Intelligent Traffic Management System for Cross Section of Roads Using Computer Vision

Tousif Osman, Shahreen Shahjahan Psyche, J. M. Shafi Ferdous, Hasan U. Zaman
Department of Electrical and Computer Engineering
North South University, Dhaka, Bangladesh
tousif.osman@northsouth.edu, shahreen.psyche@northsouth.edu,
shafi.ferdous@northsouth.edu, hasan.zaman@northsouth.edu

***Abstract*—This paper includes the design and implementation of an intelligent and automated traffic control system which takes advantages of computer vision and image processing techniques. Along with conventional computer vision techniques, this paper introduces two new methods which has low processing cost. One of the methods has been constructed with the help of hardware and the other one is designed without hardware support. This is a complete traffic management system which has been able to reduce traffic jams and congestion on simulated environment. It detects the number of vehicles on each road and depending on the vehicles load on each road, this system assigns optimized amount of waiting time (red signal light) and running time (green signal light). This system is a fully automated system that can replace the conventional pre-determined fixed-time based traffic system with a dynamically managed traffic system. It can also detect vehicle condition on road and auto-adjust the system according to the changing road conditions which makes the system intelligent. The designed system can help solving traffic problems in busy cities to a great extent by saving a significant amount of man-hours that get lost waiting on jammed roads. This research focuses on factors, low-cost image processing and traffic load balancing.**

Keywords—Traffic Jam; Traffic Management; Computer Vision; Image Analysis; Vehicle Detection; Object Identification; Load balancing

I. INTRODUCTION

One of the byproducts of the rampage of technological development, advancement and modernization is Traffic Jam. Nowadays, it is a very common phenomenon in our daily lives. This is specially a huge problem in congested cities like Dhaka where traffic jam is actually one of the larger social problems. A study has found that people yearly lose 8.15 million hours, 40 percent of which are working hours –in traffic jams [1]. In cities of developing countries like Bangladesh, traffic is mainly controlled by two ways. One way is to use traffic lights and change the lights at a pre-determined fixed time intervals and in the other way traffic polices do the work manually. Both of them are wasteful processes. Sometimes in a cross-section of roads (junction) we see that one side has lower congestion than the other side, but as the lights are changing at fixed time intervals, the jam on the road which has higher congestion keeps getting worse. On the other hand, most of the traffic polices are unaware of the situation at places away from him/her. It is also very hard to look at every side of a junction constantly by one person and decide correctly how to guide the traffic. However in developed countries, automation has been introduced in traffic management. At present, there are two available approach to control traffic. One is by sensing vehicles with pressure plates on road and the other one is by using RFID tags on number plates and placing RFID readers on road. Both methods are very expensive to implement when the traffic jam is very long. Another approach is surveillance camera based traffic system where traffic is monitored and manually controlled from a control room. In this case real-time image analysis is applied on the live video footage to aid the personals in the control room with traffic observation. This is the most adopted method for traffic controlling these days. In our research we are proposing a better solution to this problem. Using computers' visual ability and analytical capability we have built a system which can decide which road needs to be cleared quickly and dynamically change the time intervals and changing the signal accordingly and automatically. In our research we have constructed a complete, fully automated and intelligent traffic management and control system for road intersections.

II. RELATED WORK

While working on this research project, we have encountered several researches which have been conducted on traffic, analysis of traffic and optimization of traffic. We have studied those research and incorporated ideas, problems faced and their solutions to our system from those research. In this section we have briefly described those research works that we have integrated in to our system.

The most similar work with our research work is paper [2]. In this paper authors has applied almost a similar procedure like ours but the outcome of their research is different compared to ours. First they have trained background images using two methods – i) Moving Average and ii) Codebook. After that they have subtracted the foreground traffic image with the trained background image and extracted the interest area of analysis. This is done in grayscale mood. Next they have used a threshold value and connected component methods to find the area of the traffic regions. Moreover, they have extracted the region of interest from the background image so that they do not have to encounter unnecessary noise. Furthermore, they have classified the roads by counting the area of the subtracted image. After that they have taken the ratio of calculated area with the region of interest. Finally they have classified which road has more traffic than another using the value of the ratios for different roads.

On the other hand, the second research [3] we are referring, has used image analysis to solve the problem, which is more close to our research proposal. In this research they have worked on live video streams of roads. Though this algorithm is very good in terms of vehicle counting accuracy but it comes with a good cost of processing power. To run the above algorithm in real time we need a significant amount of processing power to analyze the video feeds of an entire city. Considering that in mind we have only used pattern detection combined with components analyzing technique, which is pretty straightforward and less heavier algorithm. However, with our approach accuracy drops slightly but we can ignore the fact as we don't need exact number of vehicle to provide efficient and effective traffic management system.

In another research, fuzzy logic technics have been used to solve traffic congestion problem [4]. Fundamentally this research work have introduced a fuzzy controller system for an isolated four-way junction. This total system is quite distinct from us but solves almost the same problem. The main difference is that they are working on a single isolated junction and calculating the extension time accordingly while we are actually considering other junctions which are directly connected with the certain junction. By doing this we are actually calculating the running time of a single node more precisely according to the neighborhood area

The next paper [5] discusses a new approach to minimize traffic congestion. They tried to replace the traffic light based roadside infrastructure with in-vehicle virtual signs where only vehicle to vehicle communication is allowed. In our conventional traffic controlling system, intersections are crucial and controlled by traffic lights which have high establishment and maintenance cost and traffic lights also cover less number of total intersections. On the contrary, in their system design, they have considered a virtual traffic light (VTL) protocol which will optimize traffic congestion dynamically using Virtual AdHoc Network (VANET) using vehicle to vehicle (V2V) communication. Here, each vehicle will be installed with an application unit (AU) which will contain the database of traffic intersections where the closest vehicle to the intersection will be able to create a VTL. The vehicles near to a junction will select a leader and obey the directions provided to the leader. As a result, according to this research we won't need any traffic lights to control intersections and this will lower traffic congestion in urban areas.

A further research [6] has been conducted on traffic management for emergency services. This research focuses on reducing the emergency service latency with minimal disruption to regular traffic. Initially this research identifies 3 level of emergency i.e. high, medium, low. Based on the emergency level, an adaptive framework has been designed to alter the traffic rules and the emergency vehicle will be given the faster route to its destination. This framework first verifies the emergency vehicle and next the level of emergency. Using these two central traffic controller provides the faster route to the onboard display of the emergency vehicle. We have adapted this framework's traffic management protocol to our system.

On the next paper [7] that we are adapting into our system focuses on decreasing greenhouse gas emission due to the idle vehicles at traffic intersections during traffic congestion. They have designed a system called the "Eco-Sign" system which will be able to dynamically turn engines on or off while the vehicles remain at a traffic signal. The system will include a localization unit (LU) which will be implanted at a traffic signal. A traffic control unit (TCU) will be connected with the traffic lights. All of the vehicles on the road will be equipped with a vehicle unit (VU) which will maintain connection with LU and TCU. When any vehicle enters any traffic signal zone, it's VU will get information from the TCU. In accordance with the timing of red light and green light, the onboard VU of those vehicles will turn the engines on and off automatically.

We have also incorporated the following paper [8], where a hardware and a Video based traffic management system has been developed to monitor traffic. In this research a monitoring device has been constructed to produce a cost effective replacement of CCTV camera. This device has been designed with camera phones and they have been powered by solar energy. We have developed our traffic motoring hardware and software with the help of this research.

III. Theory

We intend to make a system that minimizes the waiting time and maximizes the running time of traffic lights. The system is intended to identify number of vehicle on each traffic node (embedded device with traffic lights and other components to control traffic on each road) with computer vision. All the traffic nodes are connected to a central server. System considers roads leaving a traffic signal as outgoing edge and roads coming towards a traffic signal as incoming edge. By considering number of waiting cars on each road connected to a traffic junction, system computes minimal waiting time and maximum running time for a specific node.

Primary areas of this research is to identify number of vehicles using computer vision, optimize time by using the vehicle number on nearby connected traffic signal, develop a light intensity varying mechanism to make the vision algorithm to work independent of surrounding environment.

A. *Optimal Time computation*

Initially all nodes in the system have infinite waiting time and zero running time. System has a starting node predefined in the network. The system also has two thresholds for waiting and running time for a specific junction of nodes where nodes are connected by incoming and outgoing edges. For determining the waiting time for a specific node, the system considers nodes connected to the junction with incoming edge and number of waiting cars on those nodes. The system sums up the number of waiting cars and makes a tentative ratio for each node. Finally the system divides the threshold with the assigned ratio for each node connected in the junction and determines waiting time. For running time, by using the same algorithm for waiting time the system assigns running time for each node connected in the junction. In this case, system considers incoming nodes instead of outgoing nodes. Computation on a junction:

$$S_{out} = \sum_{i}^{n} C_{out}$$

$$S_{in} = \sum_{i}^{n} C_{in}$$

$$T_W = \alpha_W * \frac{W}{S_{out}} \quad (1)$$

$$T_R = \alpha_R * \frac{W}{S_{in}} \quad (2)$$

Where α_W waiting time threshold and α_R is the running time threshold. S_{out} represents sum of all waiting cars on the outgoing nodes and C_{out} represents number of waiting cars on each outgoing nodes. Again S_{in} represents sum of all waiting cars on the incoming nodes and C_{in} represents number of waiting cars on each incoming nodes. W is the number of waiting cars on the current node. T_w is the waiting time for the current node and T_R is the running time for current node.

B. Car detection with Computer Vision

For car detection, the system takes images of free road on a specified time interval depending on the traffic pattern of the specific road. This suitable time interval is needed to be predefined during the system setup. To identify car on a given time the system takes a snapshot of the road with vehicles and subtracts it with the image of empty road. When absolute of subtracted image is compared with a threshold value, background pixels on the image is discarded and only the pixels of foreground or vehicle are considered. Afterwards if we apply connected components algorithm on the previously processed image individual objects, can be considered as vehicle, are identified. Finally total number of vehicle on a given image can be identified by counting individual vehicle.

$$[d] = [\text{Image without cars}] – [\text{Image with cars}]$$

$$[\eta, c] = \text{connectedComponents}([d])$$

$$T = \sum_{1=0}^{\eta} 1 \quad (3)$$

Here square brackets represents image matrix. η represents number of connected components and c represents all the identified connected component as a matrix. We are considering "connectedComponents([d])" as a function that take matrix as input and cluster out the connected regions and returns a matrix with a one degree greater detention containing each cluster.

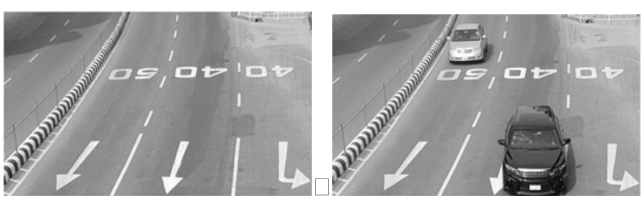

Fig. 1. Image of empty road and road with vehicles

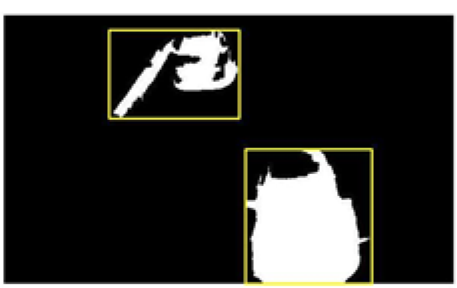

Fig. 2. Resulting image with idefied vehicles after image analyses

We have collected images of road and analyzed those images with our algorithm. Fig.1 is an image of empty road and image of same road on the same position but with vehicles. Fig.2, is the resulting image after image analysis.

C. Light intencity varing Image detection algorithm

To identify vehicles, image of empty road is needed for our algorithm but it is not possible to achieve empty road image on every signal event. Instead of taking background images on every signal event, image of empty road is taken on a specific interval, for instance a road will be blocked by its connecting roads traffic lights early in the morning or on a time when the traffic pressure is relatively low and image of that road will be stored in the system. System will use that image as image of empty road for vehicle counting for next 6 hours. However as light intensity in the day time changes based on the position and weather condition, background image captured hours ago cannot be used directly in the image subtraction process. To solve this issue system pre-process the background image to match with current time's light intensity. We have designed 2 methods in the system to preprocess the background image. One is hardware solution and other one is software solution.

As the intensity of light changes linearly on a given frame of image, we can calculate a coefficient by which we can change the intensity of the background image. If we further look into the images with vehicle, we can observe that portion of top, top left and top right of the image will be out of the boundary of road for some roads, especially for long shot of roads where traffic congestion is high. In Fig. 3 [9] regions defined by 1, 2, 3 are the out of road boundary regions. This region will be constant for every shot when the position of camera is fixed. System admin can define this region for every signal nodes. System then subtracts these three regions of the background image from the current image and gets separate subtracted matrix for every region. Finally system will take the average of every cell of matrix for all regions.

In case of situations where the image frame does not get out of the boundary of road, we have designed a second approach. In this method signal nodes are equipped with light sensors. System measures the presence of light every time an image is captured. Furthermore the system is trained by taking images of empty road on short time interval. System measures the intensity change in the image and the measurement change in the light sensor while training. From this pair of actual and measured change, system calculates the coefficient by which the actual intensity changes for one unit of measurement change in the sensor. This coefficient is constant for each geographic location. Finally system compares the sensor measurement in the background image and the current image and gets the change in measurement. Finally this measurement change is multiplied with coefficient value of that geographic region and gets the intensity change for the given image. In our system we have used Light Dependent Resistance (LDR) as light sensor and the sensor measurement we received as voltage (V).

$$I = (\Phi_i - \Phi_b) * C_\Phi \quad (4)$$

Above equation represents the intensity calculation for hardware solution. Here I represents the intensity change for the background image and current image, Φ_i represents the sensor measurement for current image, Φ_b represents the measurement for background image and C_Φ represents the regional intensity constant found from the training of the system. Form our research we found the software solution gives better result but it cannot be used in every situation where hardware solution has less accuracy.

Fig. 3. Out of road boundry reagions

In the low light condition where vehicle is barely visible, they are bound to have headlight by law. So, instead of identifying vehicles itself, system identifies headlights of cars and divides the total number of lights by 2 that gives a tentative measurement of vehicles on the traffic node. Some vehicles may have one light i.e. motor bike, but contribution to the road occupancy is low. So the error due to miss counting these vehicles has very low impact on traffic congestion.

IV. System Design

There are two major components of the system. Small embedded device to control the traffic lights and capture images from the road. Another component to process images and perform time optimization centrally. We have considered the embedded device as traffic node and distant central component as server.

A. Image Processing:

We have designed the system in such a way that the image processing is done on the server, as processing the image on the embedded devices will increase the cost of the node and hence will increase the cost of the entire system in a broad scale. Processing the image on the server makes the system cost efficient and at the same time this will allow the system to process more images simultaneously with less amount of time.

B. Network Settings:

To build the network between server and the node, we have used http protocol and network techniques as it has well defined communication protocols for images and other data as well. To reduce the power consumption of each node, instead of allowing direct wireless communication with the server, the nodes in a junction are connected with a single router and that router communicates with the server wirelessly or via wire.

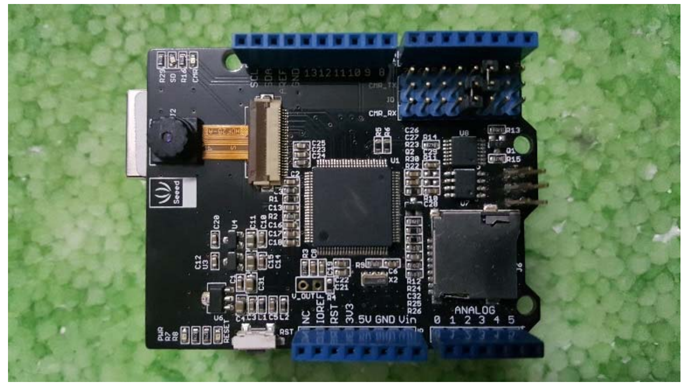

Fig. 4. A node consisting of Arduino uno, Ethernet Shield and Camera Shield with OC0706 camera. This is the top view of a node of the system.

C. Configuration of a Single Traffic Node:

A traffic node [Fig. 4] contains four different components. They are- one camera, three signal lights, an Ethernet interface and a microcontroller and a junction has an additional component, a router. Our system counts car by analyzing images of street and to do so we need a component to capture images and hence we need cameras. In our system design we have used VC0706 cameras. Cameras capture images depending on the instructions provided by the server. The camera in a node is connected to a microcontroller. The microcontroller acts as the single platform on which, the traffic node is designed. As microcontroller we prefer to use ATmega328P microcontroller which has 14 input-output pins. Using these pins, a camera communicates with the microcontroller. Arduino UNO has this microcontroller integrated in it. So we have used Arduino UNO as microcontroller platform. We have chosen Arduino over other microcontrollers because we need a microcontroller which can command the camera shield to capture an image and send it to the server. We do not need powerful microcontroller for our purpose. Hence Using Arduino is energy and cost effective. An Ethernet interface is used to connect the microcontroller with the internet through the router. For this purpose, we have used W5100 Ethernet Shield which is an Arduino compatible Ethernet module. The Router connects all of the nodes around a junction. It establishes a connection between the server and the traffic junction using http protocol. Signal lights of three different colors remain on or off during the time interval provided by the software, which is calculated using the Optimal Time Calculation algorithm. We are using LED lights as signal lights.

D. Working Principle of the System:

Each road connected to a traffic junction has its own traffic controller. Each junction has a wireless transmitter to broadcast images to server. Server has a graph of the traffic system. Each listed node on the graph has their location info and associated MAC/IP address. Initially controller device is set to red light. Server software requests for live image to traffic controller on certain time division via its IP. On request traffic controller returns live image feed to the IP/PORT from which the request was sent.

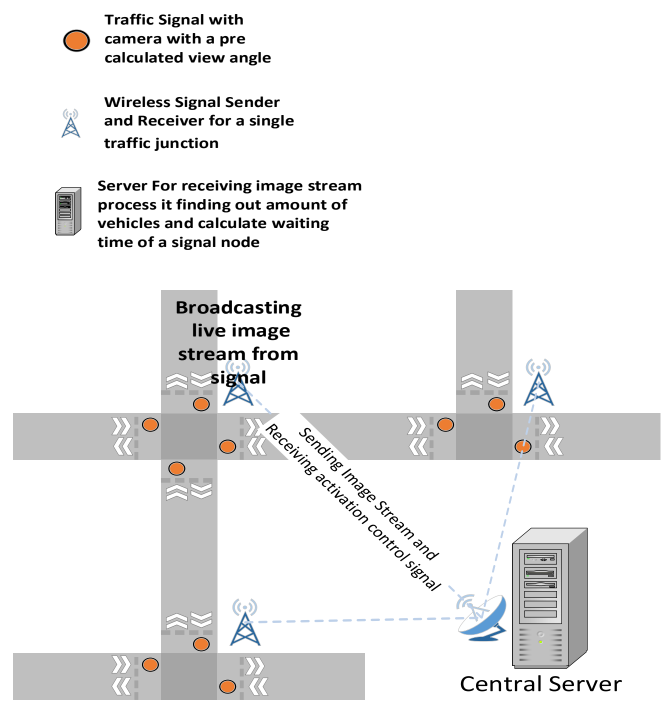

Fig. 5. System Design

On receiving image, server processes it and then computes number of vehicles and forwards to traffic management section of the software which computes optimal time for the traffic. Software sets clock for the specific node and send red light signal to controller device. Traffic controller device will set the light indicated by the server and holds its state until further instruction comes from the server. After the computed time elapse, server sends controller device to set green light. Fig. 5 describes the system, its components and its actions. All the traffic nodes are connected to a server. Server takes all the decision and commands the traffic nodes. Server has software to manage the system. Server device is connected to the network and communicates with the traffic nodes on TCP/IP port. In the server all the traffic nodes need to be registered and mapped in its memory. Server will identify each node uniquely by their MAC address. To instruct a specific node, server will use its MAC address to get the IP address from the data map and using that IP address server will communicate with the embedded device. When the system starts up, initially it commands all the connected nodes to light up red light. Upon calculating optimized waiting time and running time server can command specific node to perform specific task. In terms of scalability where many nodes are present in the traffic management system, a grid or cluster of server can be created to effectively manage the system. This feature has been integrated in the system with the help of developing tool of the system MATLAB.

V. Software Design

Two separate programs have been constructed to design the system. An embedded program has been written for the microcontroller. This software uses standard interfacing libraries to interface the components with the microcontroller. Embedded software opens a socket in TCP/IP port and creates a socket server which listens on the port for instruction. A separate program runs the server. Server software is the heart of the system. It is designed to process the images and calculated the optimal time for several junction at the same time. It can serve multiple nodes at the same time on different IP ports.

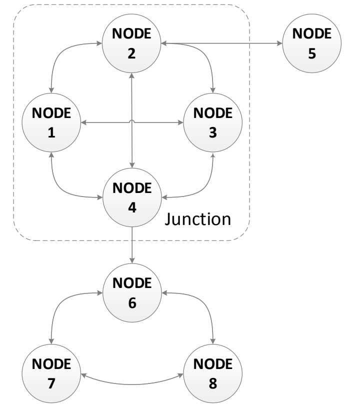

Fig. 6. Graph Representation of nodes

System is implemented with object oriented software design. Every traffic node is represented as an object in the software. Road to road connection is represented as directed edge. Three or four node combined together will form a junction.

For example, if there is a junction where 4 roads are connected then we can say that a single rode on the junction can go to 3 different road and it is represented by 3 different edge. So for 4 roads there are 12 directed edges. Every incoming node has at least 1 incoming edge and at least 1 outgoing edge. All the objects and edges are represented as a graph in the system. Fig. 3 describes the software representation of the system. Dashed Square in Fig. 6 represents a 4 road connecting junction.

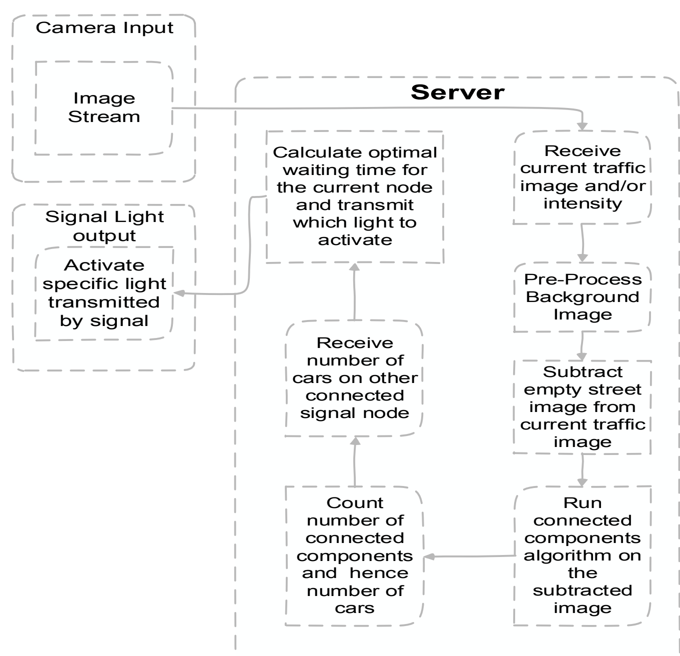

Fig. 7. Processing Software Diagram

The software has two sections. One Section is to retrieve image, process it, then derive vehicle count and the second section is optimized time computation section which is used to compute waiting time for each signal controller. These two run on different threads. Image of Empty Street is subtracted from the image from the input stream to reduce unnecessary information. Then we have used the pattern matching algorithm to identify all rectangle shaped patterns on the image. Rectangles between specific thresholds can be considered as vehicles. Then number of vehicles is counted and sent to the optimize time calculation section of the software to compute waiting time. Time calculation section seeks nearby node's waiting vehicles for a given node and makes a ratio of cars. Default time slice is multiplied with the ratio and assigns time for the given node. This calculation will be further researched for further optimizing.

Processing flow of the system is represented in Fig. 7. In the figure we can see the incoming image as input signal and each stage it passes through the system to acquire vehicle count to optimized time computation. System finally provide light activation signal as output.

VI. Operation of the System

In the previous section we have discussed about the components of our system. The total operation can be divided into three stages which are explained below.

A. *Fetching Images Data*

Each road connected to a traffic junction has its traffic controller(nodes). Each node has a camera to capture images and has access to wireless transmitter to broadcast images to the server. But to allow the server to command the embaded device a predefined instruction is needed. To achieve this purpuse, the host socket in node listens to requests and has a defined set of instruction as system commands. Node interprates http GET request "/getImage" as image capture command. When this request is received, Arduino controls the devices and captures images and sends that image back as a response to that request. Each node or traffic device has its own location information and MAC/IP address through which the server can identify the particular node. Now when the server requests an image to any node like, "http://xxx.xxx.xxx.xxx/getImage", the node will respond to the server. Here we are using TCP/IP protocol to send the image. Here "xxx.xxx.xxx.xxx" is the IP address of the node. The image is sent to the server via wireless connection from the wireless transmitter.

B. *Receiving and Analysis of the Image on the Sever*

This is the main part of our research work. We have explained briefly in the system design section how the image analysis process actually works. On the iterating process of calculating optimal time, server will request a specific node to provide street image for the time calculation of that specific node. At the end of the request, server will receive image of road and pass it to its computation section to process the image and compute optimal time. After computing optimal time, server will provide light duration command to node.

C. *Sending Signal to The Traffic Lights*

To capture images for allowing the server to light up a specific light on a specific node, a set of instructions are required. Node software interprets /red, /grn, /yhl request as light command and upon receiving one of these requests, software will light up the respective signal light. Here /red request turns on red light, /grn turns on green light and /yhl turns on yellow light. Initially server turns all the traffic lights to red. Now after getting the time period for a certain signal of a junction, the server sends signal to that particular node through TCP/IP connection. Server will command to turn the traffic light green from red like "http://xxx.xxx.xxx.xxx/grn" and server counts down the time period which has been calculated by the server. When the countdown is done server commands the traffic light to turn red again.

VII. Results and Discussion

This research is conducted in a replicated environment and in that environment significant evidence of time optimization was measured. For a fixed 180 seconds running time, four road cross junction takes around 720 seconds to complete a cycle. The dynamic system also takes 720 seconds to complete a cycle but waiting time of individual vehicle reduces according to the number of cars in each road of a junction.

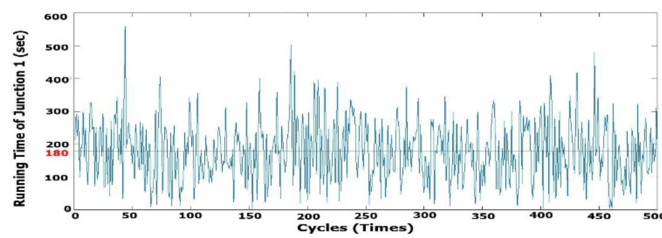

Fig. 8. Visualization of running time

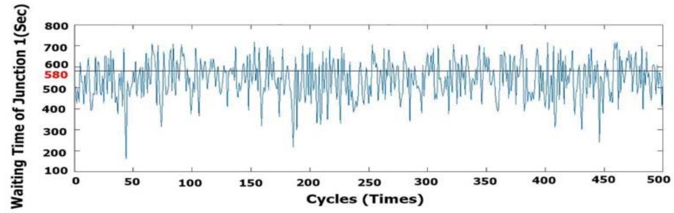

Fig. 9. Visualization of waiting time

Using randomly generated number of cars at the junctions, a simple simulation has been run in MATLAB. Here in the Fig. 8, the variation of running time for one junction has been shown. The horizontal straight line represents the constant time 180s which would be the running time for a node of a junction if the running time was same for all. But if the technique that has been proposed here is used, the running time won't be the same each time. There would be variation in time. Same logic goes for the waiting time, Fig. 9. Waiting time won't be the same all the time. But the black horizontal line represents the constant waiting time which is 540s for each cycle.

Table 1 represents the number of cars of the four roads of a junction which have been used as inputs and table 2 and table 3 represents the running time and waiting times of first 10 cycles.

TABLE I. NUMBER OF CARS

Road 1	Road 2	Road 3	Road 4
43	96	73	59
55	71	1	79
93	1	83	77
100	23	92	65
11	27	77	81
11	47	22	93
33	86	26	88
19	76	4	65
57	38	22	80
15	49	2	19

TABLE II. RUNNING TIMES

Running Time (Road1)	Running Time (Road2)	Running Time (Road3)	Running Time (Road4)
114.24	255.06	193.95	156.75
192.23	248.16	3.50	276.12
263.62	2.83	235.28	218.27
257.14	59.14	236.57	167.14
40.41	99.18	282.86	297.55
45.78	195.61	91.56	387.05
101.97	265.75	80.34	271.93
83.41	333.66	17.56	285.37
208.32	138.88	80.41	292.39
127.06	415.06	16.94	160.94

TABLE III. WAITING TIMES

Waiting Time (road1)	Waiting Time (road1)	Waiting Time (road1)	Waiting Time (road1)
605.76	464.94	526.05	563.25
527.77	471.84	716.50	443.88
456.38	717.17	484.72	501.73
462.86	660.86	483.43	552.86
679.59	620.82	437.14	422.45
674.22	524.39	628.44	332.95
618.03	454.25	639.66	448.07
636.59	386.34	702.44	434.63
511.68	581.12	639.59	427.61
592.94	304.94	703.06	559.06

VIII. FUTURE WORK

Currently the system uses fixed or predefined thresholds depending on the road to measure number of vehicles. System will be enhanced with machine learning abilities so that system itself can identify those thresholds. Currently the system runs on a single server for an individual network. It can be developed in such a way that system can run on a grid of servers to support multiple networks under a single system. Currently the components on the road run on external power. Devices on the traffic nodes can be powered by solar cell to increase power efficiency.

ACKNOWLEDGMENT

We would humbly like to thank North South University to provide us the environment to conduct the research. We would thank ACM archive to grant us access to their research works. We would like to acknowledge Shoraez Bin Shahjahan for reviewing literatures of our research. We would also thank “Tech Shop BD”, “Suzuki Electronics”, and “Tools BB” for providing the necessary equipment at our doorsteps.

REFERENCES

[1] S. Khan, 'The Financialexpress-bd', Thefinancialexpress-bd.com, 2015. [Online]. Available: http://www.thefinancialexpress-bd.com/2014/07/17/45710/print. [Accessed: 22- Nov- 2015].

[2] M. Claveria and K. Baek, "Analyzing Traffic Density in Images with Low Temporal and Spatial Resolution", *Proceedings of the 29th International Conference on Image and Vision Computing New Zealand - IVCNZ '14*, 2014.

[3] R. Cucchiara, M. Piccardi and P. Mello, 'Image Analysis and Rule-Based Reasoning for a Traffic Monitoring System', IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, vol. 1, no. 2, pp. 119-130, 2000.

[4] 'INTELLIGENT TRAFFIC LIGHTS CONTROL BY FUZZY LOGIC', Malaysian Journal of Computer Science, vol. 9, no. 2, pp. 29-35, 1996.

[5] M. Ferreira, R. Fernandes, H. Conceição, W. Viriyasitavat, and O. K. Tonguz, "Self-organized traffic control," *Proceedings of the seventh ACM international workshop on VehiculAr InterNETworking - VANET '10*, 2010.

[6] S. Djahel, M. Salehie, I. Tal and P. Jamshidi, "Adaptive traffic management for secure and efficient emergency services in smart cities", *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2013.

[7] L.-W. Chen, P. Sharma, and Y.-C. Tseng, "Eco-Sign," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, p. 438, Oct. 2011.

[8] R. Nakibuule, J. Ssenyange, and J. A. Quinn, "Low cost video-based traffic congestion monitoring using phones as sensors," Proceedings of the 3rd ACM Symposium on Computing for Development - ACM DEV '13, Jan. 2013.

[9] J. Harper, "Beijing to introduce congestion charge," in *The Telegraph*, Telegraph.co.uk, 2012. [Online]. Available: http://www.telegraph.co.uk/finance/personalfinance/expat-money/9511168/Beijing-to-introduce-congestion-charge.html. Accessed: May 30, 2016.