Intelligent Traffic Management System

Sumit Singh[1] Mr. Shivam Singh Patel[2]
[1]Student
[2]Assistant Professor
Maharishi University of Information Technology, Lucknow

Abstract— Traffic jams creates a wide variety of problems like;itwastesalotofprecioustimeofpeopletravellingfrom one place to another, also traffic may cause huge wastage of fuel and cause more pollution and this will result in an irreversible damage that will be caused to the environment. So, in this paper we solely focus on dispersing the traffic efficientlyandeffectivelybasedonthetypeoftraffic.

Key words: ITMS, Intersections, Lane Detection, Traffic Density Detection

I. Introduction

The main goal of this paper is to put forward a concept that will effectively manage traffic than the traditional technique used to manage or control the traffic.

Traditional technique of managing traffic involves either a traffic official going down to the site or controlling the movement of traffic at times of heavy traffic jam or by a central system that manually controls the traffic lights.

In the Intelligent Traffic Management System(ITMS), we will address the automation of changing the traffic signal timers based on the inputs provided by the IP cameras at the scenario of traffic. ITMS will be using certain techniques to detect the traffic conditions by using the traffic density detection technique, later on we will determine the number of lanes in the 'intersection'. Based on these inputs the ITMS can handle autonomously manage the traffic byitself.

Intersection is a word that would be used to describe a junction i.e. a place where multiple roads meet at a particular point and crosses eachother.

The thesis will be focusing on developing an algorithm to automatically manages each intersection, these intersections will be classified by the system depending on the density of the traffic that they are facing.

II. Motivation

The traditional traffic lights signaling system tends to work according to the timer that is embedded to the signals itself. It will continue to use the same time constraint even if the traffic density is high or low.

This results in wastage of time and ineffective use of the signaling system. In some situations, the traffic personnel would have to come down to the site in order to tame the intense traffic and this would make the traffic even more chaotic, often making the traffic movement in every direction uneven.

So, the ITMS will help us overcome such level of situations where the system will change the timer of every signals in a particular intersection according to the amount of traffic it will be facing. Each intersection will be categorized (which will be later on discussed) by the system depending on the basis of traffic density detection methodology.

ITMS will solely be focusing on the addressing the problem of automatically updating the system with traffic situations and take actions accordingly without the need of manual intervention.

III. Proposed Systemmodel

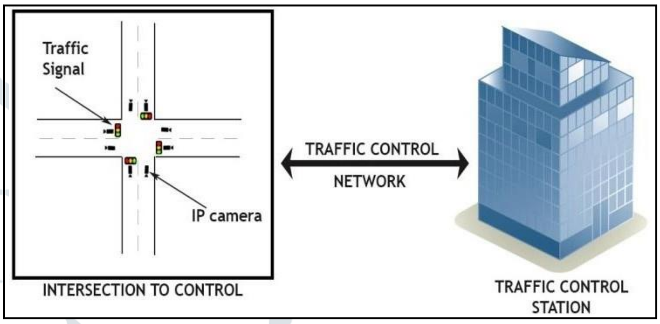

Fig. 1: Intelligent Traffic Management System (ITMS) Architecture / Model

The Fig. 1 shows that the ITMS is divided into three different layers; Intersection to Control (ITC), Traffic Control Network (TCN) and Traffic Control Station (TCS)

A. Intersection to Control (ITC)

The ITC is the setup that will be placed at each intersection of the road network, the can be implemented at a single intersection (junction) or at every intersection of the road network; this is because each ITC will be independent of the other no matter what the traffic conditions are.

The ITC layer would consist a set of signals and IP cameras placed at the intersection, ITC peripherals will be connected to a high speed network called the Traffic Control Network (TCN). The IP cameras will provide the system with real time video or imagery that will help the system for further processing and autonomous decision making.

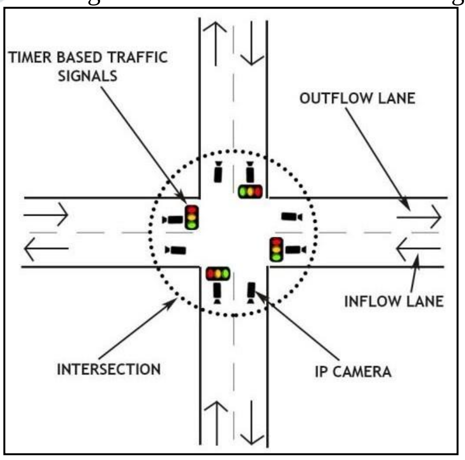

Fig. 2: Intersection to Control (ITC) layer architecture

In Fig. 2 of the ITC layer there are arrows that marks the flow of traffic through the intersection, the lanes by which the traffic exits are called the 'outflow lanes' and the lanes through which the traffic comes into the intersection is called the 'inflow lanes' as shown in the Fig.

2. Each inflow and outflow lanes are covered by the IP cameras that are used for various purpose later explained in the thesis.

B. Traffic Control Network(TCN)

The traffic control network (TCN) is the medium through which the system will receive the live traffic feed for further processing. TCN can either be a wired network with switches or may be a wireless network. The choice between the two types of network depends on the options that comes with it security, stability and scalability.

C. Traffic Control Station(TCS)

The traffic control station (TCS) layer would be making the decision for the ITC layer to act upon. The TCS will process the live feeds obtained from the ITC and based on the results the TCS will assign the new time intervals for each signals in a particularintersection.

TCS will be able to make the decisions freely and independently without the need for any human interventions. However, the system will be monitored by the traffic personnel also that whatever decision made by TCS system can be overridden by the personnel in case of emergencies.

We can use distributed computing architecture [1] for TCS which can be more efficient and effective but can be complex. The reason for choosing the distributed computing architecture for TCS layer is because the distributed computing architecture is fail-safe, because if one server or system is down then, the one that is operational can take over the operations of that server. Choosing distributed computing architecture for running TCS operations can be cost effective and also addresses the problem ofscalability.

IV. TRAFFIC FEED PROCESSINGTECHNIQUES

In this section we will focus on the techniques that will be used by the TCS layer in order to process the incoming traffic feed coming from the ITC layer. The TCS layer will be using these techniques of image processing to make a decision and make changes to the signal timers accordingly.

A. Lane Detection Technique

The number of lanes would be identified through the Edge Detection and Hough Transform methodology [2]. An IP Camera would either live stream or take a snap shot of the lane(s) to detected, then the system would split the media received into sub frames and the frames are kept in a frame buffer. Now, the image frames which is in RGB is converted to an intensity image and Y'CbCr image respectively. A 2-D filter is implemented by an image filter system object for detecting edges in the input stream. The system operates the defined segment to analyze the Hough transform. We also use Edge detection for tracing boundary of the lines. After computing the Hough transform, need to find the peaks of the Hough transform, the lane is marked in the image and thus by the number of lines we get the total number of lanes. Which is, after rendering the image, we get the total number of lines (L) that make up the lanes so, in order to find the actual number of lanes (I_{Lanes}) we use the following equation,

$I_{Lanes} = 1 - L$

The following is an example of lane detection technique by implementing the above method.

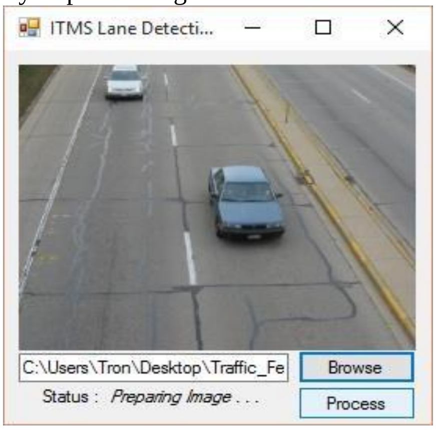

Fig. 3(a): Importing a two lane road image and process it using the lane detection technique

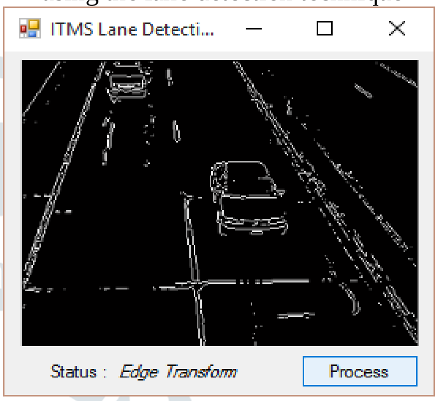

Fig. 3(b): Image after applying the edge transform

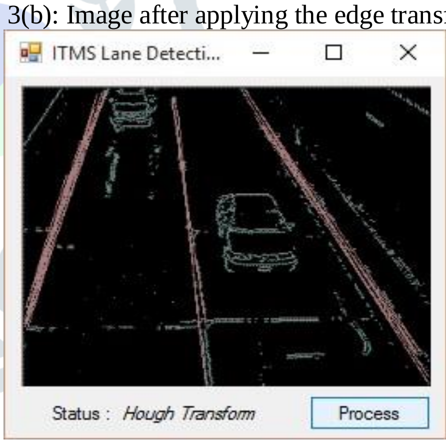

Fig. 3(c): Image after applying the Hough transform

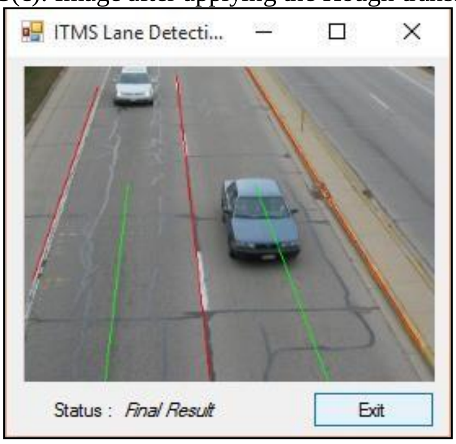

Fig. 3(d): Resulting image after applying the entire lane detection technique

In Fig 3.D we get a resulting image with three edges that is detected after the processing i.e. the number of lines (L) = 3. So we can apply the equation, I_Lanes= 1 – L; hence we get the number of lanes (I_Lanes) =2

B. Traffic Density DetectionTechnique

The RGB image or video stream from the IP Camera is used that helps detecting the number of lanes to find the density (number) of traffic (vehicles) [3] in the inflow (I_Inflow) region(s) of the intersection. Here the use of opencv comes into the scenario, opencv has a built image processing capabilities and also has a vast libraries of media processing scripts.

The RGB input image is converted into a grayscale image, after which the greyscale image is transformed into a binary (threshold) image for further processing where it is converted to canny image that is helpful in detecting the edges [4] of the objects (here in our case the vehicles will be our object), we then erode the canny image to find the edges.

Fig. 4(a): Applying the traffic density detection technique on the above raw image

Fig. 4(b): Resulting image returning the detected vehicles marked in green boxes

The system will then count the number of vehicles and assign it to a variable say, Inflow Vehicle Density (I_V_D) that would be checked against a Limited Vehicle Density (L_V_D) for traffic that would be predefined as a constant.

From Fig. 4(B) we can determine the variable (inflow vehicle density) I_V_D = 19, if the predefined value for L_V_D = 10.

So, here I_V_D > L_V_D which means the inflow lane of the intersection is facing traffic.

V. PROPOSED DECISION MAKINGMECHANISM

Here, we will discuss how to enable the system to make decisions on its own based on the results that it gets after processing the live traffic feed. For this the system needs to classify the intersection based on the traffic strength. The intersection needs to be defined under the three types reliever (RI), zombie (ZI) or clogged (CI) intersection. The reliever intersection is the section that is facing the least traffic or no traffic at all. Zombie intersection is the intersection that is slow moving and is facing the heaviest traffic, whereas the clogged intersection is one that needs to be given care so that it does not turn into another zombie intersection.

The system must implement the Intersection Status Detection Algorithm (ISDA), in order to identify whether an intersection is a reliever, clogged or a zombie intersection. The algorithm steps involve the following:

– Step 1: We need to detect the number of lanes (I_Lanes) by using the lane detection technique that was mentionedearlier.

– Step 2: Detect and record whether the lanes of the intersection are free of traffic or jammed using the IP camera and the traffic detection methodology explained earlier. We then assign the total number of lanes affected by traffic into a variable say, I_Lanes_Affected

– Step 3: Segregate the intersections based on the followingconditions:

1) Case 1 - If I_Lanes = I_Lanes_Affected, then mark the intersection as zombieintersection.
2) Case 2 - If I_Lanes_Affected > 2, then mark the lane as cloggedintersection.
3) Case 3 - If I_Lanes_Affected = 1, then mark the lane as relieverintersection.

ISDA can either be implemented at a regular interval of time or may a real time status updater for each intersection.

After marking every intersection with the current status of how the traffic is; we now need to assign separate logic or working mechanism for each type of intersections that is defined by the previous algorithm.

If the intersection is marked as a zombie intersection, we do not alter the timer settings of the signal and leave it as default until the intersection status gets changed to either clogged intersection or reliever intersection.

Now in order to manage the traffic faced by the clogged intersection we need to use the following algorithm or abbreviated as CIDA (clogged intersection disposer algorithm):

Given are the variables that would be used in CIDA algorithm; I_Delimiter & I_Status, where I_Delimiter is the limit for increasing and decreasing the traffic signal timer which actually acts as a threshold that will ensure that two-way flow of traffic is never stopped. The I_Status is used as the status checker of a particular intersection which can either be CI (Clogged Intersection), RI (Reliever Intersection) & ZI (Zombie Intersection).

- Step 1: Find the lanes in the intersection that are facing heavytraffic
- Step 2: Assign variable I_Delimiter =10
- Step 3: Increase the waiting (red signal) time & decrease the go (green signal) time of the lane(s) that are less affected by traffic by10%
- Step 4: Decrease the waiting (red signal) time & increase the go (green signal) time of the lane(s) that are affected by traffic by10%
- Step 5: Check the status of the intersection say, I_Status by using ISDA Algorithm described earlier and apply the rule with the matchingcondition:

1) Case 1 **-** If I_Status = CI go to step6
2) Case 2 **-** If I_Status = RI go to step7
3) Case 3 **-** If I_Status = ZI go to step7

- Step 6 **-** If I_Delimiter >80, then repeat the steps 3 and 4 also increment the variable I_Delimiter by10
- Else If I_Delimiter = 80, go to step5
- Step 7 **-** Reset the timer of the signals to regular intervals

The traffic management in the RI would be tricky because here we need to find out the outflow lane that will be helping the intersection to dispose of the trafficquickly.

Here we are going to address the different outflows and inflows of an intersection with the help of cardinal directions; the following are the abbreviations and their respective variables that are going to be used in the following algorithms:

North_Outflow_Vehicle_Density = N_O_V_D
South_Outflow_Vehicle_Density = S_O_V_D
East_Outflow_Vehicle_Density = E_O_V_D
West_Outflow_Vehicle_Density = W_O_V_D
Outflow_Vehicle_Density= O_V_D
Inflow_Vehicle_Density= I_V_D

There are separate IP Cameras that are specifically assigned to each of the outflow(I_Outflow) lane of an intersection. Based on the input of these IP cameras we will find the best outflow lane that is disposing of the traffic for a particular Inflow(I_Inflow) lane of the intersection, here we use the Best Outflow Lane Identifier(BOLI) Algorithm:

- Step 1: Check if I_V_D value is not null or not provided, if the value is null go to step 6 else proceed to step2
- Step 2: Use the traffic density detection method defined earlier and assign the values of the detected traffic density (Number of Vehicles) that is leaving the intersection via its outflow lane into the respective variables N_O_V_D, S_O_V_D, E_O_V_D, W_O_V_D
- Step 3: Then we need to check for the following conditions:

1) Case 1 **-** If N_O_V_D is greater than 55% of I_V_Ds value then declare N_O_V_D as the best disposer outflow lane go to step 4
2) Case 2 **-** If S_O_V_D is greater than 55% of I_V_Ds value then declare S_O_V_D as the best disposer outflow lane go to step 4
3) Case 3 **-** If E_O_V_D is greater than 55% of I_V_Ds value then declare E_O_V_D as the best disposer outflow lane go to step 4
4) Case 4 **-** If W_O_V_D is greater than 55% of I_V_Ds value then declare W_O_V_D as the best disposer outflow lane go to step 4
5) Case 5 **-** If none succeeds go to step5

- Step 4 **-** Reset the values for Outflow Vehicle Density Variables, set O_V_D to the best disposer outflowlane
- Step 5 **-** Stop

We need an algorithm that would handle the light weight traffic for the reliever intersection. The BOLI algorithm will be used inside the RIDA (reliever intersection disposer algorithm):

- Step 1: It is similar to the first step of CIDA i.e. we need to find out the lane that is facing the traffic coming from the cloggedintersection
- Step 2: Set I_Delimiter =10
- Step 3: Set I_V_D using the traffic density detection method
- Step 4: Now once that the I_V_D is set we then allow the first round of go (green) signal to be at regular timer (default duration) and also along with that we will run the BOLI algorithm to find which of the lane is best suitable for disposing the traffic efficiently. When BOLI algorithm runs check for theconditions:

1) Case 1 **-** If BOLI returns the best O_V_D then proceed to step5
2) Case 2 **-** If BOLI returns nothing or null value for O_V_D then go to step3

- Step 5 **-** Decrease the current wait (red signal) time and increase current go (green signal) time of the traffic moving from the particular 'inflow lane' to the 'best outflow lane' by10%.
- Step 6: Check for the followingconditions:

1) Case 1: If I_Delimiter < 80, find if the current inflow lane still faces traffic, increment I_Delimiter by 10 and go to step3
2) Case 2: If I_Delimiter = 80, find if the current inflow lane still faces traffic and go to step3

- Step 7: Stop and reset the timer of all the signals of that particular intersection todefault

VI. Conclusion

This paper describes various techniques used for detecting the lanes as well the traffic density faced by the lanes. These results are further used by the system to make a perfect decision.

Also the main goal of the system is to make use of the decision making algorithm to categorize each intersections and make suitable changes to the signals in an automated manner without the need for a human to intercept or govern the decision making process for clearing out the traffic.

VII. Futurework

The IP cameras of the outflow lanes can be used for detecting speeding vehicles and capture the license plate number. The ITMS will also have the ability to detect vehicle collisions and incidents that may lead to traffic, thus providing an alternate route using the Dijkstra's algorithm to find the shortest path.

ACKNOWLEDGMENT

We would like to convey our gratitude to friends and family who has been a wonderful support and helping us put together this thesis.

We would also like to show our gratitude to Prof. Shruti Pande for sharing her pearls of wisdom with us during the course of thisresearch.

Finally, we would like to thank all the authors mentioned in the references for sharing their knowledge which has been helpful for enlightening us in various prospects of technology.

REFERENCES

[1] Pradeep K. Sinha, "Distributed Operating Systems: Concepts and design", ISBN:978-81-203-1380-4 ©2007 New Delhi, India

[2] Thittaporn Ganokratanaa, Mahasak Ketcham and Sasipa Sathienpong, "Real time lane detection for driving system using image processing based on edge detection and Hough transform", ISBN:978-0-9891305-0-9 ©2013 SDIWC

[3] K.Vidhya, A.Bazila Banu, "Density Based Traffic Signal System", ISSN (Online) : 2319 -8753

[4] Guobo Xie and Wen Lu "Image Edge Detection Based On Opencv" doi:10.12720/ijeee.1.2.104-106