Check for updates

Effective Management of Delays at Road Intersections Using Smart Traffic Light System

Olasupo Ajayi[1,2](✉), Antoine Bagula[1], Omowunmi Isafiade[1], and Ayodele Noutouglo[2]

[1] Department of Computer Science, University of the Western Cape, Cape Town, South Africa
3944991@myuwc.ac.za, {abagula,oisafiade}@uwc.ac.za
[2] Department of Computer Sciences, University of Lagos, Lagos, Nigeria
olaajayi@unilag.edu.ng
http://www.uwc.ac.za

Abstract. Rapid industrialization coupled with increased human population in urban regions has led to a rise in vehicle usage. The demand for space (road) by motorists for transportation has risen. Unfortunately, infrastructural development has not been at par with vehicular growth thus resulting in congestion along major roads. Traffic lights have been used for years to manage traffic flow. While they serve a good purpose, their underlining principle of operation is to a significant degree inefficient as traffic congestion still prevails and remains a major concern till date. This study seeks to tackle this challenge by proffering a Smart Traffic Management System (STMS) based on image detection. The system incorporates cameras which dynamically capture road situation as images, run them through an image processing algorithm to obtain traffic density then automatically adjust the service times at intersections. To measure the effectiveness of the approach, mathematical models were formulated, analytical comparison as well as experimental simulations were done. Results show that SMTS out-performed the Round-Robin algorithm used by traditional traffic lights, by reducing service interruptions, cutting delay times by at least 50%, while remaining equally fair to all roads at the intersection. This system and its constituent components fall under the Edge computing paradigm as real time data capture, analysis and decisions are made by an embedded computer.

Keywords: Edge computing · Image processing · Scheduling · Smart traffic · Traffic management

1 Introduction

Traffic congestion according to the Merriam Webster dictionary refers to a congestion of vehicles in a particular area thereby hindering the expected flow.

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2020
Published by Springer Nature Switzerland AG 2020. All Rights Reserved
R. Zitouni et al. (Eds.): AFRICOMM 2019, LNICST 311, pp. 84–103, 2020.
https://doi.org/10.1007/978-3-030-41593-8_7

The transportation system in urban cities can be highly unpredictable as free highways can in a matter of minutes be inexplicable gridlocked. This can negatively affects many processes that leverage on transportation for productivity. By extension, road congestion can hinder economic growth by making people arrive late at work and delaying supply of raw materials to factories. In extreme cases it can be a hazardous to health, as drivers and commuters alike experience heightened stress levels daily. As an example, the problem of intra-urban traffic congestion in Lagos, the one-time capital of Nigeria have been examined by Bashiru, et al. [2]. In their work, they showed that 57% of commuters and motorists spend between 30 to 60 min stationary on traffic congested roads. It is therefore a note-worthy subject of interest as the present traffic situation in Lagos and similar cities around the world poses a major challenge to their economies. Reports have shown that most traffic gridlock emanate from road intersections [14]. Numerous traffic management techniques have been deployed over the years, some of which include construction of flyovers and bypass roads, creation of ring roads as well as employing traffic wardens. In recent times, the use of human traffic wardens to manually control vehicular movement at road intersections have been upgraded to traffic lights with their popular "Red, Amber, Green" lights.

While these methods have proven effective to reasonable extents; with human wardens, there is always the risk of being knocked down by reckless drivers and general inefficiencies that comes with employing humans to do repeated activities. Traffic lights have thus been deployed to augment or replace human wardens at road intersections. Traffic lights, though well capable of catering for the human deficiencies, are not without shortcomings. Prominent among which is the lack of human intuition. They simply follow a pre-programmed Round-Robin scheduling technique [24], in which movement access is granted to vehicles on a given lane for a specified time interval before switching to the next lane. The use of traffic lights alone has also proven ineffective, as they are either completely disregarded by impatient drivers or not smart enough to dynamically adjust their timing in response to traffic density on the various lanes they are controlling. The latter is especially common, as many a time, traffic lights repeatedly "pass" lanes with little or no waiting vehicles to the detriment of lanes with long queues. This is one of the major reasons why many drivers get impatient and choose to ignore the lights, ultimately resulting in gridlocks at intersections. Significant improvements are therefore necessary in order to optimize the potentials of the traffic lights. This study proposes a system that combines intelligence (as in the case of the traffic wardens) and the mechanical advantage of a machine to improve the efficiency of the traffic light system. The specific contribution of this paper consists of the design, implementation and performance evaluation of an Intelligent Traffic Management System that:

- uses a lightweight hardware based on Raspberry pi to manage vehicular traffic
- utilizes image processing using the Single-Shot Multibox Detector (SSD) to detect and count the number of vehicles on the road.

– intelligently manages traffic by modelling a T-junction as an M/M/1 queue and utilizes an intelligent algorithm to schedule the departure of vehicles at the junction.

The proposed system falls under the broad umbrella of edge computing, as it involves infusing cameras and processing units into traffic light systems.

The rest of this paper is organized as follows: Related research on traffic management are reviewed in Sect. 2. Section 3, presents the cyber-physical traffic management framework (cyTAC). In Sect. 4, the smart traffic management system (a subset of cyTAC) and its corresponding models are shown. Section 5 details the phases of STMS. System implementation and performance evaluations are discussed in Sects. 6 and 7 respectively. The paper is concluded with future works highlighted in Sect. 8.

2 Related Research

The authors in [14] identified "+" and "Y" junctions as roots of traffic congestion. Their work attempted to pinpoint the exact causes by simulating road traffic situations in software. A hybrid of Structured Systems Analysis and Design Methodology (SSADM) with Fuzzy-Logic based design methodology was used to perform the traffic analysis. Finally they presented a system that could be used to tackle traffic congestion at junctions. The results of the study done in [21] show that poor driving habits, poor road network, inadequate road capacity, and lack of parking facilities constitute the greatest causes of traffic congestion in developing countries. Their work highlighted remedies for improving traffic conditions, which included: good road network, encouragement of mass transport system, proper traffic planning & management and regular education of road users.

Similarly, after a thorough analysis of causes and effects of traffic congestion along certain roads, the following suggestions were also made by [16]: doubling (dual lane) of roads, mounting traffic control devices at junctions, providing designated parking lot along the roads, and removal of shops or markets along the sides of the major roads. Their work investigated only a few roads in Oyo State, Nigeria but many of the observed issues do not apply to roads in other cities.

Bramberger et al. [4] stated that the integration of advanced CMOS image sensors with high-performance processors into embedded systems can be useful. In their work, they implemented a smart camera system for traffic surveillance. The image processing approach used was based on long-term intensity changes of background pixels in videos. Though they recorded positive results, their work assumed consistent ambient light conditions; also stored videos would quickly fill up storage while the transmission over a network for processing would both consume bandwidth and suffer from significant latency, hence limiting real time application.

The authors in [5] proposed an intelligent traffic control system using Radio Frequency Identification RFID. In their work they highlighted the major disadvantages of the timing circuit currently in use by traffic lights. It was noted that existing systems do not take the current volume of traffic on roads at intersections into consideration. Traffic congestion, according to them translates to lost time, missed opportunities and in general wastage. In an attempt to solve these problems, they proposed an Intelligent Traffic Control System (ITCS) using Radio Frequency Identification.

In [11] a system which used RFID to tackle traffic congestion around toll gates was presented. The system was effective as it saved time and reduce the need for manpower in its operation. Their system however required vehicles to be pre-registered into the system and an RFID tag stuck at a visible area on the vehicle. A limitation not considered is the traffic backlogs that would build up if an unregistered vehicles approaches the toll or when drivers do not have sufficient funds in their account.

Similar to the work done in [5], Gadekar et al. [7] also worked on implementing an Intelligent Traffic Control for Congestion, Ambulance clearance, and stolen vehicle detection using RFID. In their model, every vehicle was to have a RFID enabled device that stored the Vehicle Identification Number (VIN), owner details and priority. Vehicles were then divided into 4 categories. The first had the highest priority and experienced little or no delays. The second included school buses which often need to reach their destination on time hence also required fast service. The third category included the car, motorcycle and scooter. The last category catered for heavy vehicles.

Bommes et al. [3] in their work discussed various camera types ideal for different applications in traffic management. Webcam were best suited for quick overview of traffic status. Surveillance cameras were most suited for monitoring road traffic conditions, traffic counts and possibly identifying causes of congestion. While very high resolutions cameras should be used when detailed vehicular identification such as number plates are needed.

There are many related research on vehicle detection. In [10] and [23] for instance, the authors used a computer vision based traffic counting systems. Cameras were used to capture traffic images and passed through image processing models to obtain vehicle count. The most commonly used approaches for traffic image processing are frame differentiation [1] and Landmarking as used in [9].

Image processing library such as OpenCV [15], Single Shot Multibox Detector (SSD) [12], You Only Look Once (YOLO) [17] have also been used. However, according to [25], most of these algorithms use either segmentation or scale based models for object detection. In segmentation based models, pixel-wise predictions are used to determine if a pixel belongs to an object or not. While in scale-based models, a strong classifier is built and used to determine if an image patch belongs to an object. Repeating the process with different resolutions makes objects of different sizes and aspect ratios detectable. In this paper, SSD is used as it has been shown to have a fast prediction time and works well with low resolution images.

Edge Computing is defined in [20], as a computing paradigm in which computing and storage devices are placed at the edge of networks, close to the data sources (sensors) for the purpose of reducing latency and improved utilization of bandwidth. Edge computing is particularly suited for this work because quick decisions are needed in real time. Sending hundreds of captured images over a network to a remote location for processing and then waiting for response before making decisions would be grossly inefficient [22].

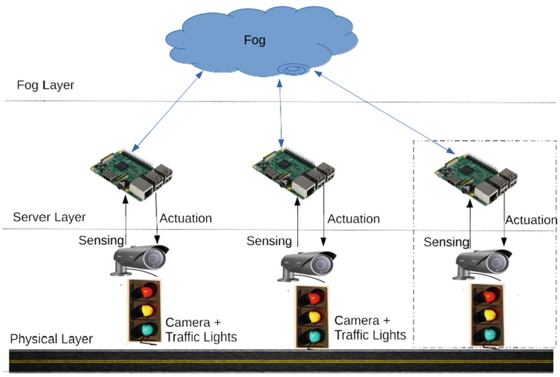

Fig. 1. Cyber-physical traffic management framework (Color figure online)

3 The Cyber-Physical Traffic Management Framework

This paper proposes a cyber-physical traffic management framework called cyTAC. The goal of cyTAC is to introduce human-like decision making process into traffic lights for the purpose of improving the efficiency of traffic management. The proposed framework is shown in Fig. 1 as a layered structure, with layers described as follows:

- **Physical Layer:** This is the lowest layer of the framework and closest to the users. It consists of the traffic lights and cameras. It is responsible for accepting input i.e. sensing the environment and performing specified actions (actuation). Input received are forwarded to the server layer, while output from the server layer are displayed/implemented here.
- **Server Layer:** This layer is responsible for data gathering. It comprises of light-weight computers which accept, analyze and process data received from the physical layer. Output of computational processes are sent back to the physical layer for actuation. Data at this layer are also sent to the Fog layer for storage or further analysis.

– **Fog Layer:** This is the top-most layer of cyTAC and it handles data storage and advanced computation (such as trend analysis and surveillance). It is also responsible for coordinating multiple servers across wide geographical locations.

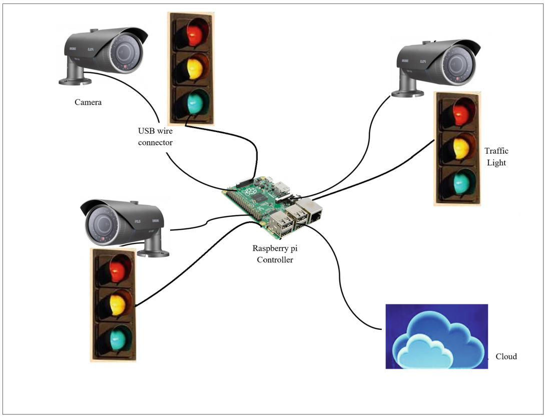

Fig. 2. Smart traffic management system (Color figure online)

In this paper, only a sub-set of the cyTAC framework is presented. This sub-set, enclosed in a dotted block in Fig. 1 is shown in Fig. 2 and referred to as the Smart Traffic Management System (STMS). The rest of this paper would focus on the STMS.

4 Smart Traffic Management System

The Smart Traffic Management System model shown in Fig. 1 is made up of the following parts:

1. **Cameras:** These serve as input source to the system. Though high definition cameras might be desirable, a balance must be struck between crystal clear images and image size. Due to the limited storage capacity of the edge device and because surveillance and/or vehicle identification are not the priority of this work, cameras with capture quality set to 1.3MP were used. Images taken with this resolution were sufficient for the image processor yet minimal in size.
2. **Traffic Light:** The features of the classic traffic lights are retained and used to communicate with the vehicle drivers. It was important to retain the universally understood, red, amber and green lights. The only change made was to the control circuitry, which was changed from basic time switched to a more intelligent image processed switching.

3. **Controller:** This serves as the brain of the system, and where the image processing and switching decisions are done. It is made up of three parts, which are:
   (a) **Input:** Images are received from cameras connected via USB or similar interface
   (b) **Processor:** A Micro-controller such as the Raspberry Pi [19] is to be used. Images captured by the camera are processed here.
   (c) **Output:** Switching decisions obtained from the processed images are sent as output to the connected traffic lights.

4.1 Image Processing

SSD is an object detection algorithm well suited for real-time image identification. It is an extension of the Faster Region Convoluted Neural Networks (FR-CNN) [18] that eliminates the need for the region proposal network. This enables it to be as fast as FRCNN yet utilize lower quality images. A comparison of SSD and some state-of-the-art object detection algorithms has been done in [12] with SSD shown to be better than most. SSD is designed in such away that its primary function is to detect objects present in an image. This unique feature makes it well suited for application in this paper, as the objective is to detect vehicles within an image. SSD works by sliding detection windows of various sizes across an image, it then determines if the windows contains objects of interests or not. The result of this process is a prediction map. As with most computational intelligence algorithms, SSD also has to be trained to identify objects. This training process involves comparing the obtained prediction map against a ground truth (in this case various pictures of vehicles). SSD then compares each presumed object with the ground truth. Matching segments (Intersect over Union (IoU)) are recorded. The higher the IoU, the more likely the presumed object is a target image (in this case a vehicle).

For this work, we are only interested in vehicles, hence our ground truth was made up of pictures of different kinds of vehicles.

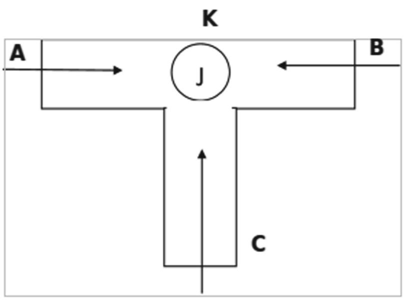

Fig. 3. Sample 'T' road intersection

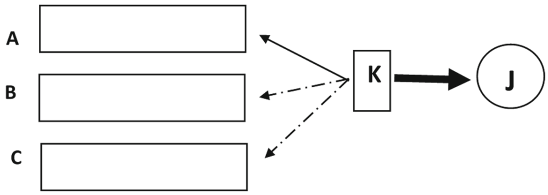

Fig. 4. Road intersection model

4.2 Switching Decision System

The following assumptions are made in establishing the feasibility and reliability of this system:

1. Only T-junctions are considered i.e. three roads connected at an intersection.
2. Each road is single carriage i.e. vehicles move only in one direction on each road.
3. Vehicles have Poisson arrival rates.
4. There are no other factors contributing to traffic congestion (such as bad road or broken down vehicles) except the coordination of the traffic flow.
5. Delays emanating from pedestrian crossings are minute hence not considered.
6. Road switching times are also not considered, that is the time it takes the STMS to switch access between the various roads at the junction.
7. Vehicle depart at a rate of one vehicle per second.

The switching decision system is modelled as follows:

Assume a T-junction, with three connected roads A, B and C, an intersection J and a traffic control light K. This is as shown in Fig. 3. The following are properties relating to the junction:

1. Vehicle Arrival: Vehicles arrive at each road randomly. This can be model using the Poisson model and described in 1

$$VehicleArrival : P_n = \frac{\lambda t^n}{n!} * exp^{\lambda t} \tag{1}$$

   Where λ is the arrival rate, t = time interval and n = number of vehicles, Pn is the probability of n vehicles arriving at a given time.
   J is the common resource all vehicles need to use. However, allowing all vehicles into J would result in traffic congestion, hence a controller K is required. The situation presented in Fig. 3, is thus analogous to a multi-queue single processor system, or M/M/1. It can thus be remodelled as shown in Fig. 4.
2. Vehicle Scheduling and Departure: Traditionally, the traffic controller K operates using a time-based circuit using a Round-Robin algorithm. It gives each road (A, B or C) access to the junction J for a pre-set time interval t. On expiration of t, it pre-empts the current road and gives the next road access to the Junction, This process is repeated continuously and modelled as follows:

Let λ_a = arrival time of vehicles on road A
Let λ_b = arrival time of vehicles on road B
Let λ_c = arrival time of vehicles on road C
Let μ be departure rate = 1 vehicle/second
Let delay (d) be the time a vehicle has to wait **at the front** of a road for a GREEN signal.

After a time interval t has elapsed, the number of vehicles waiting on each road A, B and C would respectively be:
$\alpha = \lambda_a t$, $\beta = \lambda_b t$, $\gamma = \lambda_c t$ from little's theorem 2

$$N = \lambda * t \tag{2}$$

$\alpha = \beta = \gamma$ *iff* all the arrival rates are equal, in which case the traditional turn based fixed time interval Round-Robin would be fair to all roads and the number of vehicles that leave each road would simply be the time frame allocated per round.

3. Vehicle Delay Time: Let μ_a, μ_b, μ_c be the departure time for each road. The number of vehicles that leave each road per round would thus be:

$$\alpha' = \mu_a t, \beta' = \mu_b t, \gamma' = \mu_c t \tag{3}$$

Traditional traffic lights using Round-Robin assumes that all μ are equal therefore the delay experienced by a vehicle waiting at the front of a road for a GREEN signal when round-robin is used can be given by:

$$d = 2 * \mu * t \tag{4}$$

Let n = number of rounds (i.e. from A to C and back). If at a time t = 0, road A is given access to J; then: At time 0 + t, A is pre-empted and access is given to road B. At time 0 + t + t = 2t, B is pre-empted and access given to road C. At time 0 + t + t + t = 3t, C is pre-empted and access is returned to road A.
From this, road A would next have access to J at time (n*3t) and wait for a period of n*2t. Similarly, road B would next have access at time (n + 1)*3t and wait for a period of (n + 1)*2t. Finally, road C would next have access to J at time (n + 2)*3t and wait a duration of (n + 2)*2t.

If arrival times were equal across all three roads, then the system would be completely fair to all roads and there would be equal number of vehicles on each road. However this is not always the case, as in reality arrival processes are usually random. Unfortunately, the Round Robin system used in traditional traffic lights does not compensate for this but rather stick to a constant service rate. This invariable results in an unbalanced system and situations were certain roads with short or no queue are being serviced to the detriment of others having long(er) queues. Thus vehicles at the front of a road would have to wait at least $2\mu t$ even if the other roads are empty.

In solving this, STMS takes vehicular density into consideration and in essence converts the Round-Robin to a Priority-Queue. Using this, roads judged to have longer queues are given immediate access to the junction. Using this approach alone would lead to starvation for roads with shorter queues. To avoid this, STMS keeps track of the number of times each road is passed, and progressively increases the priority of the longest waiting road. In essence STMS can be described as a Priority-Queue + Suffrage. The STMS can thus be modelled as follows:

1. Vehicle Arrival: This is similar to the traditional Round-Robin and modelled with 1
2. Vehicle Scheduling and Departure: Similar to the traditional Round-Robin, after a time interval t has elapsed, the number of vehicles waiting on each road A, B and C can also be obtained using 2
3. Vehicle Delay Time: The number of vehicles leaving each road per round is obtained using 3

STMS however improves on the short-coming of Round-Robin by dynamically adjusting the wait time. The delay experienced by a vehicle waiting at the front of a road A, B or C for a GREEN light when STMS is used is be given by:

$$delaytime(d) = \begin{cases} 0, & \text{if the other roads have no queue} \\ \mu t, & \text{if any other roads has longer queue} \\ 2\mu t, & \text{if the other roads have longer queues} \end{cases} \tag{5}$$

Where μ and t are respectively the vehicle departure rate and time elapsed.

From 5, if a vehicle arrives at an intersection:

1. At the best case it would immediately be given the GREEN signal, if there are no waiting vehicles on the two other roads.
2. On the average it would wait μt for the road with the longer vehicle queue.
3. At the worst case it would wait $2\mu t$, as is the case with the traditional Round-Robin based system.

This implies that STMS is better than Round-Robin in 2 of the 3 scenarios and at par in the worst case scenario.

5 Phases of STMS

Smart Traffic Management System (STMS)'s process flow is shown in Fig. 5.

5.1 The Phases of STMS

The system goes through different phases before decisions are made and communicated to the vehicles. These phases are described as follows:

1. Image Capturing: This is the point of entry into the system. For every round of decision, images are first captured. The cameras mounted at strategic locations take shots of the roads and send the images to the controller for processing.

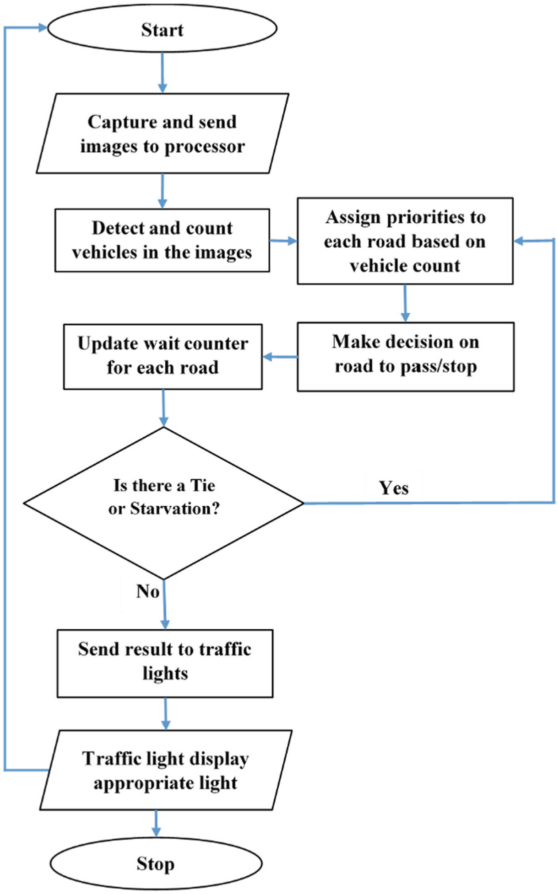

Fig. 5. The STMS process flow

2. Image Detection and Processing: The images are received from the cameras and vehicle detection is done on each of them using the SSD algorithm. The image detection process identifies and counts the vehicles in the image and prioritizes them accordingly. This means that for every image received, there is a priority attached based on the number of vehicles counted.
3. Result Processing: On completion of the detection and prioritization process, a Green signal is sent to the road with the highest priority, while Red signals are sent to the other roads. A time interval is also set for which the signal is to be displayed. After each decision process, a wait counter is updated for each road not passed (Red signal shown). This update, gradually increases the priority of such roads. This is used to prevent starvation or pro-longed denial of service to such roads as well as to break ties (situation where two roads have the same number of vehicles waiting).

4. Optionally, the results obtained alongside images and other useful statistical data could be sent to the Cloud to conserve storage on the Edge device and possibly for further processing.

6 System Implementation

A software prototype was developed using Java and implemented on a system running Windows 10, with 6 GB of RAM and a Core i5 processor. The SSD object detection library used was obtained from Github [6].

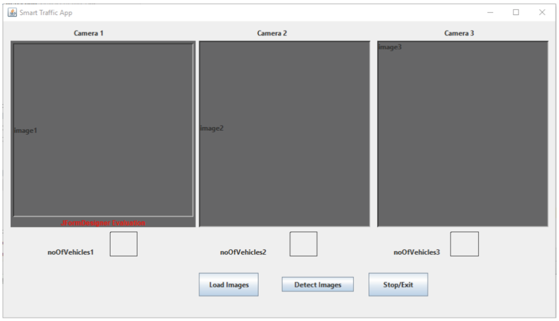

Fig. 6. Interface smart traffic management system

Fig. 7. Smart traffic management system with images loaded

To test the functionality, numerous pictures were captured at different times and at various T-junctions. These pictures were cropped to equal dimensions and saved in a directory on the system. The interface of the program is shown in Fig. 6.

At the start of the simulation, the program randomly selects 3 pictures from the image pool. This is synonymous to the controller receiving pictures from the

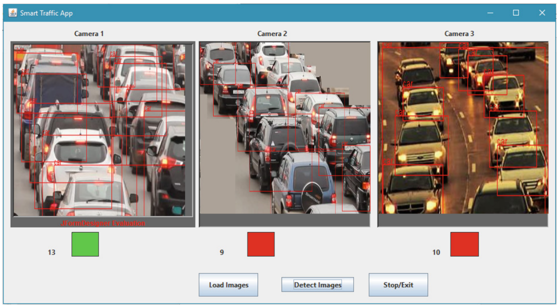

Fig. 8. Smart traffic management system with vehicle detection (Color figure online)

attached cameras as shown in Fig. 7. The images are then passed through the SSD object detection algorithm, which detects and counts the number of vehicles present in the 3 images. The obtained results are compared, and the image with the highest number of vehicles gets the GREEN signal. This is as illustrated in Fig. 8.

The image depicted in Fig. 8 shows that the application performed well in terms of identifying and counting the number of vehicles in the supplied images using SSD. The image supplied by Camera 1 had the highest number of vehicles (13) and as such was sent the GREEN signal. It must however be noted that the SSD object detection algorithm was not 100% accurate. For instance in Fig. 8, the image supplied by camera 2 had 11 vehicles, however only 9 were detected. This is possibly because the last two vehicles not detected were further up in the image and not in the camera's perfect line of sight. Increasing the height and re-positioning the angle of the camera could ensure that the captured images are aligned within the view of the camera.

7 Performance Evaluation

Experimental simulations were also carried out for the purpose of determining the efficiency of the proposed model. Five different scenarios were simulated. In the first, arrival rate was set to be half of the service rate, while the second was a variant of the first, wherein adaptive service rates were used. Using adaptive service rate, STMS was able to dynamically changes the service rate for each lane based on the number of vehicles queued. In the third, equal values of arrival and service rate were used, while in the fourth simulation, the arrival rate was set to twice the departure rate. Finally for the fifth experiment, random arrival rates were used.

For each simulation 200 traffic were generated and since the model only considered a T-junction, a simulation run consisted of circling through three (3) lanes. Vehicles were then processed (scheduled) using Round-Robin (as in the case of the traditional traffic lights) and STMS as proposed in this paper. A comparison of both approaches was done using throughput, makespan, service interruptions and fairness index as metrics. These metrics are defined as follows:

1. Throughput: The number of vehicles processed per round and defined as Vehicles Serviced/Number of Rounds. A round is defined as a complete circle through the three lanes. A higher throughput value is desirable.
2. Jain's Index: In order to compare the fairness of the STMS model, Jain's Index [8] is used and shown in 6.
3. Makespan: The amount of time needed to process the last vehicle in the system. A lower Makespan value is desirable.
4. Service interruptions: The number of times service to a given lane is interrupted in favour of a different lane. Lower number of service interruptions are desirable.

$$f(x) = \frac{\sum_{i=1}^{n} x_i^2}{\sum_{i=1}^{n} x_i^2} \tag{6}$$

With Jain's Index, $0 \leq f(x) \leq 1$, the closer the value of f(x) is to 1, the fairer the system. The results of these simulations are now presented.

7.1 Arrival Rate Equals Half Departure Rate

For this experiment, the number of vehicles arriving at each lane per unit time is set to exactly half the number leaving. The following were observed from this experiment:

1. When departure rate (service rate) is greater than arrival rate, there would be no queue.
2. Similar results are obtained when arrival rate is equal to departure rate.
3. Round-Robin is fair to all the lanes, as vehicles on all the lanes experience equal wait time.
4. STMS (without adaptive service rate) behaves exactly like the traditional Round-Robin.

If however, adaptive service rate is enabled, that is the service rate changes dynamically with respect to the length of each lane, then STMS results in a 50% reduction in delay for all lanes compared to the traditional Round-Robin. This is as shown by the first two columns in Fig. 9. Only make span is shown as all other metrics had similar values.

7.2 Arrival Rate Equals Double Departure Rate

When arrival rate was set to exactly double the value of the departure (service rate), results similar to those of equal arrival and departure rates were obtained. This is as depicted by the last two columns in Fig. 9.

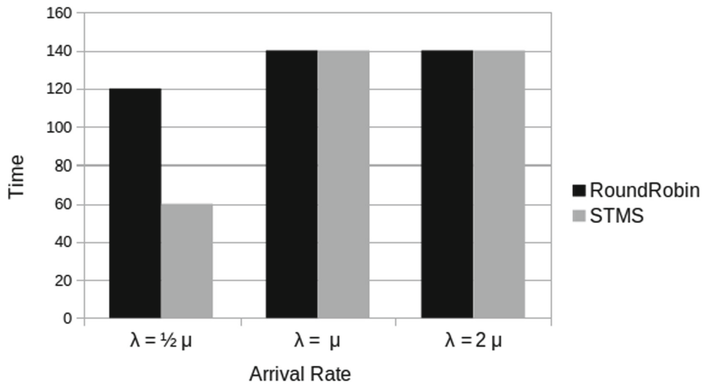

Fig. 9. Variation of Makespan with Arrival rate

7.3 Random Vehicle Arrival

For this experiment, at each run, a number of vehicles were randomly assigned to each lane. This was used to simulate the Poisson arrival of vehicles at an intersection. It is believed that gives a result closer to reality. Other viable models, though not considered in this work are: negative exponential, normal and pearson distribution [13].

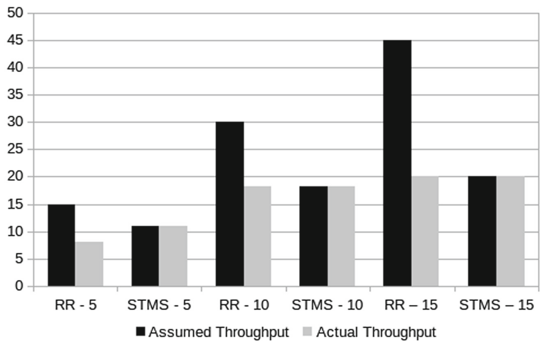

Fig. 10. Throughput comparison

Figure 10 shows a comparison of the throughput values of Round-Robin Scheduling versus STMS based scheduling. In the figure three scenarios are presented; in the first a service rate of 5 vehicles per lane was used, this was increased to 10 and 15 in the second and third scenarios. In the first scenario, the Round-Robin scheduling (RR-5) had an actual throughput of 8 vehicles/round and an assumed throughput of 15 vehicles/round. In the second scenario (RR-10), actual throughput increased to 18 while assumed was 30. In the final scenario with a service rate of 15, RR-15 recorded an actual throughput of 20 versus assumed throughput of 45. Across all three scenarios, Round-Robin resulted in an average of 50.6% lower actual throughput than assumed throughput. This in essence implied that Round-Robin is inefficient with about 50% of its service turns wasted. On the contrary, in all three scenarios, the assumed and actual throughputs of STMS were a perfect match implying zero wasted turn. In this paper, assumed throughput is defined as the number of vehicles assumed to be serviced divided by the number of rounds. In essence, it is approximately the time during which the traffic light is denying service to lanes that have vehicles queued up, in favour of an empty lane.

Round-Robin is known to be a fair algorithm, as it provides all lanes equal access to the intersection for an equal amount of time. To verify the fairness of STMS, the Jain's index was used along side the average throughputs for each of the three service rate scenarios described above. The obtained results are as follows: for the scenario where service rate was set to 5, Round-Robin had an average actual throughput of 8, while STMS has 11. Substituting these values into 6, resulted in a Jain's index value of 0.976. For the two other scenarios, where service rate was set to 10 and 15 respectively; Round-Robin and STMS had equal average throughput values, this resulted in a Jain index value of 1 in both cases. With an index value of approximately 1 in all tested instances, it can be concluded that STMS is as fair to all lanes as Round-Robin is.

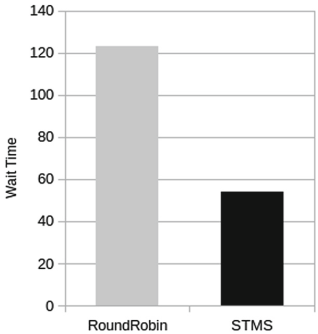

Fig. 11. Comparison of overall Makespan

In Fig. 11, a comparison of the Makespan for both algorithms is shown. Makespan is calculated by multiplying the number of rounds utilized by the number of lanes and service rate (number of vehicles served per run). In this work, service rate is assumed to be one vehicle per second. From Fig. 11, after 200 simulations, the average Makespan for Round-Robin was 123 s, while that of STMS was 54 s. This shows that using STMS on traffic lights, vehicular delay at traffic intersections can be reduced by as much as 56% versus when traditional Round-Robin is used.

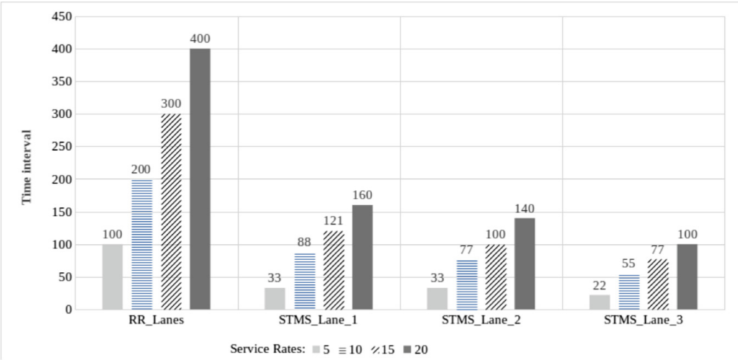

Fig. 12. Per lane delay for random arrival rates

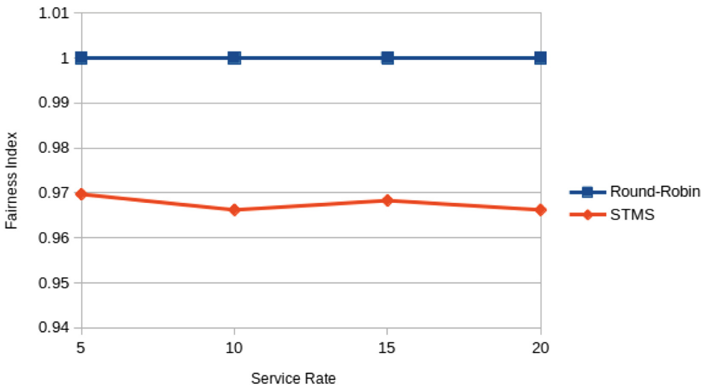

Fig. 13. Comparison of fairness indices

The result shown in Fig. 11 is however a reflection of the overall makespan (delay) of the system. It does not reflect the actual delay on each lane. Owing to the fact that vehicles were randomly assigned to lanes, the obtained result was very sporadic and did not follow any given pattern. To structure the values, a vehicle arrival ratio of 2:3:5 for the three lanes was used and the results of this when compared to the traditional Round-Robin are shown in Fig. 12.

In Fig. 12, the delay on each of the three lanes are compared for both Round-Robin and STMS. Four different service rates were used - 5, 10, 15, 20; and these represented the number of vehicles to serve on each lane. For Round-Robin all three lanes experienced the same amount of delay, and this is represented by the RR_Lane. Using STMS, each lane experienced different delays. Ideally, lanes with the shortest queue should experience the maximum delay (due to starvation), but with suffrage limiting feature of STMS, the effect is minimal. The difference between the waiting times for all three lanes is not too pronounced. Particularly, for small service rates (5 for instance), the delays experienced by both lanes 1 and 2 are exactly the same, while lane 3 is not too far off. The fairness indices for both the traditional Round-Robin and STMS, across the four service rates tested are as shown in Fig. 13.

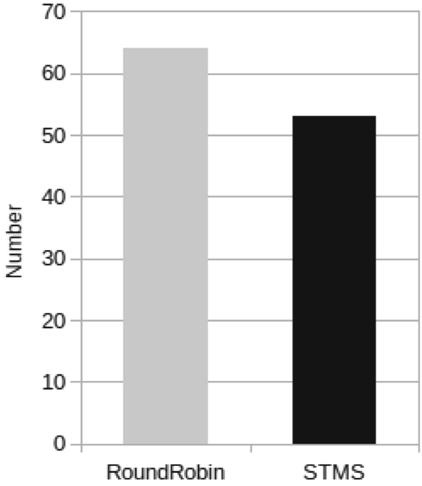

Fig. 14. Service interruptions

Finally, a comparison of the average number of service interruptions of both algorithms was done. The results shown in Fig. 14, shows that after 200 iterations, STMS with an average of 53 interruptions is about 17% better than Round Robin's 64.

8 Conclusion and Future Work

Road traffic congestion is a problem that plagues many societies today. In developing cities, the explosion in number of vehicles and the unmatched infrastructural development results in seemingly unending traffic gridlock on a daily basis. Commuters spend an average of between 30 to 60 min in traffic jams daily. The results of this are wasted man-hours, delays in delivery of raw materials to factories, reduced production powers, among others. These impact negatively on the economy and on human health. The major causes of these traffic gridlocks have been traced to road intersections. Various approaches have been used over the years to tackle this problem, prominent among them is the use of traffic control lights at road intersections. These lights have however proved inefficient as they operate on pre-set time based Round-Robin model and lack human intuition needed to intelligently adjust scheduling and adapt to the traffic situation in real-time. This work proposed a cyber-physical traffic management framework and more specifically a sub-set of this framework called the Smart Traffic Management System (STMS). STMS can run on a light-weight edge device, and utilizes image detection to dynamically alter traffic flow in real time. T-junctions was modelled mathematically and simulations carried out with obtained results showing an average reduction in delay time and service interruptions by as much as 56% and 17% respectively when compared to Round-Robin used in traditional traffic light systems. Overall STMS was at least 50% more efficient in terms of vehicular throughput and delay that Round-Robin yet remained equally as fair. In this work, only software simulations were carried out, however construction of actual physical prototypes would be desirable in future works. An ideal road situation with no vehicle breakdowns or bad roads was assumed. These are interesting factors that plague developing countries and could be considered. Also this paper treated all vehicles with equal priority; in the future, a system that prioritizes special vehicles such as ambulances and emergency services could be considered. Finally, this work focused on a sub-set of the cyber-physical framework, concentrating on detecting and counting vehicles at one intersection; future systems could be designed to consider multiple traffic lights connected to a Fog. With this a lot more information can be captured, analyzed and stored, thus opening up a vista of opportunity for applications in analytics, security and urban city planning.

References

1. Bandarupalli, V.: Evaluation of video based pedestrian and vehicle detection algorithms. Ph.D. thesis, University of Nevada (2010)
2. Bashiru, A., Waziri, O.: Analysis of intra-urban traffic problems in Nigeria: a study of Lagos metropolis. Indonesian J. Geogr. **40**(1), 997–1004 (2008)
3. Bommes, M., Fazekas, A., Volkenhoff, T., Oeser, M.: Video based intelligent transportation systems - state of the art and future developments. Transp. Res. Procedia **14**, 4495–4504 (2016)
4. Bramberger, M., Pflugfelder, R., Maier, A., Rinner, B., Strobl, B., Schwabach, H.: A smart camera for traffic surveillance. In: Proceedings of the 2003 First Workshop on Intelligent Solutions in Embedded Systems, pp. 153–164 (2012)
5. Chattaraj, A., Bansal, S., Chandra, A.: An intelligent traffic control system using RFID. IEEE Potentials **28**(3), 40–43 (2016)
6. Chen, X.: Java SSD object detection (2018). https://github.com/chen0040/java-ssd-object-detection
7. Gadekar, T., Chavare, P., Chipade, K., Togrikar, P.: Implementing intelligent traffic control system for congestion control, ambulance clearance, and stolen vehicle detection. Imperial J. Interdisc. Res. **2**(4) (2016)
8. Jain, R., Durresi, A., Babic, G.: Throughput fairness index: an explanation, pp. 1–13 (1999)
9. Hsieh, J.-W., Yu, S.-H., Chen, Y.-S., Hu, W.-F.: Automatic traffic surveillance system for vehicle tracking and classification. IEEE Trans. Intell. Transp. Syst. **7**(2), 175–187 (2006)
10. Kastinaki, V., Zervakis, M., Kalaitzakis, K.: A survey of video processing techniques for traffic applications. Image Vis. Comput. **21**(4), 359–381 (2003)
11. Khatile, R., Gore, P., Gagre, V., Deshmukh, N.: RFID based traffic control and toll collection. Int. J. Adv. Eng. Res. Dev. **3**(1), 345–360 (2016)

12. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
13. Mathew, T.: Vehicle arrival models: headway. Transp. Syst. Eng. **39** (2014)
14. Osigwe, C., Oladipo, F., Onibere, A.: Design and simulation of an intelligent traffic control system. Int. J. Adv. Eng. Technol. **1**(5), 47 (2011)
15. Pulli, K., Baksheev, A., Kornyakov, K., Eruhimov, V.: Real-time computer vision with OpenCV. Commun. ACM **55**(6), 61–69 (2012)
16. Raheem, S., Olawoore, W., Olagunju, D., Adeokun, E.: The cause, effect and possible solution to traffic congestion on Nigeria road (a case study of Basorun-Akobo road, Oyo state). Int. J. Eng. Sci. Invention **4**(9), 10–14 (2012)
17. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
18. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
19. Richardson, M., Wallace, S.: Getting Started with Raspberry Pi. O'Reilly Media Inc., Newton (2012)
20. Satyanarayanan, M.: The emergence of edge computing. Computer **50**(1), 30–39 (2017)
21. Ukpata, J., Etika, A.: Traffic congestion in major cities of Nigeria. Int. J. Eng. Technol. **2**(8), 1433–1438 (2012)
22. Varghese, B., Wang, N., Barbhuiya, S., Kilpatrick, D., Nikolopoulos, S.: Challenges and opportunities in edge computing. In: Proceedings of the 2016 IEEE International Conference on Smart Cloud, pp. 20–26 (2016)
23. Velastin, N., Orwell, J.: A review of computer vision techniques for the analysis of urban traffic. IEEE Trans. Intell. Transp. Syst. **12**(3), 920–939 (2011)
24. Yadav, R., Mishra, A., Prakash, N., Sharma, H.: An improved round robin scheduling algorithm for CPU scheduling. Int. J. Comput. Sci. Eng. **2**(4), 1064–1066 (2010)
25. Yadav, V.: Towards a real-time vehicle detection: SSD multibox approach