Road Traffic Monitoring by Intelligence-Driven Window based Image Analysis

Shivam Tyagi
B.Tech,
Department of Computer Science, College of Engineering Roorkee

Vikas Tripathi
Asst. Professor
Department of Computer Science, Graphic Era University, Dehradun

ABSTRACT

The term Traffic Analysis is the process to identify the different aspects of traffic and resolve the problems in concerning areas. These traffic problems may belong to traffic congestion and in order to remove this traffic congestion, determination of Road Traffic Volume or Road Traffic Identification is expected.

There are several techniques [6], [9], [11] of determining traffic volume on a specific road or highway such as Road Traffic Volume Detection by using LASER Sensors [11], Road Traffic Volume Detection by using inbuilt electromagnetic loops installed in roads and Road Traffic Volume Detection by using concepts of Digital Image Processing [3], [4], [5].

Now these techniques have their pro and cons. But by far the best technique, from the point of view of Expense and Results, is Digital Image Processing.

The goal of this paper is to analyze the density of road traffic by using the techniques and methods of Digital Image Processing and we are achieving this goal by using an intelligent window based system which has the capability to enhance its power according to the need of the system.

General Terms

Digital Image Processing, Feature Extraction, Object Recognition, Pattern Recognition, Road Monitoring.

Keywords

Intelligence-driven window, Frames Generation, Identification of Vehicles, Video Regeneration, Problem Description, Parallel Vehicle.

1. INTRODUCTION

1.1 Traffic & its Analysis

Traffic has its branches everywhere on roads in many forms. It may be in the form of vehicles, like cars, scooters, motorbikes, ridden animals, pedestrian and other conveyance. Basically it is the public way for purposes of travel.

Some places may have consistently, extremely large traffic volume, either during periods of time referred to as rush hour or perpetually. Exceptionally, traffic jam can be the result of many situations like an accident or an obstruction, such as construction. Such dynamics in relation to traffic congestion is known as traffic flow.

In measured traffic data, traffic congestion's common spatiotemporal empirical features have been found and on analysis it is found that for different highways in different countries, it is qualitatively same. Wide moving jam and synchronized flow phases of congested traffic can be distinguished by some of these common features in Kerner's three-phase traffic theory [1].

1.2 Digital Image Processing

The use of computer algorithms to perform image processing on digital images is Digital image processing. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. Digital Image Processing allows a much wider range of algorithms which can be applied on the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Due to two dimensional (perhaps more) attribute of an image, digital image processing may be modeled in the form of multidimensional systems.

In image processing, feature extraction has a great importance. Basically it is a special form of dimensionality reduction. When the input data to an algorithm is too large to be processed and it is suspected to be notoriously redundant (much data, but not much information) then the input data will be transformed into a reduced representation set of features (also named features vector). Now feature extraction is to transform the input data into the set of features. The features set will extract the relevant information from the input data if the features extracted are carefully chosen. This features set is required to perform the desired task using this reduced representation instead of the full size input.

The technique in this article focuses on the methods of digital image processing, computer vision algorithms and pattern recognition [10] to be applied to road traffic monitoring and analysis. Our main concern was to alter and modify these algorithms so that it can get fit into real-time road monitoring processes [9], and as a consequence the prototype system for traffic analysis was developed. Technically this system is based on stationary video cameras as well as computers connected to wide area network..

2. PROBLEM DESCRIPTION

Complex mathematical, algorithmic and programming problems were there when the techniques of image processing [7], [8] and object/pattern recognition [9] were applied on moving objects. Many articles have considered particular questions related: scene modeling, object geometry accounting, image contours processing. There is a lack of information on methods and algorithms used in digital monitoring technology, perhaps for commercial reasons.

Our major problem during the development of the algorithm was to handle multiple vehicles at once. The process of identifying the presence of multiple vehicles and their counting was a tricky and brain storming problem. Our second major problem was to monitor the vehicles under various atmospheric situations which led us to unwanted results in early stage of algorithm development which has been resolved later.

3. MODEL & SOLUTION

The problem of road monitoring as it is chosen in our research is presented as a sequence of independent processing steps intended to solve tasks logically connected to each other.

This intelligence-driven window based system works on the sequence of frames. Now the question is why this system is intelligent. Simple and straight forward explanation, because it works that way. Like in a situation if there are parallel vehicles running together then this system automatically detects the no of vehicles and accordingly it resize the window, here window is the part the frame where vehicle is being detected, means that portion of frame where our algorithm is working . These steps are in hand of the following order of algorithmic processing:

- Video stream input to computer (personal computer or specialized one),
- Its conversion to a sequence of single frames,
- Noise filtering, linking and labeling,
- Moving vehicle tracking,
- Video regeneration.

Such a sequence of steps is determined by the order of logical stages. For that first of all initial data have to be given in the form of video sequence, and then process each frame within tracking lines to locate vehicle and gives it a count when it passed away .One type of data is used in the system, related to the location of stationary camera: motion scenes may be filmed with a view from above to the road surface that are suitable for vehicle tracking.

Notice that in the system there are no sensors used or electromagnetic loops installed in the roadbed to detect moving vehicles.
In our algorithm, we used some of the special libraries of JAVA like xuggle, util, awt etc.. Some of the java's advance libraries of image processing may have been used but we didn't use them because of the easy and more precise implementation of the libraries that we've used.

Video sources, that we used to define and justify our algorithm, are taken directly from google and various research papers on the particular subject that led us to more comparable and accurate results.

The whole process of implementing algorithm can be defined in following steps:

3.1 Frames Generation

To implement the algorithm first step required is to convert the video into collection of sequential frames. Theses frames are then processed for the rest of the procedure.

The frames generation process is carried out according to the frame rate of the video and generates equivalent frames. As if any the frame rate of any video is 25 frames/sec and the video is of length of 2 minutes. Then equivalent frames for the specific video.

*No. of frames = Frame Rate * Video Length (in sec.)*

Thus for the above example it will generate total 25X120=3000 frames and then these frames are processed.

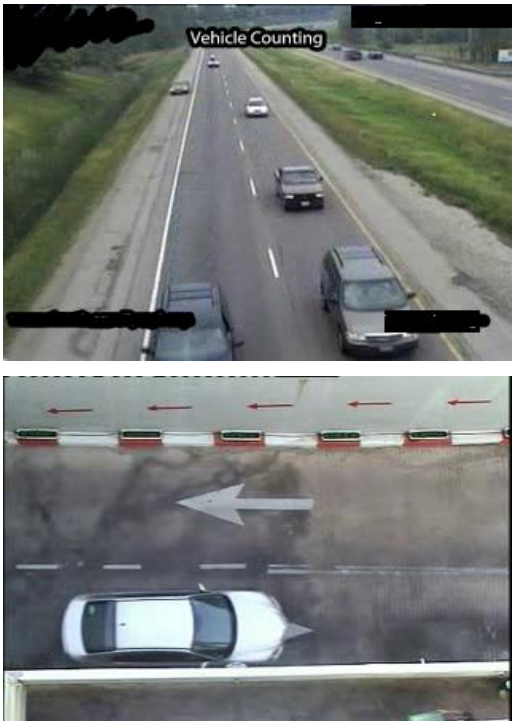


Fig 1. Frames generation using Multiple Example Videos

Pseudo Code for Frames Generation:

Input: Example Video which has to be processed and its frame rate.

Output: Frames generated from video.

```
1. Take video as an input file
2. Define the frame rate of the video
3. Invoke the MediaWriter and start generating the frames
IMediaReader mediaReader = ToolFactory.makeReader(inputFile);
4. Invoke the image snap listener and convert the images into BGR 24 bits color space system
mediaReader.setBufferedImageTypeToGenerate(BufferedImage.TYPE_3BYTE_BGR);
5. Store in generated frames into HDD
ImageIO.write(image, "png", new File(outputFilename));
6. End of the process.
```

3.2 Identification of Vehicles

The process of vehicle identification is carried out at specific pixels location that is called the window. Whenever any vehicle enters in this region, the process of vehicle counting starts and this process is based on our intelligence-driven system. This process is carried out by the comparison of pixel intensity at two specific pixel positions in that particular window and the comparison is viewed by the two lines at these pixel positions in the perpendicular direction of the motion of the vehicle.

The comparison is done by using a particular frame as a reference frame and comparing these points value of reference frame to all the proceeding frames' points that generate accurate results.

Selection of these two points can be set manually at one time according to the direction of the traffic and camera video capturing size and quality. But this is not fixed and can be altered later for another type of traffic flow and identification process, thus provide us the flexibility and power.

Basically there are two scenarios of the selection of the points:

- Points in Horizontal Direction (Along X-Axis)
- Points in Vertical Direction (Along Y-Axis)

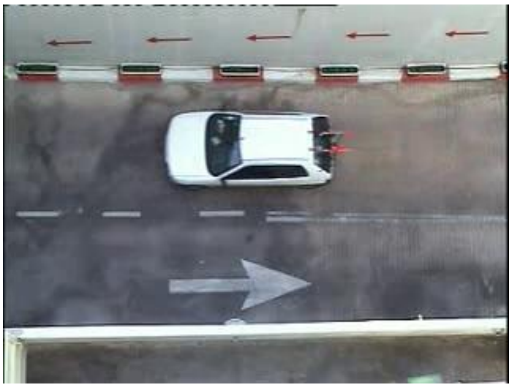


Fig 2. Points in Horizontal Direction

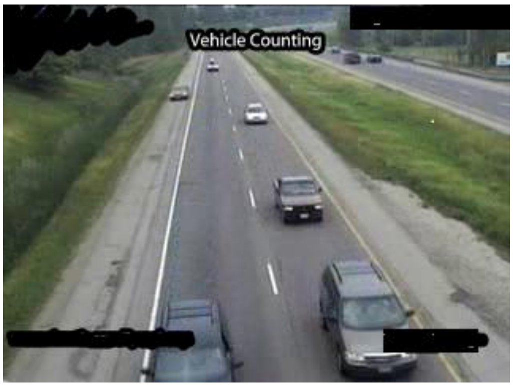


Fig 3. Points in Vertical Direction

The first scenario is used when the direction of the motion is along X-axis and the second scenario is used when the direction of the motion is along Y-axis. If the direction of the traffic motion is not particular in any of the above direction then any of the scenarios can be used according to the alignment of the traffic motion towards the specific direction.

Now after the fixation of the points system starts stamping the vehicle whenever it hits both of the Base Line coordinates and when the vehicle leaves any of the Base Line, system increases the count of the vehicle by one. It also detects the direction of the motion by the same way.

In fig 4, vehicle is just going to hit the first Base Line coordinate, means at this position system doesn't know anything about the vehicle. Whenever the vehicle hits the Base Line coordinates (fig 5), system knows that something has entered the zone but till this time also it doesn't start counting. But when the vehicle hits both of the Base Line coordinates, system starts detecting the vehicle (fig 6). Now to continue its motion vehicle should be moving in its direction that means sooner or later it will leave both of the Base Lines but as the vehicle leaves any one Base Line out of two, system increase the count of the vehicle by one and the counting procedure is done (fig 7). Here figures are edited for the better understanding of the concept concerning Base Line. In actual system of detecting vehicle these Base Lines are only visible when any vehicle hits any of the Base Lines; otherwise these Base Lines are completely invisible.

Base Lines are introduced in this system for the better understanding of the algorithm. From the algorithm point of view these lines are just for the presentation, they don't play any role in the detection of the vehicle.

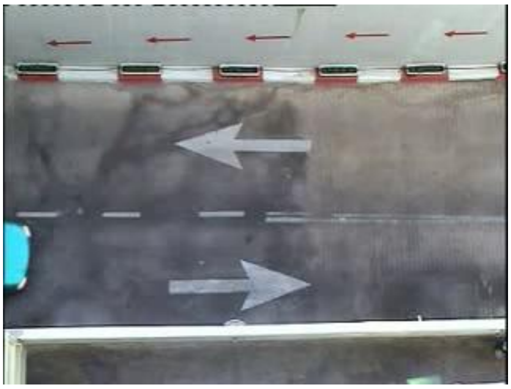


Fig 4. Vehicle Before Hitting the first Base Location

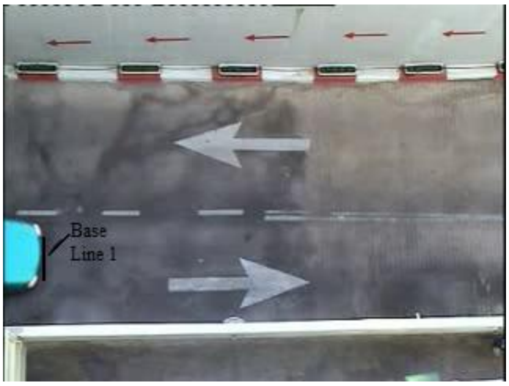


Fig 5. Vehicle after Hitting the First Base Location

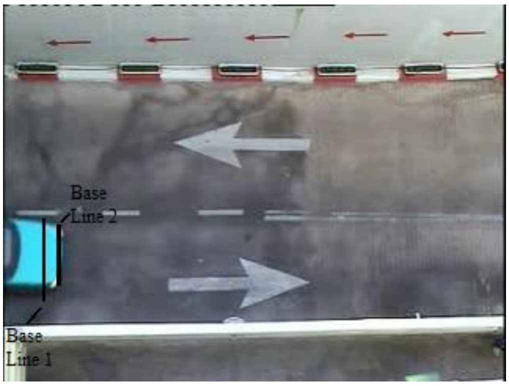


Fig 6. Vehicle after hitting both of the Base location

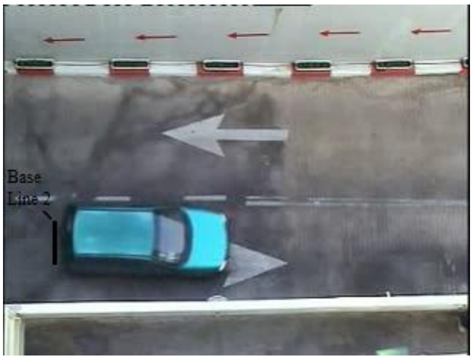


Fig 7. Vehicle after leaving one of the Base Locations and count by 1

Parallel vehicles can also be recognized by the system due to its intelligence power. For this, algorithm uses a specific calculation, based upon the width of the frame and if system finds the relation incorrect it breaks the whole scenario into two systems that is what we call intelligent system. Now these two systems will work in the same manner by using two windows but count the vehicle separately thus providing exact results.

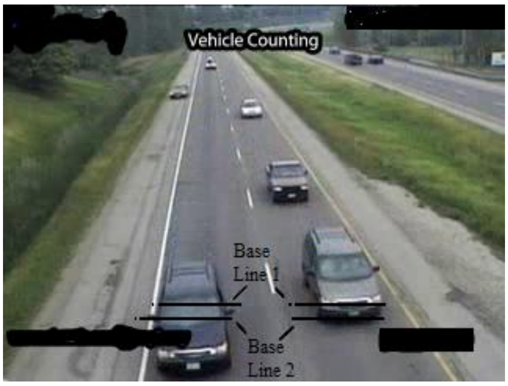


Fig 8. Detecting two Parallel vehicles

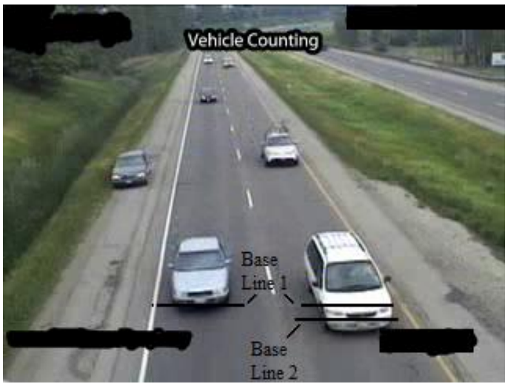


Fig 9. Start detecting a vehicle and second one is just arrived

Figures are again represented by using the presentation logic of Base Lines. In these figures there are atmost two vehicles coming together parallelly but the algorithm is developed in a way that it works fine if the number of vehicle increases.

Algorithm works the same way as stated before, the only difference is that it just breaks the whole window into two then it the counts the vehicles (fig 8,9,10,11).

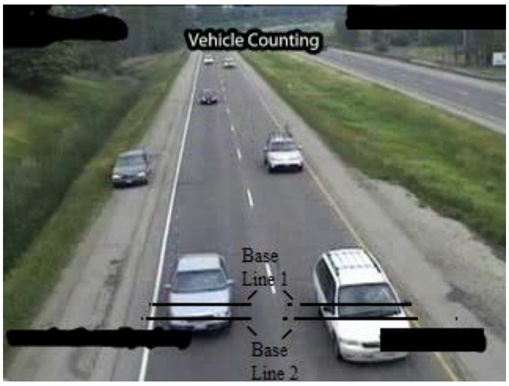


Fig 10. Both vehicle are in detection process

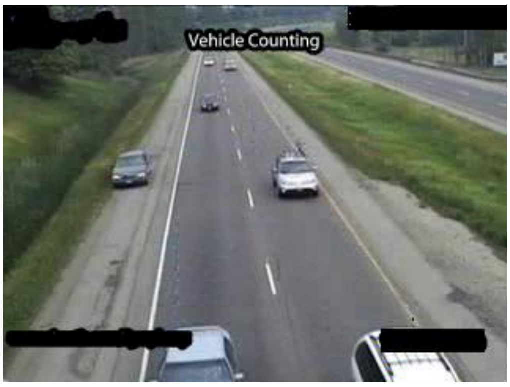
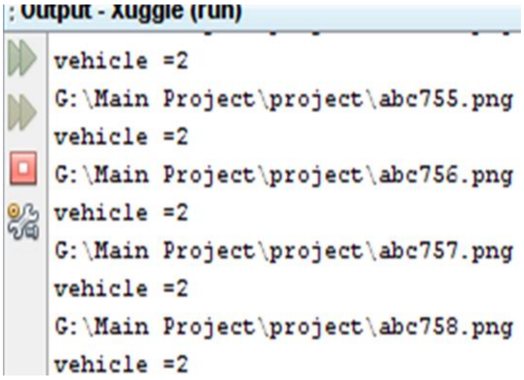
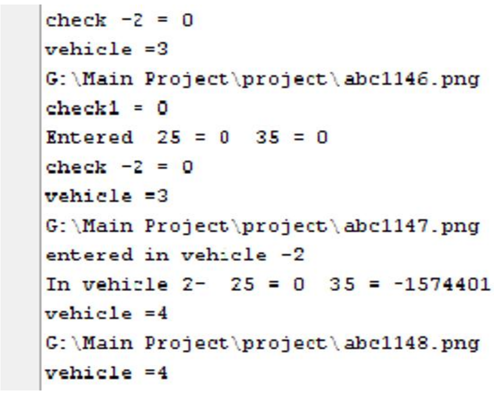


Fig 11.   Vehicles are detected and count is increased by two



```
vehicle =2
G:\Main Project\project\abc755.png
vehicle =2
G:\Main Project\project\abc756.png
vehicle =2
G:\Main Project\project\abc757.png
vehicle =2
G:\Main Project\project\abc758.png
vehicle =2
```

Fig 11. Vehicle Counting Window



```
check -2 = 0
vehicle =3
G:\Main Project\project\abc1146.png
check1 = 0
Entered  25 = 0  35 = 0
check -2 = 0
vehicle =3
G:\Main Project\project\abc1147.png
entered in veh:cle -2
In vehi:le 2-  25 = 0  35 = -1574401
vehicle =4
G:\Main Project\project\abc1148.png
vehicle =4
```

Fig 12. Vehicle Counting Window: No of vehicles changed

Fig 11 and fig 12 shows the counting result during the detection process. An interesting fact is here that we used a seperate window for the result and didn't include the result into the figures as it gives us the capability of knowing the direction of motion,time at which vehicle hits the locations and the ease of working.

After the identification and counting of vehicle next step that comes into the light is Video Regeneration.

Pseudo code for Vehicle Counting:

```
Input: Reference Frames which are to be processed.
Output: Modified Frames and the counter which sets the no of vehicles.
1. Enter the reference frame for once as input.
BufferedImage bufferedImage = ImageIO.read(inputFile);
2. Store the pixels' intensity of the reference frame inside an array.
3. Now select the input frames stream for vehicle counting.
4. Choose from horizontal or vertical system according to the motion of the traffic.
5. If horizontal system is chosen, reference points will be in the direction of X-axis otherwise it'll be in the direction of Y-Axis.
6. Whenever the vehicle reaches the reference points, compare its pixel values to the reference pixel values and the detection of the vehicle starts.
For horizontal System:
if((k==25 || k==35) && !(rgbs[i]/100000<=(ar[k][j]/100000)+15 && rgbs[i]/100000>=(ar[k][j]/100000)-15 ));
For vertical System:
if((i==190 || j==120) && !(rgbs[i]/100000<=(ar[k][j]/100000)+15 && rgbs[i]/100000>=(ar[k][j]/100000)-15 ));
Where i,j,k are loop variables ,rgbs is the buffer image array & ar is reference array.
7. Check for the Parallel System, if condition is found true, call parallel system function vCount().
  if(la-f>(.3*w)){
        if(la1-f-20>20)
          vCount(f,la1-20,rgbs,0,1);
        if(la-la1-20>20)
          vCount(la1+20,la,rgbs,0,2);}
Where la,la1,f are pixel edge pixel values of buffer image
8. Count the vehicle using vehicle counting function vCount() function.
public void vCount(int a,int b,int c[],int d,int m)
 Condition inside vcount() that check for vehicle occurrence:
 if(((rgbs[f+(diff/2)]!=0 && rgbs[f+(diff/2)+10*w]==0) || (rgbs[f+(diff/2)]==0 && rgbs[f+(diff/2)+10*w]!=0)) && ((check1_1==0 && check_1==0) || (check1_2==0 && check_2==0)))
9. Rewrite the buffer image into the HDD replacing its previous form.
        ImageIO.write(image, "png", new File(ImageName+".png"));
10. End of the process.
```

3.3 Video Regeneration

After the frame modification and vehicle counting video regeneration is necessary for the manual checking of the system which provides better system check-up for errors.

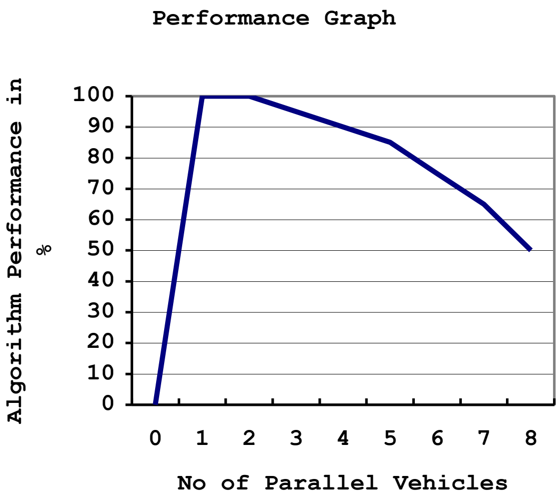
Pseudo code for Video Regeneration:

```
Input: Frame Rate of the video, stream of processed frames.
Output: Final video.
1. Enter the frame rate of the video.
2. Enter the stream of processed frames a input.
3. It generates the output video and stores it into the HDD which is ready to play.
   writer.encodeVideo(0,bgrScreen, System.nanoTime() - startTime,
          TimeUnit.NANOSECONDS);
4. End of the process.
```

4. RESULTS

It is clear from the above specification that the algorithm is working fine and producing exact result for both the single lane and parallel system. All the results are shown in figures included in section 3.2.

A graph may be constructed based on the performance of the algorithm. This performance of algorithm is based on the increasing no of parallel vehicles.



4.1 Comparison

4.1.1 *Comparison with Optical Flow Algorithm*

- If we talk about the optical flow algorithm, then it is kind of dependent on prediction that how the vehicle is changing its position within the frames, so if any vehicle suddenly changes its speed or direction then it'll produce wrong results while our algorithm doesn't get affected by speed of the vehicle because it always consider and start counting the vehicle when it reaches the detection points.

• Sometimes it is seen in optical flow algorithm that if two vehicles are coming very near to each other then it treats them as a single vehicle while our algorithm works intelligently on this scenario also.

• If two vehicles are crossing each other, system may get confuse and produces arbitrary results while our system is totally full proof of that due to multiple window system.

• If two vehicles are moving in opposite direction then it cannot detect the direction exactly while our algorithm does it perfectly.

4.1.2 Comparison with Other Techniques

Till the time of development of this algorithm, there are so many techniques to count the no of vehicle like laser sensors, inbuilt electromagnetic loops. So if we compare our algorithm, which uses the technique of digital image processing, this is the best developed algorithm yet by using the best technique available i.e. digital image processing from the frame of expense and results. Hardware requirements and processing time are considerably less and enhanced capability of the algorithm gives it a boost to lead the others.

5. CONCLUSION

Thus our algorithm is by far the best algorithm for vehicle counting problem and produces good results in many challenging situation such as vehicle coherence, multilane vehicle motion, parallel vehicles motion, vehicles in opposite direction, vehicle in parking. The qualities or the advantages of the algorithm can be understood as it doesn't get affected by the direction of the motion of the vehicle, works fine with single lane, multilane and parallel system, no other code required for the detection of the motion of the vehicle, works faster as we using JAVA thus platform independent and can work on any JAVA enabled system and definitely its easily modifiable and can be altered according to the requirement of the system.

6. FUTURE WORK

The algorithm is working accurate in vehicle counting problem in many contexts as stated above. Now its capability can be enhanced in other areas of Road Traffic Analysis also such as vehicle speed recognition, vehicle type recognition (means is the vehicle four wheeler or two wheeler, if four wheeler then is it a car or a truck, or a bus, or some other type of vehicle), vehicle motion path. We are working in the positive direction to achieve our goals and very sooner we'll be able to put all these enhancement together to provide much more powerful and capable algorithm.

7. REFERENCES

[1] Boris S Kerner, Sergey L Klenov and Dietrich E Wolf. Journal of Physics A: Mathematical and General, Volume 35, Number 47, 2002. Cellular automata approach to three-phase traffic theory.

[2] Martin Treibe, Arne Kestinga, Dirk Helbingb. Transportation Research Part B: Methodological Volume 44, Issues 8–9, September–November 2010. Three-phase traffic theory and two-phase models with a fundamental diagram in the light of empirical stylized facts.

[3] Antonio Fernández-Caballero, Francisco J. Gómez, Juan López-López. Expert Systems with Applications, Volume 35, Issue 3, October 2008. Road-traffic monitoring by knowledge-driven static and dynamic image analysis.

[4] Baik Hoh, Marco Gruteser, Ryan Herring, Jeff Ban, Daniel Work , Juan-Carlos Herrera, Alexandre M. Bayen, Murali Annavaram, Quinn Jacobson. MobiSys 2008, 6th international conference on Mobile systems, applications, and services. Virtual trip lines for distributed privacy-preserving traffic monitoring.

[5] Nicolas Hautiere, Erwan Bigorgne, Jérémie Bossu, Didier Aubert. The Eighth International Workshop on Visual Surveillance - VS2008. Meteorological Conditions Processing for Vision-based Traffic Monitoring.

[6] Vargas, Toral, Barrero, Milla. Intelligent Transportation Systems(ITSC), Oct. 2008. An Enhanced Background Estimation Algorithm for Vehicle Detection in Urban Traffic Video.

[7] Jutaek Oh, Joonyoung Min, Myungseob Kim, Hanseon Cho. Transportation Research Record: Journal of the Transportation Research Board, January 2010. Development of an Automatic Traffic Conflict Detection System Based on Image Tracking Technology

[8] Robert Bain. Transportation, Volume 36, Number 5, February 2009. Error and optimism bias in toll road traffic forecasts.

[9] Prashanth Mohan, Venkata N. Padmanabhan, Ramachandran Ramjee. SenSys 2008, 6th ACM conference on Embedded network sensor systems. Rich monitoring of road and traffic conditions using mobile smartphones.

[10] Yegor Malinovskiy, Yao-Jan Wu, Yinhai Wang. Transportation Research Record: Journal of the Transportation Research Board, December 2009. Video-Based Vehicle Detection and Tracking Using Spatiotemporal Maps.

[11] Shehata, Jun Cai, Badawy,Burr, Pervez, Johannesson,Radmanesh. Intelligent Transportation Systems, June 2008. Video-Based Automatic Incident Detection for Smart Roads: The Outdoor Environmental Challenges Regarding False Alarms.