Real-Time Traffic Management System

Tarandeep Singh Mandhiratta
Master of Applied Computing
Wilfrid Laurier University
Waterloo, Canada
mand4710@mylaurier.ca

***Abstract*— People nowadays choose to commute in their own private vehicles rather than via public transportation, resulting in a greater number of private vehicles on the road. It results in traffic congestion on all roadways. In such a circumstance, individuals cannot be forced to reduce their use of private vehicles, but we may control traffic flow in such a way that congestion is not alleviated. With ever-increasing traffic congestion around the world, traditional approaches to traffic control are ineffective for seamless commuting. As a result, a solution that is widely acceptable and leads to better traffic management is required. A rapid advancement in artificial intelligence has resulted in more powerful tools, which can learn semantic, high-level, deeper features that can address the underlying problems of traditional architectures. Due to the growing population and automobiles in cities, traffic congestion is becoming one of the most critical issues. Increasing traffic congestion makes it imperative to calculate traffic density in real-time for improved traffic management through better signal control. Traditional traffic management strategies are only successful when there is less traffic; however, as the number of cars on one edge of the way grows more than on the other, this method fails. As a consequence, we wish to convert from static to dynamic signal altering in the traffic signal system so that we can monitor and manage signals in real time. Traffic congestion not only add to the time and frustration of drivers, but they often increase carbon emissions and pollution. Metropolitan areas are indeed the ones most afflicted by it, despite the fact that it seems to be everywhere. Its ever-increasing nature necessitates the calculation of real-time road traffic density for improved signal control and traffic management. One of the important aspects controlling traffic flow is the traffic controller. As a result, there is a need to improve traffic management in order to better meet this growing demand. Utilizing computer vision and AI, our proposed system tries to calculate traffic density using live pictures from cameras at traffic intersections. It moreover concentrates on the mechanism for altering traffic lights depending on traffic density to alleviate traffic congestion and pollution.**

Keywords—— YOLO, Intelligent transport systems, Object detection, Darknet, Traffic control, Computer Vision

I. INTRODUCTION

The Traffic Management System was established in 1972 to centralize supervision of the Twin Cities metro area's motorway system. The Traffic Management System aims to provide passengers with a quicker, safer journey on metro area highways by maximizing the use of current road capacity, effectively handling accidents and special events, providing traveler information, and creating incentives for ride sharing. Cities and traffic have coexisted since the dawn of significant human settlements. The same dynamics that drive people to assemble in large cities also cause traffic congestion on city streets that is often unpleasant. The Traffic Management System was built in 1972 to oversee the Twin Cities metro area's highway system from a single location [1]. By maximizing the utilization of current road capacity, effectively handling accidents and special events, providing passenger information, and giving incentives for ride sharing. Since the beginning of substantial human settlements, cities and traffic have coexisted. The same forces that lead people to congregate in large cities also produce unpleasant traffic congestion on city streets. Cities are the engines of economic development in any country [2]. The transportation system facilitates movement and provides a means of reaching locations. An insufficient transportation system stifles economic activity and obstructs progress.

As the number of roads and automobiles has grown, traffic management has become an essential part of the intelligent transportation system. With ever-increasing traffic congestion around the world, traditional approaches to traffic control are ineffective for seamless commuting. As a result, a solution that is widely acceptable and leads to better traffic management is required [3]. The signal switches at its predetermined regular interval in today's traditional technique, but the number of vehicles on the road at each light does not remain the same, so the static approach fails. In this situation, if the signal switches at its normal interval, the heavily populated side of the road will always be completely filled.

Currently, there are three common systems for traffic control in use:

1) Manual Traffic Control: As the name implies, manual traffic control necessitates the use of human resources. A needed area for traffic control is assigned to the traffic police. To manage traffic, traffic cops carry a signboard, a sign light, and a whistle.

2) Static timer-controlled traffic lights: These are regulated by fixed timers. The timer is set to a fixed integer number. According on the timer setting, the lights automatically transition between red and green.

3) Electronic Sensors: Installing proximity sensors or loop detectors on the road is another sophisticated option. This sensor collects information regarding traffic in the area. The traffic lights are managed based on sensor data.

These traditional procedures have several disadvantages. The manual control mechanism requires a significant quantity of personnel. We cannot have traffic police manually managing traffic in all regions of a major metropolitan area due to a lack of resources. As a result, a better traffic management system is required. Static traffic control employs a traffic signal with a countdown for each phase that is set in stone and does not respond to real-time traffic on the road [13]. Since the extraction of high-quality data is frequently reliant on complex and costly technologies, and therefore limited budget will lower the number of facilities, precision and availability are often in conflict when utilizing electronic sensors, such as sensing devices or loop sensors [4] [5]. Furthermore, since most sensors have a limited effective range, comprehensive coverage throughout a network of sites generally necessitates a large number of sensors.

II. EXISTING TRAFFIC MANAGEMENT SYSTEM

With global traffic congestion on the rise, and traditional approaches to managing it ineffective for smooth commuting, a solution that can be widely accepted and lead to better traffic management is required. In today's society, when technology has broken down all barriers, it is now possible to solve most human problems, including traffic congestion. Traffic congestion has risen dramatically in recent years, resulting in negative consequences such as road rage, accidents, air pollution, fuel waste, and, most importantly, avoidable delays [6]. The fact that traffic light distribution in many cities throughout the world is still based on a timer prompted the development of a new approach.

The Timer Approach has the disadvantage that even if one of the roads has fewer traffic, green signal is still assigned to that road until its timer value goes to 0, whereas traffic on another road sees red signal at that time [7]. Commutators face congestion and lost time as a result of this. The majority of current systems are not automated and are susceptible to human mistakes. Many projects are growing to transform city's conventional transportation systems to new modern systems, a good example of this is the dynamic automatic transportation system [14]. Several steps have been used to develop a method that can do dynamic traffic signal supervision, which means that the sign changing conditions will not be predetermined, but will instead be determined by the amount of carriers on both sides of the road.

Currently, there are three common systems for traffic control in use:

1) Manual Traffic Control: As the name implies, manual traffic control necessitates the use of human resources. A needed area for traffic control is assigned to the traffic police. To manage traffic, traffic cops carry a signboard, a sign light, and a whistle.

2) Static timer-controlled traffic lights: These are regulated by fixed timers. The timer is set to a fixed integer number. According on the timer setting, the lights automatically transition between red and green.

3) Electronic Sensors: Installing proximity sensors or loop detectors on the road is another sophisticated option. This sensor collects information regarding traffic in the area. The traffic lights are managed based on sensor data.

These traditional procedures have several disadvantages. The manual control mechanism requires a significant quantity of personnel [7]. We cannot have traffic police manually managing traffic in all regions of a major metropolitan area due to a lack of resources. As a result, a better traffic management system is required. Static traffic control employs a traffic signal with a countdown for each phase that is set in stone and does not respond to real-time traffic on the road. Since the extraction of high-quality data is frequently reliant on complex and costly technologies, and therefore limited budget will lower the number of facilities, precision and availability are often in conflict when utilizing electronic sensors, such as sensing devices or loop sensors [8]. Furthermore, since most sensors have a limited effective range, comprehensive coverage throughout a network of sites generally necessitates a large number of sensors.

III. REAL-TIME TRAFFIC MANAGEMENT SYSTEM

With ever-increasing traffic congestion around the world, traditional approaches to traffic control are ineffective for seamless commuting. As a result, a solution that is widely acceptable and leads to better traffic management is required [9]. The signal switches at its predetermined regular interval in today's traditional technique, but the number of vehicles on the road at each light does not remain the same, so the static approach fails. If the signal continues to change on a regular basis, the densely populated side of the street will always be entirely packed [10]. As previously stated, the above systems are exclusively used to get vehicle counts in order to conduct comparative traffic studies and analyses.

Many projects are growing to transform city's' conventional transportation systems into dynamic automated systems, a good example of this is the dynamic transport System [10][11]. Numerous efforts have been made to build a method that can do dynamic sign supervision, which means that the sign-changing conditions will not be predetermined, but will instead be determined by the amount of carriers on both sides of the road [13]. Various detection techniques can be employed to obtain a tally of the amount of carriers on the road. Vehicle detection utilizing sensors, for example, may fail in situations when traffic density is higher at peak time.

YOLO is an object detection interface. In terms of robustness, it is one of the best pre-trained models for providing the highest level of accuracy [8]. Yolo is a hybrid of SSD and RCNN, which makes the algorithm significantly faster, more efficient, and powerful. Using the YOLO object identification algorithm, one may not only discern what is in an image, but also where a particular object is positioned, or its location. Furthermore, because the model was trained on a large dataset, it can recognize images put in any arbitrary order. In other words, it can recognize objects that have been rotated 360 degrees [21].

YOLO is an efficient model that distinguishes between two items that are very close together. Rather than applying a classifier to each image and creating a forecast, one look at the image from Yolo is all it takes [22]. It uses the algorithm to estimate confidence scores in partitions one by one. The confidence score indicates whether or not an object is present. YOLO finds an object based on the confidence score [23].

Our objective is to develop a system that will enable us to estimate not only the number of passing cars in the lanes, but also their direction of travel. Furthermore, such a system should be able to categorize traffic flow into key vehicle groupings, allowing for more accurate traffic forecasting and better coordination tactics [14]. Based on the density of cars on a specific route, it becomes feasible to construct a system that effectively minimizes and anticipates traffic congestion by taking certain measures in advance. For example, traffic lights and signage must be carefully adjusted. The detection approach demands high precision and image processing speed in order to categorize an item and establish its position in an image [15]. The number of strategies available to address this issue is adequate, enabling individuals to choose the most suited solution for their requirements.

Surveillance systems and video tracking have been widely employed in traffic management in recent years for protection, ramp metering, and delivering real-time updated information to passengers [15]. Video surveillance systems may also be used to estimate traffic density and vehicle categorization, which can subsequently be utilized to manage traffic signal timings to improve traffic flow and reduce congestion [16]. Our suggested system attempts to provide a computer vision-based traffic signal controller that really can adjust to the present traffic scenario. It calculates real-time traffic density using live pictures from traffic junction CCTV cameras by recognizing the number of cars at the signal and adjusting the green and red time appropriately [17]. To get an accurate estimation of the traffic signal time, the vehicles are classed as a bus/truck, car, motorbike, or tricycle. It employs YOLO to identify the number of cars and then sets the traffic signal timing based on the density of vehicles in the direction indicated. This helps to optimize green signal periods, and congestion is handled at a much quicker pace than with a static system, resulting in less bottlenecks, congestion, and waiting times, as well as lower fuel usage and pollution.

IV. RELATED WORK

A. *Vehicle Detection, Tracking and Counting*

Computer vision researchers have been investigating the detection of vehicles in sequential blocks as a way to resolve an ever-increasing problem of traffic monitoring and safety. It involves both theoretical and algorithmic approaches to achieve autonomous visual understanding. Computer vision entails the automatic removal, perception, and interpretation of meaningful data from a particular picture or series of images. In this paper, a Gaussian mix-up design is used to reduce the background in an automated carrier identification, counting, and tracking system [40]. After the foreground mask is treated to remove noise, morphological operators (e.g., opening, expansion, decay, and conclusion) are utilized followed by the BLOB (Binary Large Object) analysis to distinguish the automobiles [20]. The system detects clusters of connected pixels that are indicative of moving objects. Then, a binary classifier helps differentiate between vehicles and pedestrians. The Kalman filter then aids in predicting the location of vehicles during the following update period due to the fact that the height to width ratio of cars is always greater than it is for foot-travelers. As the next step, the Hungarian algorithm is used to associate labels with the tracked cars. Each of these processing steps is performed in a defined ROI so only cars entering the ROI are recognized. As a result, the carrier count is incremented only when a vehicle joins the ROI [13]. In the initial step, blocks of video data are used as inputs to the Gaussian Mixture Model. By employing morphological operators, noise is filtered out. As a result of the noise removal, the cars that were identified are outlined using BLOB analysis, and a vehicle count is completed based on the highlighted cars. With this framework, cost estimation is done using the Hungarian approach. This method is utilized for assigning routes to associated organizations.

B. *Gaussian Mixture Model and Kalman Filter Vehicle Detection and Tracking*

In recent years, technical advancement has accelerated in a variety of disciplines, particularly in the world of transportation, including the Intelligent Transportation System (ITS). ITS was a traffic-control mechanism utilized in developed countries to create an efficient road-based transportation system. The usage of CCTV cameras for surveillance is an example of an ITS application [7]. The transportation authorities and decision-makers may readily access data on the number of automobiles and vehicle speed to be used in traffic engineering. The detection of vehicle objects is the first stage in acquiring statistics on the number of autos and their speeds through CCTV. The HOG, Viola Jones, and GMM are all methods for detecting vehicles. When using CCTV to identify things, it is necessary to distinguish between the item to be identified and other items. The detection algorithms HOG and Viola Jones depend on pre-existing datasets [14]. There are two categories of data in this database: positive and negative. The affirmative database includes data containing the item to be identified, while the deleterious database contains data that will not. This scenario for data differentiation is widely used to locate objects in photographs. The GMM technique (stationary object) is a detection method of comparing forefront (moving) and backdrop (stationary) items. This approach is often used to identify items in movies [18]. The second step, in addition to vehicle detection, is to use CCTV to monitor the item in traffic surveillance. The Kalman Filter method is one of the ways that can be employed in object tracking. Vehicle detection is carried out in this study utilizing the GMM approach in conjunction with object tracking using the Kalman Filter. The combination of these strategies is likely to improve system detection accuracy.

C. Computer Vision-Based Intelligent Traffic Light System with Android Control and Monitoring

For decades, traffic congestion has been one of the primary unresolved challenges affecting the Philippines' citizens and industry sectors, and it is becoming worse every year. As the number of automobiles grows, this is becoming a global phenomenon. This traffic problem has a negative impact on a country's productivity and economy. In actuality, the United States spent around $300 billion on gas and time, with over $70 million wasted each year due to heavy traffic in the corporate sector. Furthermore, the Boston Consulting Group estimates that traffic in Manila, Philippines, would approach "standstill levels" by 2022, implying that cars will drive at a speed of 5 mph or less at rush hours. According to a research undertaken by the JICA, growing traffic congestion costs Metro Manila, Philippines, 3.5 billion work hours and economic opportunities every day [9]. Several initiatives have been made throughout the country to address the problem of traffic congestion, including expressways and path expansion and the deployment of various congestion plans. One of the study topics being looked at is a solution to the constraints of standard stoplights. The primary purpose of introducing intelligent transportation systems (ITS) as a potential replacement for traditional traffic light systems is to increase traffic effectiveness [15]. The street signage are utilized to improve traffic efficiency by changing signal lights for managing traffic flow at road junctions, pedestrian crosswalks, and other sites. Conventional traffic light systems operate on fixed-timed cycles, which means the signaling lights alternate at periodic intervals. It is useless since traffic is an irregular occurrence that changes frequently owing to various situations, such as peak hours, accidents or collisions, rallies, roadshows, and so on. Because of these restrictions, ITS was developed. The goal of ITS is to govern traffic flow activities in an adaptable way based on real-time traffic circumstances. An intelligent traffic light system is one that can detect and compute the traffic density of each route using any sensors or cameras positioned on crossings, and then prioritize lanes that are more crowded and demand attention [5]. However, because these types of systems require a significant amount of human engagement to become completely functioning, automated computer vision-based approaches have been developed. The researchers installed a rotational camera in the intersection's center to take views from multiple lanes, after that, edge detection image processing methods were used to process the pictures. Traffic density and duration were determined for the operation and management of signalized intersections lights. A camera installed on a DC motor at a traffic crossing was used to take photographs for another investigation. Image processing methods employed included cropping, boosting, thresholding, edge detection, and object counting [19]. However, implementing computer vision-based traffic signal systems remains problematic due to the fact that most of these systems were not designed to withstand severe weather conditions (stormy, foggy, and rainy), which might influence vehicle detection accuracy. Furthermore, some trials that employed a camera for image processing had difficulty determining traffic volume at nighttime, and controls were generally done through a computer or microcontroller, which had restricted mobility and access. The purpose of this research is to develop a smart traffic signal system that can function at any time of day and employs computer vision to evaluate traffic density [15]. The system captures traffic pictures using CCTV cameras in each lane of the junction, which are then sent to the Raspberry Pi 3 microprocessor for image processing and traffic density calculation. An android application enables for automatic and manual traffic light control, which improves mobility and accessibility.

D. Image Vision for Automated Traffic Monitoring

Installing CCTV cameras with a 15-frame-per-second frame rate in important areas is the most common approach for maintaining an automated traffic monitoring system. Modern towns have taken use of this capability with the launch of IoT-based cameras [15]. Certain specialized algorithms must be performed to lower the method's complexity in order to improve it. This is critical in order to deliver results as rapidly as feasible. Traffic data is presently collected in real time from several systems using inductive loop sensors, spectrometers, radar guns, and video-based systems. The need for an intelligent traffic monitoring system is urgent. Manual traffic control is the method that is used in our country. It has a number of downsides, including the usage of personnel, regardless of the fact that it is a commonly used and dependable technique. Another problem is the inability to accurately determine the number of cars on a specific route and reduce congestion by preferring the roads with the highest traffic. The usage of pressure plates and the notion of RFID on vehicle number plates might be utilized to automatically calculate the number of vehicles [16][20]. Another possibility is to employ video monitoring of traffic. A popular traffic monitoring approach is for traffic officers to stay in a control room and semi-manually watch the numerous traffic cameras. Because it demands the use of human resources, it is also not a very efficient technique. As a consequence, we propose in this paper a technique for detecting the number of vehicles in each route at the traffic cross-section utilizing Digital Image Processing (DIP) concepts, and therefore providing inputs to any system for traffic management scheduling. Our technique relies on a basic edge detection accompanied by enclosed figure recognition to count the number of cars with minimum effort, rather than high-resolution video. This research proposes a two-step method for calculating the number of cars on each roadway at the merge, followed by a Traffic Scheduling Algorithm to relieve traffic congestion at highly populated road crossings. The result of the Vehicle Identification Algorithm is sent into the Traffic Scheduling Algorithm. The second comprises picking the best suitable scenario out of 12 potential traffic circumstances in the case of traffic crossroads of four roads with three assigned lanes for Left, Straight, and Right [16].

E. Object detection and recognizition by YOLO

In the realm of computer vision, real-time object identification is a popular issue, with various research indicated in the literature. For face detection, the Haar features hased cascaded Adaboost classifier was initially developed [17]. The use of a Deformable Parts Model (DPM) to identify an item using HoG [18] and part-based approaches has become popular. Due to the availability of Graphical Processing Units (GPUs) and a large number of datasets, deep learning-based algorithms have recently become popular. These strategies make advantage of CNN features with a time-consuming sliding window or selective search mechanism [21][22]. However, a reliable procedure is required. YOLO, which translates pixels into bounding boxes with class probabilities, treats object recognition as a regression problem [18]. Additionally, it does all calculations in a single evaluation, leading in a refresh rate of 45 fps.

We accommodate in the suggested work to serve the detection purpose, motivated by the generalisation property, performance accuracy, and speed of YOLO [21]. YOLO handles the detection method as a regression issue in order to turn the picture into object bounding boxes. Each grid predicts B (example: 5) enclosing boxes with reliability ratings from the input picture, which is split into M x M (example: 13 x 13) grids. Each enclosing box contains an x, y, w, h, and a corresponding output as predictions. Furthermore, each grid cell predicts C (example 20) conditional class probabilities, which indicate the likelihood of an item in that cell. The PASCAL VOC dataset was used to train YOLO, which can predict 20 distinct classes, including bicycles, boats, vehicles, buses, humans, and motorbikes [8]. The confidence score of each bounding box and class predictions are used to estimate the item. As a consequence, a single convolutional neural network is employed to anticipate the bounding box in a single assessment.

V. PROPOSED SYSTEM

The traffic flow does not follow any particular pattern, and indeed the static light timings exacerbate the already serious issue of congestion. Implementing a system that aims to reduce the likelihood of these scenarios by instantaneously processing the ideal green and red time based on actual number of vehicles at the signal will guarantee that the path with much more traffic is given a green signal for a longer period of time than the path with far less traffic. This technology may override the earlier method of hard-coded signals, which causes unnecessary delays, lowering traffic and waiting times, reducing the frequency of fatalities and fuel usage, and thereby reducing air pollution.

Our suggested system uses an image from traffic junction CCTV cameras as input to calculate real-time traffic density utilizing computer vision and object identification. As may be seen in Figure. 1. This picture is sent to the YOLO-based vehicle identification algorithm.

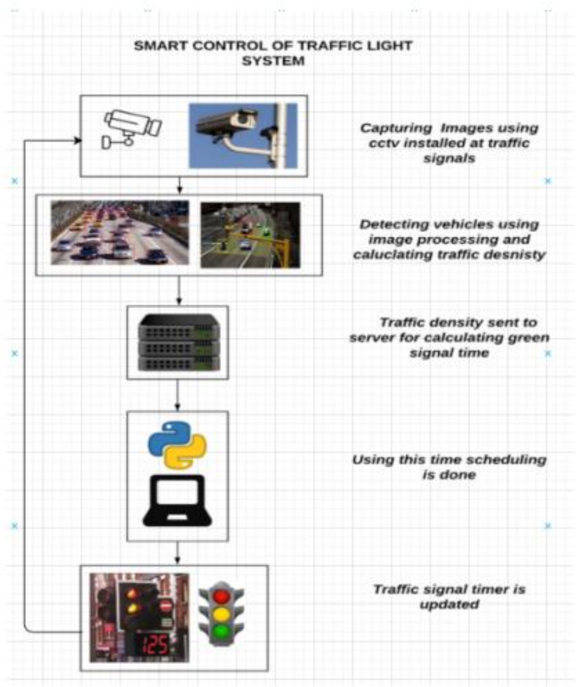

Figure 1: Proposed System Model

The vehicle of each type, such as motorbikes, buses,cars and rikshaws, is tracked in order to compute traffic density. This density, along with a few other criteria, is used by the light switching algorithm to establish the green signal timing for each lane. The red signal timings have been modified to reflect this. In order to prevent lane hunger, the green signal time is limited by the maximum and minimum value. To show the system's efficacy and compare it to the present static system, a simulator is also created.

VI. DATASET

To design a model, we must first generate an image dataset. A dataset is a collection of annotated photos that are used as reference for things that developers use to test, train, and assess the performance of computer techniques. Algorithms trained on bigger datasets outperform those taught on tiny ones by a wide margin.

While various datasets for developing machine learning models are currently available, they tend to concentrate on nicely organized driving settings. This is frequently

TABLE I: Comparison of various object detection datasets.

Dataset	Classes	Train			Validation			Test
		Images	Objects	Objects/Image	Images	Objects	Objects/Image	
PASCAL VOC 12	20	5,717	13,609	2.38	5,823	13,841	2.37	10,991
MS-COCO	80	118,287	860,001	7.27	5,000	36,781	7.35	40,670
ILSVRC	200	456,567	478,807	1.05	20,121	55,501	2.76	40,152
OpenImage	600	1,743,042	14,610,229	8.38	41,620	204,621	4.92	125,436

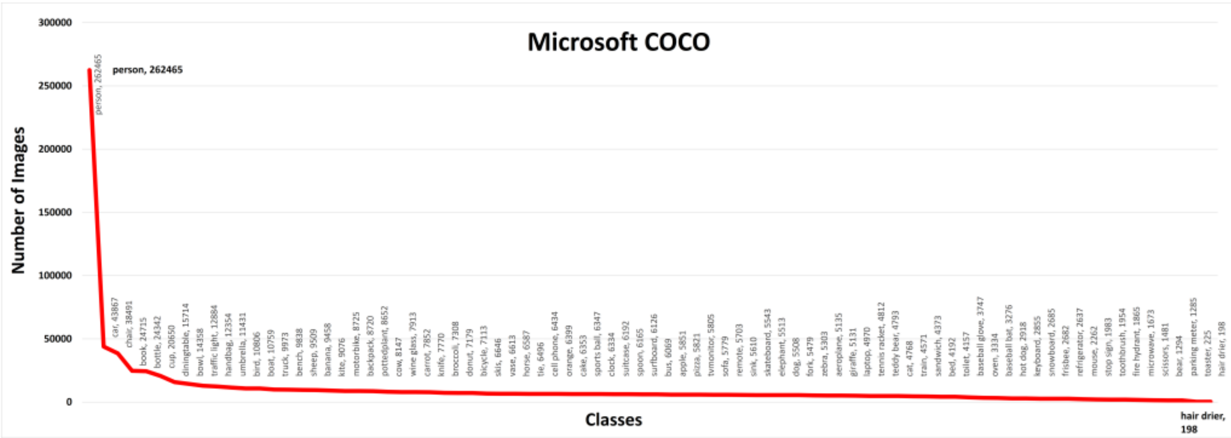

Figure 2: Number of images for different classes annotated in the MS-COCO dataset

associated with well-defined architecture, such as lanes, a limited number of well-defined traffic participant types, little variety in object or backdrop appearance, and rigorous adherence to traffic laws. For our research, we utilized COCO as the primary source of training pictures.

One of the most challenging datasets, MS-COCO, has 91 common objects in their natural context that can easily be recognized by a 4-year-old [40]. As well as containing images from a variety of perspectives, MS COCO introduces a stringent test to measure the performance of a detector. It was launched in 2015 and has only grown in popularity since then. Additionally, the MS-COCO dataset utilizes AP separately for small, medium, and large objects in order to compare performance at various scales [39] .

Scraping photos from Google and manually labeling them utilizing LabelIMG, a graphic annotation tool, was also performed. The model was then trained using weights obtained from of the YOLO portal that had already been trained.

VII. SYSTEM IMPLEMENTATION

A. Vehicle Detection Module

For vehicle detection, the presented system employs YOLO, which gives the requisite accuracy and processing speed. Vehicle detection was trained using a unique YOLO model that can recognize vehicles of various classes, including automobiles, motorcycles, large vehicles and rickshaws.

YOLO is a smart CNN that can conduct real-time object identification. The technique splits the picture into areas then estimates boundary boxes with likelihood for each region using a singular neural network applied to the whole image [8]. The projected probabilities are used to weight these bounding boxes. YOLO is significant because it has a high level of accuracy and can run in real-time. To create predictions, the method "only looks once" at the picture in the sense that it only takes single forward transmission passage through the neural network. It then returns recognized objects along with coordinates after nonmax suppression (this ensures that the object recognition system only identifies each item once). A single CNN estimates numerous coordinates and class label for those regions using YOLO [29].

The vehicle detection framework was employed to test photos in Figure 3. The original picture is on the left of the image, and the output is now on the right side, with coordinates and labels, after the vehicle recognition model has been applied to the image.

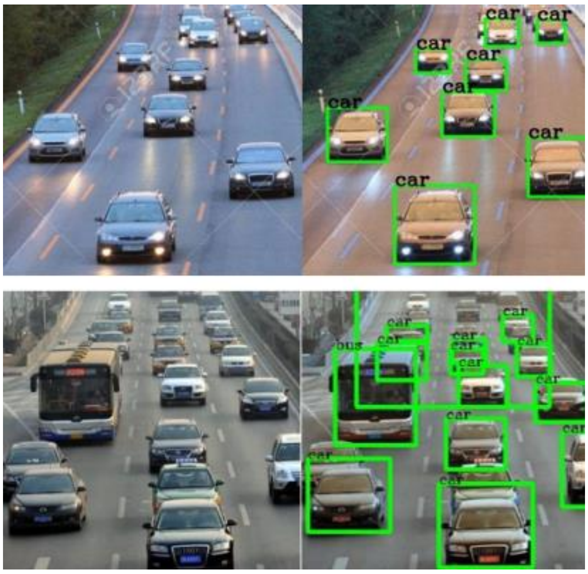

Figure 3: Vehicle Detection Results

B. Parameter tuning for customized YOLO

To reduce processing time, the backbone CNN employed in YOLO may be streamlined further. Darknet is a C and CUDA-based open - sourced neural net framework [31]. It's quick to set up and supports both CPU and GPU computing. On ImageNet, YOLO obtains a top 1 accuracy of 72.9 percent and a top 5 accuracy of 91.2 percent on DarkNet. Darknet mostly employs 3 3 filters for feature extraction and

1 1 filters for output channel reduction. It also makes predictions using global mean pooling [30].

Scraping photos from Google and manually labeling them utilizing LabelIMG, a graphic annotation tool, provided the data for training the model [32]. The model was then trained using weights obtained from the YOLO site that had already been trained. The settings of the.cfg document used for learning was altered to match our model's criteria. By altering the 'classes' variable, the total current neurons in the final layer were set equivalent to the number of categories the model is expected to identify. This was 4 in our system, i.e. Bus/truck, car, bike, and rickshaw are the four modes of transportation. The number of filters must also be adjusted using the equation 5*(5+number of classes), which in our instance is 45. The network is trained after these configuration adjustments until the loss was greatly reduced and then no longer seemed to have been decreasing. The training came to an end at this point, and the weight was adjusted to our specifications. With the aid of the OpenCV library, those weight then were loaded into code and utilized for vehicle detection. The minimal level of confidence necessary for successful detection is established as a threshold [32]. After the model has been loaded and a picture has been fed into it, the output is returned in JSON format, which is a set of key-value pairs with labels as keys and confidence and coordinates as values. From the coordinates and label acquired, OpenCV was used to construct boxes upon that pictures.

C. Signal Switching Module

The Light Switching Algorithm adjusts the red traffic timers of other lights based on the traffic density given by the vehicle detecting module, and changes the green traffic timer appropriately. It also cyclically alternates between the signals based on the timings.

The intelligence about the cars observed by the detector, as stated in the preceding section, is sent into the algorithm. It’s in JSON format, with the key being the identified entity's label, and the value being the confidence and coordinates. The aggregate number of cars in each class is then calculated using this data. Following that, the signal's green signal time is computed and assigned, as well as the red traffic timings of other lights are altered as necessary. The number of indicators at an intersection may be increased or reduced using this approach.

During the development of the algorithm, the following aspects were taken into account:

1) The algorithm's processing time for calculating traffic density, followed by the green signal duration - this determines when the picture must be collected.

2) Amount of lanes

3) The total number of vehicles in each class, such as automobiles, trucks, motorbikes, and so on.

4) The above-mentioned parameters were used to compute traffic density.

5) Time increased owing to the latency each car experiences upon startup, as well as the non-linear rise in lag experienced by vehicles in the rear [33].

6) An average speed of every vehicle class when the light becomes green, average time necessary for each vehicle type to crossing the signal [34].

7) A min and max time frame for the length of the green light - to avoid starving the default time for first signaling of the first loop is established when the method is first run, and the algorithm sets the timings for other indicators of the very first loop as well as all signals of future cycles.

The detection of cars for each direction is handled by a different thread, while the current signal's timing is handled by the main thread. When the current signal's green light timer (or the following green signal's red light timer) hits five seconds, the detecting threads take a picture of next direction. The result is subsequently processed, then next green signal's timer is set. All this occurs in the background, whereas the main thread counts down the new green signal's timeout. This enables for a smooth timer assignment and hence eliminates any latency. The following signal turns green for the period of time defined by the algorithm whenever the current signal's green timer reaches zero. When the signal that will change color iis 5 seconds away from being caught, the photograph is taken. This provides the system 10 seconds to analyse the picture, determine the number of cars within every class present in an image, compute the green traffic time, and set the timing of this signaling and the red traffic time of the following signal appropriately. The average speeds of cars at starting and their acceleration durations were utilized to calculate the best green signal time determined by the number of automobiles of each category at a signal, whereby an estimation of the estimated duration every class of vehicles takes to traverse an intersection was obtained [34]. Equation is then used to compute the green signal timing (1).

$$GST = \frac{\sum_{vehicleClass} (NoOfVehicles_{vehicleClass} * AverageTime_{vehicleClass})}{(NoOfLanes + 1)} \quad (1)$$

Where:
• GST is green signal time
• noOfVehiclesOfClass is the volume of traffic of each class identified by the vehicle detection module at the signal,
• averageTimeOfClass is the average time it takes for automobile of that class to cross a junction, and
• noOfLanes is the amount of lanes at the junction.

The signals are switched in a cyclic pattern rather than in the order of densest to least dense. This is in line with the present system, which requires individuals to change their routes or create any confusion since the light is green one after another in a predetermined sequence. The arrangement of the indicators is however the same as it is now, and the yellow indicators were taken into consideration.

D. Simulation Module

To imitate real-life traffic, a simulation was created from the ground up using Pygame. It helps in conceptualizing the process and compares it to a static system that already exists. It has a four-way junction with four traffic lights. On atop of every signal is a timer that displays the amount of time before the signal changes. The amount of vehicles that already have crossed the junction is also shown next to each light. Automobiles, bicycles, vans, trucks, and rickshaws arrive from all directions. Some of the cars in the right - hand column move to cross the junction to keep the scenario more realistic. When a vehicle is produced, random numbers are used to determine whether it will turn or not. It also has a timer that shows how much time has passed since the simulation began. Figure 4 depicts a glimpse of the simulation's ultimate outcome.

Figure 4: Simulation output

Pygame is a suite of Python modules for creating video games that may be used on any platform. It offers Python-compatible graphics and sound libraries. On atop of the fantastic SDL library, Pygame adds capabilities. Users may use Python to construct full-featured games and multimedia products. Pygame is very portable, and it can be used on almost any device or system software. Its open source and LGPL licensed [35][36][40].

VIII. PERFORMANCE COMPARISON

One of most reliable real-time detector is YOLO [4][8]. The accuracy and speed of quick detectors are compared. Fast YOLO is the world's fastest COCO detector, but it is still two times as precise as every other real-time detector. YOLO is 10 mean Average Precision more precise as compared to the the fast variant while maintaining a pace that is substantially above real-time.

Real-Time Detectors	Train	mAP	FPS
100Hz DPM	2007	16.0	100
30Hz DPM	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM	2007	30.4	15
R-CNN Minus R	2007	53.5	6
Fast R-CNN	2007+2012	70.0	0.5
Faster R-CNN VGG-16	2007+2012	73.2	7
Faster R-CNN ZF	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

Figure 5: Real-time systems on COCO

R-CNN is still far from real-time, and the lack of strong suggestions has a substantial impact on accuracy [41].Fast R-CNN accelerates the categorization step of R-CNN, but it still depends on selective search to create bounding box suggestions, which may take up to 2 seconds each picture. As a result, it has a high mAP, but at 0.5 frames per second, it is very far from realtime.

To propose bounding boxes, the Faster R-CNN substitutes selective search with such a neural network. Their most precise model produces 7 frames per second in our testing [42], while a simpler, less precise obtains 18 frames per second. The VGG-16 version of Faster R-CNN is 10 mAP faster than YOLO, but it is also 6 times slower. Although the Faster R-CNN is just 2.5 times slower than YOLO, it is less accurate.

IX. RESULTS AND ANALYSIS

A. Performance evaluation of Vehicle Detection Module

The traffic surveillance module was put to the test with a wide range of experimental photos comprising varied quantities of cars, with detection accuracy ranging from 75 to 80%. In figure 3, several test results are given. This is OK, but not ideal. The absence of a good dataset is the fundamental cause of poor accuracy. To enhance this, real-world film from traffic cameras may be utilized to train the model, resulting in increased system accuracy.

B. Performance evaluation of the proposed adaptive system

Multiple simulations of both systems were ran for 5 minutes each, with various traffic patterns across the four directions, to see how the proposed adaptive system compared to the present static system. The number of cars that were able to pass through the junction in a certain amount of time was used to assess performance. In other words, the signal's idle time is compared, which is the period when the signal is green but no automobile passes through the junction. This has an influence on vehicle wait times as well as the duration of other signal queues. The chance of a vehicle being in lane 1, lane 2, lane 3, or lane 4 is a/d, (ba)/d, (c-b)/d, and (d-c)/d, respectively, according to the distribution [a,b,c,d]. In simulation 1, for example, the distribution is [300,600,800,1000], implying probabilities of 0.3, 0.3, 0.2, and 0.2. The findings were reported in the form of the overall number of cars passed and the vehicles passed per lane.

No.	Distribution	Lane1	Lane 2	Lane 3	Lane 4	Total
1	[300,600,800,1000]	70	52	52	65	239
2	[500,700,900,1000]	112	49	48	31	240
3	[250,500,750,1000]	73	53	63	62	251
4	[300,500,800,1000]	74	44	65	71	254
5	[700,800,900,1000]	90	32	25	41	188
6	[500,900,950,1000]	95	71	15	14	195
7	[300,600,900,1000]	73	63	69	24	229
8	[200,700,750,1000]	54	89	10	67	220
9	[940,960,980,1000]	100	10	8	4	122
10	[400,500,900,1000]	81	29	88	37	235
11	[200,400,600,1000]	42	47	54	86	229
12	[250,500,950,1000]	39	52	93	22	206
13	[850,900,950,1000]	74	10	13	17	114
14	[350,500,850,1000]	49	46	69	50	214
15	[350,700,850,1000]	51	64	37	43	195

Table II: Simulation Results of Current Static System

Figure 6 shows that, regardless of the distribution, the suggested adaptive system always outperforms the existing static system. The amount of increase in performance is determined on how lopsided the traffic distribution is throughout the lanes. The higher the skewness of the traffic distribution, the better the performance.

• When traffic is distributed evenly or almost evenly across the four lanes, the proposed approach performs just marginally better than the present system. In simulations 1, 2, 3, and 4, this is the case. In this case, the performance gain is roughly 9%.

• When the traffic dispersion is substantially skewed, the suggested approach outperforms the present system significantly. In simulations 5, 6, 7, 8, 14, and 15, this is the case. Here, the performance boost is roughly 22%. This is often the type of traffic distribution found in real-world circumstances.

No.	Distribution	Lane1	Lane 2	Lane 3	Lane 4	Total
1	[300,600,800,1000]	87	109	41	50	287
2	[500,700,900,1000]	128	55	49	25	257
3	[250,500,750,1000]	94	50	60	58	262
4	[300,500,800,1000]	89	46	69	59	263
5	[700,800,900,1000]	185	25	23	28	261
6	[500,900,950,1000]	94	118	11	16	239
7	[300,600,900,1000]	87	68	70	33	258
8	[200,700,750,1000]	56	108	19	78	261
9	[940,960,980,1000]	193	6	5	7	211
10	[400,500,900,1000]	97	29	100	34	260
11	[200,400,600,1000]	26	52	67	99	244
12	[250,500,950,1000]	52	75	101	7	235
13	[850,900,950,1000]	154	17	12	18	201
14	[350,500,850,1000]	64	53	80	47	244
15	[350,700,850,1000]	66	82	40	48	236

Table III: Simulation Results of Proposed Adaptive System

• When traffic distribution is skewed, the suggested approach outperforms the present system significantly. This is the situation in simulations 9 and 13, when the red line lowers abruptly and the red and green lines are separated by a wide gap. Here, the performance gain is roughly 36%.

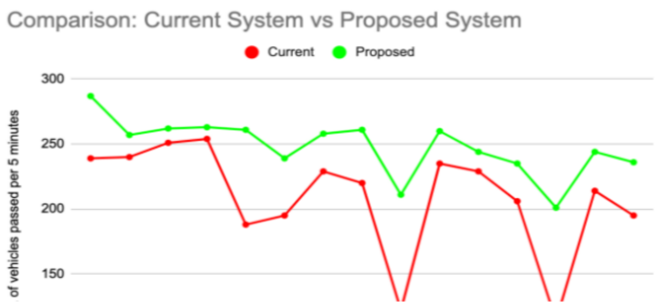

Figure 6: Comparison of current static system and proposed adaptive system

The simulations were conducted for a total of 1 hour 15 minutes, with 300 seconds i.e. 5 minutes for each distribution, with all simulation conditions being the same, and it was discovered that the proposed system fortified by about 23% on average when compared to the current system with fixed times. This means less time spent waiting for cars and less time spent idling at green signals. When these findings were compared to those of several other adaptive systems, it was discovered that the suggested system outperforms them. For example, [22] has a 70 percent accuracy, but the suggested method has an accuracy of 80 percent. When compared to static systems, reference obtains a performance gain of 12%, while the suggested system achieves a performance improvement of 23% [23].

X. FUTURE PROSPECTS

Object detection has advanced tremendously in the recent decade. Despite this tremendous progress, numerous concerns remain for future research. Despite the fact that some algorithms have achieved human-level accuracy in specific categories, there are still numerous difficulties to overcome [1].

When the model fails to recognize critical moments such as bad weather and low visibility, the timer-approach might be enabled [5]. In the future, cloud computation functionality can be added to the system so that it can register traffic in appropriate lanes with date and time, which will be extremely useful in evaluating traffic data for road improvement. With continued use of our model, this scenario can be greatly reduced, as machine learning models can learn to adapt to a variety of conditions. Our model can add even more custom-functions to the software, such as a closure signal for people crossing, lane priority for ambulances, and vehicle monitoring, among others [11].

To improve traffic management and reduce congestion, the project may be developed to incorporate the following functionalities:

1) Vehicles running red lights may be spotted in an image or video stream by drawing a violation line and collecting the picture's number plate if the line is crossed while the signal is red. Changing lanes may also be recognized in the same way. Background removal or image processing methods may be used to accomplish this.

2) Detecting pedestrians and adjusting the signal time as necessary.

3) Accident or failure detection: Because various forms of injurious accidents, such as angle and left-turn collisions, often occur near intersections, they are prone to catastrophic wrecks. As a result, precise and timely identification of accidents at junctions has enormous advantages in terms of saving property and lives while also reducing congestion and delay. This may be accomplished by recognizing automobiles that stay motionless for an extended period of time in an inconvenient location, such as in the center of the road, while excluding parked vehicles [36].

4) Traffic signal synchronization at several intersections: Synchronizing lights along a roadway may help commuters since once a vehicle joins the street, it can proceed without halting [38].

5) Adapting to emergency vehicles: Emergency vehicles, such as ambulances, must be allowed to pass through traffic signals more quickly. The model may be taught to identify not just automobiles, but also emergency vehicles, and to adjust the timings appropriately so that the emergency vehicle is given precedence and is allowed to pass the signal as soon as possible.

XI. CONCLUSION

YOLO, a unified object detection model, is simple to create and can be taught on full images. Unlike classifier-based approaches, YOLO is developed on an algorithm that directly correlates to detection performance, and the whole model is learnt concurrently. YOLO is the quickest object classification detector in the world, pushing the limits of real-time object recognition. YOLO may also be adapted to new areas, making it ideal for applications requiring fast and accurate object detection. The identification and counting of automobiles in a mixed traffic situation is proposed in this research utilizing the YOLO framework. With the aid of this project, which will concentrate on minimizing vehicle jams, the main purpose of the Dynamic Traffic Management System is really to tackle the traffic issue that most cities in urban and rural regions are facing.

Despite the fact that object detection has progressed significantly over the last decade, the finest detectors are still far from reaching peak performance. The setup requires traffic data as input, which will be combined with our machine learning model to ensure effective traffic flow while minimizing road congestion. Although the model will take longer to train, the reaction time will be shorter.

Finally, the suggested system adjusts the green signal time based on the traffic density at the light, ensuring that the direction with the most traffic is given a green signal for a longer period of time than the direction with the least amount of traffic. This will eliminate unwelcome delays as well as congestion and waiting times, lowering fuel usage and emissions. According to simulation data, the system improves the number of cars crossing the junction by around 23% over the present method, which is a substantial improvement. This system may be upgraded to perform even better with additional calibration utilizing real-life CCTV data for training the model. Furthermore, the suggested system has several benefits over other intelligent traffic management systems that are already in use, such as Pressure Mats and Infrared Sensors. The cost of deploying the system is minimal since it uses video from traffic signal CCTV cameras, which needs no extra hardware in most situations because crossroads with significant traffic already have such cameras. It's possible that just little alignment is required. In comparison to other traffic surveillance systems like as pressure mats, which generally suffer stress and strain owing to their location on roadways where they are continually exposed to great pressure, the maintenance cost is also reduced. As a result, the suggested system may be connected with CCTV cameras in large cities to assist improved traffic control.

REFERENCES

[1] G. Dimitrakopoulos and P. Demestichas, “Intelligent transportation systems,” IEEE Vehicular Technology Magazine, vol. 5, no. 1, pp. 77–84, 2010.

[2] M. Cremer and M. Papageorgiou, “Parameter identification for a traffic flow model,” Automatica, vol. 17, no. 6, pp. 837–843, 1981.

[3] Pornpanomchai, C., Liamsanguan, T., & Vannakosit V.”Vehicle detection and counting from a video frame . In Wavelet Analysis and Pattern Recognition” , ICWAPR'08 . International Conference on, vol. 1, pp . 356 - 361, 2008

[4] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection” arXiv preprint arXiv, 2016.

[5] Xia, Y., Shi, X., Song, G. , Geng. Q., & Liu, Y Towards improving quality of video-based vehicle counting method for traffic flow estimation . Signal Processing, vol120, . 672-681, 2016.

[6] Barcellos, P., Bouvié, C. Escouto, E. L., & Scharcanski, novel video based system for detecting and counting vehicles at user-defined virtual loops. Expert Systems with Applications, vol 42 no. 4 , pp. 1845 1856, 2015 .

[7] BouvieC. Scharcanski, Barcellos, , & Escouto, L Tracking and counting vehicles in traffic video sequences using particle filtering In Instrumentation and Measurement Technology Conference (). 2013 IEEE Internationalpp 812-815..2013 .

[8] C. S. Asha and A. V. Narasimhadhan, "Vehicle Counting for Traffic Management System using YOLO and Correlation Filter," 2018 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, 2018, pp. 1-6, doi: 10.1109/CONECCT.2018.8482380

[9] K. H. Molloy, J. P. Ward, and V. M. Benson, “Traffic signal control system,” Jul. 4 1972, uS Patent 3,675,196

[10] G. Dimitrakopoulos and P. Demestichas, “Intelligent transportation systems,” IEEE Vehicular Technology Magazine, vol. 5, no. 1, pp. 77–84, 2010.

[11] K. Morinaga, “Car navigation system,” May 16 1995, uS Patent 5,416,478.

[12] Indrabayu, R. Y. Bakti, I. S. Areni and A. A. Prayogi, "Vehicle detection and tracking using Gaussian Mixture Model and Kalman Filter," 2016 International Conference on Computational Intelligence and Cybernetics, Makassar,2016,pp.115-119,doi: 10.1109/CyberneticsCom.2016.7892577.

[13] M. Cremer and M. Papageorgiou, “Parameter identification for a traffic flow model,” Automatica, vol. 17, no. 6, pp. 837–843, 1981.

[14] J. Tiwari, A. Deshmukh, G. Godepure, U. Kolekar and K. Upadhyaya, "Real Time Traffic Management Using Machine Learning," 2020 International Conference on Emerging Trends in Information Technology and Engineering (icETITE), Vellore, India, 2020, pp. 1-5, doi: 10.1109/ic-ETITE47903.2020.462.

[15] R. Krishnamoorthy and S. Manickam, "Automated Traffic Monitoring Using Image Vision", 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, 2018, pp. 741-745, doi: 10.1109/ICICCT.2018.8473086.

[16] S. Maqbool, M. Khan, J. Tahir, A. Jalil, A. Ali and J. Ahmad, "Vehicle Detection, Tracking and Counting", 2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP), Shenzhen, 2018, pp. 126-132, doi: 10.1109/SIPROCESS.2018.8600460.

[17] Viola , P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Vol. 1, pp. I-. IEEE.

[18] Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on (Vol. . 886 893). IEEE.

[19] Felzenszwalb, P. F., Girshick, R. B., & McAllester, D., Cascade object detection with deformable part models. In Computer vision and pattem recognition (CVPR), 2010 IEEE conference on, pp. 2241-22482010.

[20] Ren, S.He, K.Girshick, , & Sun, (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

[21] Redmon. S.. GirshickR.& Farhadi, A. (2016). You only look once : Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp 779-788).

[22] R. A. Asmara, B. Syahputro, D. Supriyanto and A. N. Handayani, "Prediction of Traffic Density Using YOLO Object Detection and Implemented in Raspberry Pi 3b + and Intel NCS 2," 2020 4th International Conference on Vocational Education and Training (ICOVET), Malang, Indonesia, 2020, pp. 391-395, doi: 10.1109/ICOVET50258.2020.9230145.

[23] 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.

[24] 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.

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

[26] 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.

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

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

[29] 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

[30] 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

[31] J. Redmon, ‘Darknet: Open Source Neural Networks in C’, 2016. [Online]. Available: https://pjreddie.com/darknet/

[32] Tzutalin, ‘LabelImg Annotation Tool’, 2015. [Online]. Available: https://github.com/tzutalin/labelImg

[33] Li, Z., Wang, B., and Zhang, J. “Comparative analysis of drivers' start- up time of the first two vehicles at signalized intersections”, 2016 J. Adv. Transp., 50: 228– 239. doi: 10.1002/atr.1318

[34] Arkatkar, Shriniwas & Mitra, Sudeshna & Mathew, Tom. “India” in Global Practices on Road Traffic Signal Control, ch.12, pp.217-242

[35] ‘Pygame Library’, 2019. [Online]. Available: https://www.pygame.org/wiki/about

[36] ‘Traffic Signal Synchronization’. [Online]. Available: https://www.cityofirvine.org/signal-operations-maintenance/trafficsignal-synchronization

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

[38] 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.

[39] A. Aslam and E. Curry, “A survey on object detection for the internet of multimedia things (iomt) using deep learning and event-based middleware: Approaches, challenges, and future directions,” Image and Vision Computing, vol. 106, p. 104095, 2021.

[40] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, “Microsoft coco: Common objects in ´ context,” in Computer Vision – ECCV 2014, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[41] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger.” [Online]. Available: https://arxiv.org/abs/1612.08242v1

[42] J. Redmon, “Darknet: Open source neural networks in c,” http:// pjreddie.com/darknet/, 2013–2016.