reduces idle green signal time, minimizes delays, and lowers waiting times. The developed model also demonstrated considerable improvement not only over conventional traffic control systems but also over some adaptive solutions proposed by other researchers by effectively optimizing traffic signal timing and enhancing the flow of vehicles at intersections.

While the results of this research are promising, further studies can expand on the capabilities of the Smart Traffic Signal Control Model. Future work could involve integrating the system with technologies capable of tracking vehicles, prioritizing emergency vehicles such as ambulances, and identifying traffic violations. Additionally, enhancing the model to work seamlessly with smart city infrastructures, including IoT-enabled devices and CCTV systems, could further optimize traffic management and ensure safer, more efficient transportation systems.

7. REFERENCES

[1] United Nations Department of Economic and Social Affairs, *World Population Prospects 2022: Summary of Results*, no. July. New York: United Nations, 2022. doi: 10.18356/cd7acf62-en.

[2] A. R. Goetz, "Transport challenges in rapidly growing cities: is there a magic bullet?," *Transp. Rev.*, vol. 39, no. 6, pp. 701–705, 2019, doi: 10.1080/01441647.2019.1654201.

[3] A. G. L. . Neto, M. L. Galves, O. F. L. Júnior, and D. Tacla, "Challenges of urban transport problems and city logistics: Sao Paulo city center case," *Urban Transp.*, vol. XIV, pp. 133–142, 2008, doi: 10.2495/UT080131.

[4] D. A. Duwaer, "On deep reinforcement learning for data-driven traffic control," University of Technology, 2016. [Online]. Available: https://pure.tue.nl/ws/files/46945124/855429-1.pdf

[5] M. Gross, "A planet with two billion cars," *Curr. Biol.*, vol. 26, no. 8, pp. R307–R310, 2016, doi: 10.1016/j.cub.2016.04.019.

[6] X. Wang, R. Zeng, F. Zou, L. Liao, and F. Huang, "STTF : An Efficient Transformer Model for Traffic Congestion Prediction," *Int. J. Comput. Intell. Syst.*, vol. 16, no. 2, pp. 1–16, 2023, doi: 10.1007/s44196-022-00177-3.

[7] A. A. F. Al-sabaawi, "Traffic Congestion Control based In-Memory Analytics : Challenges and Advantages," *nternational J. Comput. Appl.*, vol. 170, no. 6, pp. 39–42, 2017.

[8] P. Sangaradasse and S. Eswari, "Importance of Traffic and Transportation Plan in the Context of Land Use Planning for Cities – A Review," *Int. J. Appl. Eng. Res.*, vol. 14, no. 9, pp. 2275–2281, 2019.

[9] C. Arti, G. Sharad, K. Pradeep, P. Chinmay, and S. S. Kumar, "Urban traffic congestion: its causes-consequences- mitigation," *Res. J. Chem. Environ.*, vol. 26, no. 12, pp. 164–176, 2022, doi: 10.25303/2612rjce1640176.

[10] R. Yu, Y. Zhang, L. Wang, and X. Du, "Time headway distribution analysis of naturalistic road users based on aerial datasets," *J. Intell. Connect. Veh.*, vol. 5, no. 3, pp. 149–156, 2022, doi: 10.1108/JICV-01-2022-0004.

[11] S. Rosenbloom, "Peak-period traffic congestion: A state-of-the-art analysis and evaluation of effective solutions," *Transportation (Amst).*, vol. 7, no. 2, pp. 167–191, 1978, doi: 10.1007/BF00184638.

[12] N. Lanke and S. Koul, "Smart Traffic Management System," *Int. J. Comput. Appl.*, vol. 75, no. 7, pp. 19–22, 2013, doi: 10.5120/13123-0473.

[13] A. A. Obiri-yeboah, A. S. Amoah, and M. S. Gbeckor-kove, "Analysis of Congestion on Some Road Link Sections Using Roadside Friction in Congestion Index in Kumasi," *Int. J. Traffic Transp. Eng.*, vol. 10, no. 1, pp. 31–40, 2020, doi: 10.7708/ijtte.2020.10(1).03.

[14] A. Sharma, V. Madan, V. Bhargav, and N. Gulati, "Smart City Traffic Control System: A Literature Review," in *14th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2024, pp. 36–40. doi: 10.1109/Confluence60223.2024.10463364.

[15] C. D. Higgins, M. N. Sweet, and P. S. Kanaroglou, "All minutes are not equal: travel time and the effects of congestion on commute satisfaction in Canadian cities," *Transportation (Amst).*, vol. 45, no. 5, pp. 1249–1268, 2018, doi: 10.1007/s11116-017-9766-2.

[16] P. M. Evaluation, T. Congestion, U. Roads, A. Journal, and A. Sciences, "Evaluation of Traffic Congestion in an Urban Road: A Review," *ABUAD J. Eng. Appl. Sci.*, vol. 2, no. 2, pp. 1–7, 2024, doi: 10.53982/ajeas.2024.0202.01-j.

[17] S. Xu, C. Sun, and N. Liu, "Road congestion and air pollution -Analysis of spatial and temporal congestion effects," *Sci. Total Environ.*, vol. 945, no. 2024, p. 173896, 2024, doi: 10.1016/j.scitotenv.2024.173896.

[18] R. McConnell *et al.*, "Childhood incident asthma and traffic-related air pollution at home and school," *Environ. Health Perspect.*, vol. 118, no. 7, pp. 1021–1026, 2010, doi: 10.1289/ehp.0901232.

[19] S. P. Mohanty, U. Choppali, and E. Kougianos, "Everything you wanted to know about smart cities," *IEEE Consumer Electronics Magazine*, vol. 5, no. 3, pp. 60–70, 2016. doi: 10.1109/MCE.2016.2556879.

[20] J. O. Ukpata and A. A. Etika, "Traffic Congestion in Major Cities of Nigeria," *J. Eng. Technol.*, vol. 2, no. 8, pp. 1433–1438, 2012.

[21] M. Kanabe, "Traffic Congestion: Lagos to lose $ 21 billion monthly by 2030 – Expert," *PremiumTimes*, 2022.

[22] K. Olagunju, "Evaluating Traffic Congestion in Developing Countries – A Case Study of Nigeria," in *2015 Chartered Institute of Logistics and Transport (CILT) Africa Forum*, 2015, pp. 1–28.

[23] J. O. Ukpata and A. A. Etika, "Traffic Congestion in Major Cities of Nigeria Traffic Congestion in Major Cities of Nigeria," *Int. J. Eng. Technol.*, vol. 2, no. 8, pp. 1433–1438, 2012.

[24] R. A. Olawepo, Y. A. Ahmed, and A. Asaju, "Planning For Sustainability: Transportation and Land Use in Ilorin, Nigeria," *J. Art, Archit. Built Environ.*, vol. 3, no. 2, pp.

18–30, 2020, doi: 10.32350/jaabe.32.02.

[25] G. C. Oweisana and V. N. Ordua, "Influence of Traffic Congestion on Psychological Stress and Pro-Social Behaviour among Commuters in Port-Harcourt Metropolis," *Nnamdi Azikwe Journals*, pp. 1–15, 2022.

[26] D. Stokols and R. W. Novaco, "Traffic Congestion, Type A Behavior and Stress," *J. f Appl. Psychol.*, vol. 63, no. 4, pp. 467–480, 1978, doi: 10.1037/0021-9010.63.4.467.

[27] A. O. Awosusi and I. . Akindutire, "Urban Traffic Congestion and Its Attendant Health Effects," *An Int. Multi-Disciplinary J.*, vol. 4, no. 17, pp. 434–446, 2010.

[28] C. E. Ojieabu, "Developed Automated Vehicle Traffic Light Controller System for Cities in Nigeria," *J. Adv. Sci. Eng.*, vol. 1, no. 1, pp. 19–25, 2018, doi: 10.37121/jase.v1i1.6.

[29] A. Jibrin, "Design and Simulation of an Intelligent Road Traffic Control System," *Eng. Res. J.*, vol. 2, no. 11, pp. 1–17, 2022.

[30] L. I. Igbinosun and O. Izevbizua, "Some control strategies for road traffic flow in Nigeria," *Int. J. Stat. Appl. Math.*, vol. 5, no. 4, pp. 56–61, 2020, [Online]. Available: www.dft.gov.uk/traffic-count/.

[31] D. B. O. Prince and O. Desmond, "Occupational Hazards Associated with Traffic Warden in Port Harcourt Metropolis, Rivers State, Nigeria.," *Asian J. Res. Nurs. Heal.*, vol. 5, no. 1, pp. 203–216, 20225.

[32] I. Rasdi, N. F. Din, N. Roni, A. S. Nizam, and I. Isa, "Fatigue Among Traffic Police Officers in Metropolitan City: Exploring Factors of Noise Exposure and Work Stressors," *Malaysian J. Med. Heal. Sci.*, vol. 16, no. Supp 11, pp. 109–117, 2020.

[33] G. P. Felicio, L. C. Grepo, V. F. Reyes, and L. C. Yupingkun, "Traffic Light Displays and Driver Behaviors: A Case Study," *Procedia Manuf.*, vol. 3, no. 2015, pp. 3266–3273, 2015, doi: 10.1016/j.promfg.2015.07.879.

[34] Department for Transport UK, "Highway Code: Light signals controlling traffic," The Highway Code. Accessed: Dec. 19, 2024. [Online]. Available: https://assets.publishing.service.gov.uk/media/560aa3f9e5274a036900001c/the-highway-code-light-signals-controlling-traffic.pdf

[35] Q. F. Zhou and J. Yang, "Design and Implementation of Traffic Lights Control System Based on FPGA," in *Proceedings of the International Conference on Chemical, Material and Food Engineering*, 2015, pp. 822–825. doi: 10.2991/cmfe-15.2015.195.

[36] N. Tsalikidis, A. Mystakidis, P. Koukaras, M. Ivaškeviˇ, D. Ioannidis, and C. Tjortjis, "smart cities Urban Traffic Congestion Prediction : A Multi-Step Approach Utilizing Sensor Data and Weather Information," *Smart Cities*, vol. 7, pp. 233–253, 2024, doi: 10.3390/smartcities7010010.

[37] A. Amin, S. Bahnasy, A. Elhadidy, and M. Elattar, "Real-time 4-way Intersection Smart Traffic Control System," *2nd Nov. Intell. Lead. Emerg. Sci. Conf. NILES 2020*, pp. 428–433, 2020, doi: 10.1109/NILES50944.2020.9257949.

[38] R. Chauhan, K. K. Ghanshala, and R. . Josh, "Convolutional Neural Network (CNN) for Image Detection and Recognition," in *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, IEEE, 2018, pp. 278–282. doi: 10.1109/ICSCCC.2018.8703316.

[39] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, "Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach," *Procedia Comput. Sci.*, vol. 132, pp. 679–688, 2018, doi: 10.1016/j.procs.2018.05.069.

[40] A. O. Ige and M. H. M. Noor, "A lightweight deep learning with feature weighting for activity recognition," *Comput. Intell.*, vol. 39, no. 2, pp. 315–343, 2022, doi: https://doi.org/10.1111/coin.12565.

[41] Z. Sun, "Pattern Recognition in Convolutional Neural Network (CNN)," *Appl. Intell. Syst. Multi-modal Inf. Anal.*, vol. 138, pp. 295–302, 2022.

[42] A. O. Akinwumi, A. O. Akingbesote, O. O. Ajayi, and F. O. Aranuwa, "Detection of Distributed Denial of Service (DDoS) attacks using convolutional neural networks," *Niger. J. Technol.*, vol. 41, no. 6, pp. 1017–1024, 2022, doi: 10.4314/njt.v41i6.12.

[43] H. M. Z. Haque, "Aortic Valve Segmentation using Convolutional Neural Network with Skip Mechanism," *Commun. Appl. Electron.*, vol. 7, no. 29, pp. 1–5, 2019.

[44] F. O. Aranuwa and O. B. Fawehinmi, "Classification Model For Iris Images Using Convolutional Neural Network (CNN)," in *Proceedings of the 32nd Accra Multidisciplinary Cross-Border Conference (AMCBC)*, Accra, 2022, pp. 7–22.

[45] C. Szegedy, A. Toshev, and D. Erhan, "Deep Neural Networks for Object Detection," *Adv. neural Inf. Process. Syst.*, pp. 2553–2561, 2013, doi: 10.3928/19404921-20140820-01.

[46] Y. Zhang *et al.*, "Towards end-to-end speech recognition with deep convolutional neural networks," *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, vol. 08-12-Sept, no. September, pp. 410–414, 2016, doi: 10.21437/Interspeech.2016-1446.

[47] D. Issa, M. Fatih Demirci, and A. Yazici, "Speech emotion recognition with deep convolutional neural networks," *Biomed. Signal Process. Control*, vol. 59, p. 101894, 2020, doi: 10.1016/j.bspc.2020.101894.

[48] X. Luo, R. Shen, J. Hu, J. Deng, L. Hu, and Q. Guan, "A Deep Convolution Neural Network Model for Vehicle Recognition and Face Recognition," *Procedia Comput. Sci.*, vol. 107, no. Icict, pp. 715–720, 2017, doi: 10.1016/j.procs.2017.03.153.

[49] A. Ucar, "Deep Convolutional Neural Networks for facial expression recognition," *Proc. - 2017 IEEE Int. Conf. Innov. Intell. Syst. Appl. INISTA 2017*, pp. 371–375, 2017, doi: 10.1109/INISTA.2017.8001188.

[50] A. E. Ebitigha, O. O. Ajayi, O. D. Akinrolabu, A. Adegbite, J. Obafemi, and J. K. Ogunleye, "Facial Appearance Analysis for Age Group Prediction Using Convolutional Neural Network," in *International Conference on Science, Engineering and Business for Driving Sustainable Development Goals*, 2024.

[51] A. O. Akinwumi, A. O. Ige, J. R. Obafemi, O. D. Akinrolabu, and B. O. Akingbesote, "CBDS-ConvNet : A

Cyber-Bullying Detection Model using Convolutional Neural Network," *Commun. Appl. Electron.*, vol. 7, no. 40, pp. 11–21, 2025, doi: 10.5120/cae2025652905.

[52] J. Natarajan, W. Wang, Y. Jiang, Z. Zhang, H. Ye, and L. Kuang, "Generative-CNN for Pattern Recognition in Finance," in *ICAIF 2024 - 5th ACM International Conference on AI in Finance*, 2024, pp. 142–149. doi: 10.1145/3677052.3698622.

[53] A. Dibouliya and V. Jotwani, "Traffic Density Monitoring Control System Using Convolution Neural Network," *Int. J. Sci. Res. Eng. Dev.*, vol. 6, no. 5, pp. 31–39, 2024, [Online]. Available: www.ijsred.com

[54] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, p. 788. doi: 10.1145/3243394.3243692.

[55] S. Gothane, "A Practice for Object Detection Using YOLO Algorithm," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 7, no. 2, pp. 268–272, 2021, doi: 10.32628/cseit217249.

[56] V. Viswanatha, R. Chandana, and A. C. Ramachandra, "Real Time Object Detection System with YOLO and CNN Models: A Review," vol. XIV, no. 7, pp. 144–151, 2022, [Online]. Available: http://arxiv.org/abs/2208.00773

[57] R. Cheng, "A survey: Comparison between Convolutional Neural Network and YOLO in image identification," in *Journal of Physics: Conference Series*, 2019, p. 012139. doi: 10.1088/1742-6596/1453/1/012139.

[58] P. Kunekar, Y. Narule, R. Mahajan, S. Mandlapure, and E. Mehendale, "Traffic Management System Using YOLO Algorithm," in *Engineering Proceedings*, 2023, p. 210.

[59] A. S. Salama, B. K. Saleh, and M. M. Eassa, "Intelligent cross road traffic management system (ICRTMS)," *ICCTD 2010 - 2010 2nd Int. Conf. Comput. Technol. Dev. Proc.*, no. Icctd, pp. 27–31, 2010, doi: 10.1109/ICCTD.2010.5646059.

[60] V. Samhita, S. Prachi, A. Rajesh, and P. Aparna, "Intelligent Traffic Density Monitoring Using RFID," no. March, 2018.

[61] A. Yadav, V. More, N. Shinde, M. Nerurkar, and N. Sakhare, "Adaptive Traffic Management System Using IoT and Machine Learning," *Int. J. Sci. Res. Sci. Eng. Technol.*, no. January, pp. 216–229, 2019, doi: 10.32628/ijsrset196146.

[62] O. L. Ogbeide, O. Olabode, A. Oluwatoyin, and O. Boyinbode, "Deep Sarsa Replay model for real Time Traffic Control," *Int. J. Adv. Eng. Manag.*, vol. 6, no. 11, pp. 398–408, 2024, doi: 10.35629/5252-0611398408.

[63] A. M. Miyim and M. A. Muhammed, "Smart traffic management system using Arduino and RFID Tags," *2019 15th Int. Conf. Electron. Comput. Comput. ICECCO 2019*, vol. 6, no. 2, pp. 377–383, 2019, doi: 10.1109/ICECCO48375.2019.9043219.

[64] M. A. Khan, H. Park, and J. Chae, "A Lightweight Convolutional Neural Network (CNN) Architecture for Traffic Sign Recognition in Urban Road Networks," *Electron.*, vol. 12, no. 8, 2023, doi: 10.3390/electronics12081802.

[65] Khusi, "Smart Control of Traffic Light System using Image Processing," in *International Conference on Current Trends in Computer, Electrical, Electronics and Communication (ICCTCEEC-2017)*, 2017, pp. 99–103.