Imperial Journal of Interdisciplinary Research (IJIR)
Vol-2, Issue-6, 2016
ISSN: 2454-1362, http://www.onlinejournal.in

Density Based Traffic Control System Using Video Processing (Hardware and Software Implementation)

Prof. D. D. Pukale[1], Palak Chauhan[2], Adhikari Siddhi Satish[3], Preeti Nawal[4] & Neha Kumari[5]
[1]Associate Professor, [2,3,4,5] Student
Department of Computer engineering, BVCOEW, Pune

***Abstract**: This paper presents the development of Density based traffic system using video processing with opencv. Real time vehicle density is calculated using blob algorithm from live video feed. Detect emergency vehicle using opencv library template matching and give priority to emergency vehicle, report violation of traffic rule.*

1. Introduction

Road traffic congestion is a recurring problem worldwide. This is primarily because infrastructure growth is slow compared to growth in number of vehicles, due to space and cost constraints [1].

To eliminate the delay on roads by reducing traffic and controlling the signals automatically using video processing. Precisely the system also keeps watch for violation of rules. Match for emergency vehicle and Give priority if detect any emergency vehicle.

2. Related Work

"Controlling Traffic Light Signals to Implement Traffic Scheduling Using Density Calculation", Fahreena Shaikh, Naisha taban khan, Saima Zareen Ansari
An intelligent traffic light controlling system with a enhance method of vehicle density calculation and dynamic traffic signal time manipulation along with emergency vehicle alert[1].

"Computer Vision based vehicles Detection and Counting for Four Way Traffic"
Meru A.V, Mujawar I. I
Background subtraction method gives us better and accurate result for moving vehicles detection. After the vehicles detection counting is accurately done using the Gaussian mixture model and BLOB analysis method[7].

"Traffic Jam Detection Using Image Processing"
Prof. Uma Nagaraj, Jitendra Rathod,Prachi Patil,Sayali Thakur,Utsav Sharma
Image processing is a better technique to control traffic jam. It is also more consistent in detecting vehicle presence because it uses actual traffic frames [5].

3. Proposed System

3.1. System Architecture

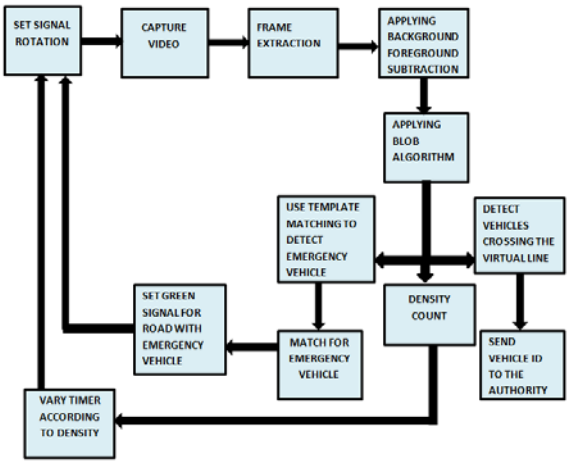


Figure 1 System Architecture

3.1.1. Overview of The System

- Capture the video and extract frames.
- Calculate vehicle density.
- Match for the rule violation and send the message.
- Detect priority vehicle and give green signal.
- Circular Switch based on density.

3.2. Methodology

3.2.1. Video Acquisition

Video is captured through cameras mounted on Signal polls. This video is further converted into frames.

```
#region cameracapture
  if (comboBox1.Text == "Capture Fr
om Camera")
    { try
        {
       _capture = null;
       _capture = new Capture(0);

  _capture.SetCaptureProperty(Emgu.
CV.CvEnum.CAP_PROP.CV_CAP_PROP_FPS,
 30);
 _capture.SetCaptureProperty(Emgu.C
V.CvEnum.CAP_PROP.CV_CAP_PROP_FRAME
_HEIGHT, 240);
 _capture.SetCaptureProperty(Emgu.C
V.CvEnum.CAP_PROP.CV_CAP_PROP_FRAME
_WIDTH, 320);
#endregion cameracapture
```

3.2.2. Frame Extraction

The following function processes the frame. Frame gets converted into byte array. That byte array gets converted into hex values of each frame. Then those hex values are stored into the array list for further process. Extract the current frame from video capture, be it device or video file.

```
Frame=_capture.QueryFrame();

Framesno=_capture1.GetCaptureProper
ty(Emgu.CV.CvEnum.CAP_PROP.CV_CAP_P
ROP_POS_FRAMES;
```

Frame is converted into bitmap and assigned to picture box to display.

```
pictureBox1.Image=frame.ToBitmap();
```

3.2.3. Background Detection

Background subtraction, also known as Foreground Detection, is a technique in the fields of image processing. Generally an image's regions of interest are objects (humans, cars, text etc.) in its foreground.

```
privateIBGFGDetector<Bgr>_forground
Detector;
if (_forgroundDetector == null)
  {
   _forgroundDetector      =      new
FGDetector<Bgr>
(Emgu.CV.CvEnum.FORGROUND_DETECTOR_
TYPE.FGD_SIMPLE);
   }

_forgroundDetector.Update(frame);
```

3.2.4. Blob Tracker

```
private static BlobTrackerAuto<Bgr>
_tracker;
_tracker=new
BlobTrackerAuto<Bgr>();

foreach(MCvBlob blob in _tracker)
{
 int blob_id = (blob.ID + 1);

framecopy.Draw(blob_id.ToString(),
ref_font, Point.Round(blob.Center),
new Bgr(Color.Red));
     if (Point.Round(blob.Center).X
   > rectangleShape1.Location.X)
     {
    if (!rectCnt.Contains(blob_id))

   rectCnt.Add(blob_id);
       }
    }
```

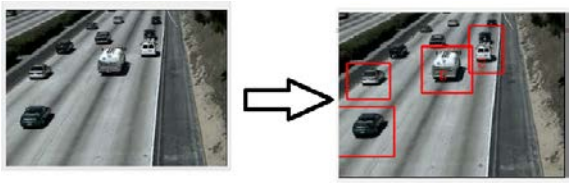


Figure 2 Blob Count

3.2.5. Template Matching

Template matching is a technique in digital image processing for finding small parts of an image which match a template image. Our project uses template-matching method to recognize emergency vehicle.

```
Public int
templateMatchAmbulance(Image<Bgr,
byte> source{
```

```
Emgu.CV.CvEnum.TM_TYPE.CV_TM_CCOEFF
_NORMED))}
```

Figure 3 Emergency Vehicle Detection

3.2.6. Flood Fill

Flood fill algorithm paints some area by replacing a specific interior color instead of searching for a particular boundary color[9].

```
Gray white = new Gray(255);
Gray black = new Gray(0);

  if (contour.Area > 200)
resultImage.Draw(contour.BoundingRe
ctangle, white, -1);
   else
resultImage.Draw(contour.BoundingRe
ctangle, black, -1);
```

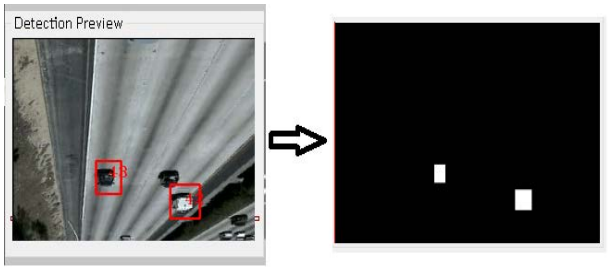


Figure 4 Flood Fill

3.2.6. Signal Switching Algorithm

Table 1 Threshold for Switching

Timer	Threshold
60	Density count<=10
90	Density count<=20
120	Density count<=30

Step 1: Initialize the signal to 1.

Step 2: Count the vehicle density at each lane and store it in a variable

Vehicle_count[i] = _tracker.Count

Step 3: If Vehicle count is greater than calculated threshold then give more green delay else normal delay will be assigned.

```
if (one >= threshold)
    textBox2.Text = "" + 90;
    else
     textBox2.Text = "" + 60;
```

Step 4: Glow the green led of that particular lane indicated by the signal variable and rest of the lanes will be assigned red light delay.

```
Green1.Visible = true;
Red2.Visible = true;
Red3.Visible = true;
Red4.Visible = true;
```

Step 5: After the completion of current timer Increment the signal value by 1

And do switching in circular manner.

Step 6: If signal is equal to 5 go to step 1.

4. Hardware implantation

4.1. Arduino board

Arduino is microcontroller board designs, produced by several vendors, using various microcontrollers. These systems provide sets of digital and analog I/O pins that can interface to various expansion boards and other circuits. The Arduino project provides an IDE based on a programming language named processing, which also supports the languages C and C++.

4.2. Printed Circuit Board

A printed circuit board (PCB) mechanically supports and electrically connects electronic components using conductive tracks, pads and other features etched from copper sheets laminated onto a non-conductive substrate.

4.2.1. PCB circuit diagram

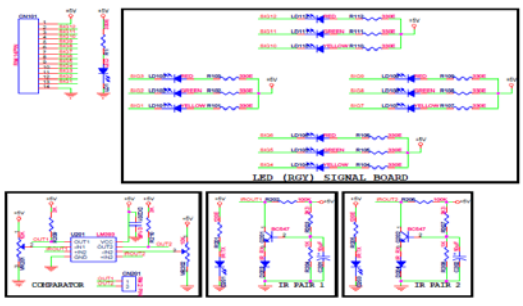


Figure 5 PCB Circuit Diagram

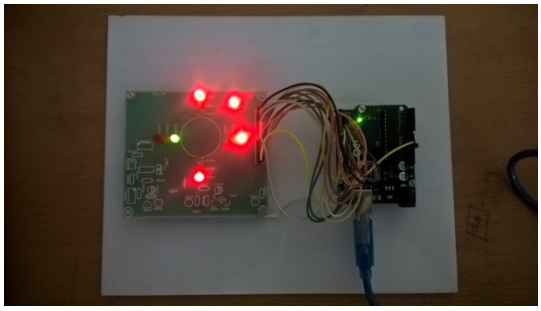
5. Result And Analysis

5.1. Simulation

Result of simulation is shown below. In our proposed system we use visual studio 2012 for showing the simulation.

Figure 6 Simulation

5.2. Hardware Result



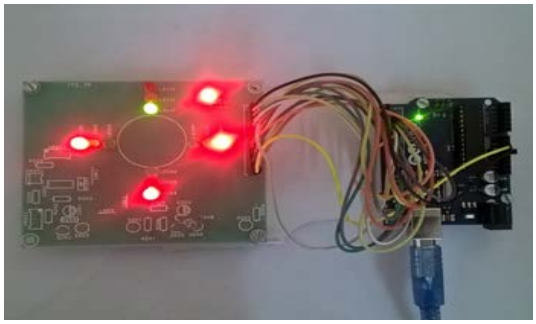


Figure 7 Hardware Result

5.3. Analysis

Table 2 Blob Analysis

Actual blob count	Expected blob count
22	19
36	32
47	41
19	16

Table 3 Timer Allocation Analysis

Density count(number)	Timer(seconds)	Threshold
8	60	Density count<=10
15	75	Density count<=20
21	90	Density count<=30
35	120	Density count<=40

6. Conclusions

We conclude that video processing is a better technique for calculation of traffic density and controlling the state change of traffic light also use of opencv library for video processing is good tool as software. Our proposed system reduce the traffic congestion by managing traffic light efficiently and save the time wasted by the green light on empty road. Our system gives priority for emergency vehicle if detect using template matching. The reporting of rule violation if any vehicle crosses virtual line. Overall, the system is good but it still needs improvement to achieve a hundred percent accuracy.

7. Acknowledgements

We would like to thank our head of the department and our guide Prof.D.D.Pukale for motivating us in doing such kind of real time projects and thanks to our entire family member and the web sources.

8. References

[1]. Fahreena Shaikh, Naisha taban khan, Saima Zareen Ansari,"*Controlling Traffic Light Signals to Implement Traffic Scheduling Using Density Calculation*" (ICAET15TR012000) , February 2015

[2] Sabya sanchi Kanojia "*Real time light control and congestion avoid system*"(IJERA) , March-April 2012

[3].Payal Gupta, Dhananjay V. Gadre, Tarun Kumar Rawat "*Real Time Traffic Light Control System*" (Hardware and Software Implementation)(ijeeev7n5_10) , 2014

[4].R Nithin Goutham, J.Sharon Roza, M Santhosh "*Intelligent Traffic Signal Control System*"(International Journal of Advanced Research

in Electrical, Electronics and Instrumentation Engineering) ,May 2014

[5] Prof. Uma Nagaraj, Jitendra Rathod,Prachi Patil,Sayali Thakur,Utsav Sharma*"Traffic Jam Detection Using Image Processing*".(IJERA) ,March-April 2013

[6] G Lloyd Singh, M Melbern Parthido, R Sudha "*Embedded based Implementation: Controlling of Real Time Traffic Light using Image Processing"*((NCACSA 2012) Proceedings published in International Journal of Computer Applications® (IJCA)) ,2012

[7] Meru A.V, MUJAWAR I. I " *Computer Vision based vehicles Detection and Counting for Four Way Traffic*"(International journal of Advanced Research in Computer Science and Software Engineering)(IJARCSE) ,February 2015

[8] George Law"*Quantitative Comparison of Flood Fill and Modified Flood Fill Algorithms*"(International Journal of Computer Theory and Engineering,) , June 2013

[9] Prof Mrs Priti Tyagi, Mr Sandeep Oberoi, Mr Satish Sonawane, Tejal Rane and Mahip Shrawag *"Control of Real Time Traffic with the Help of Image Processing"* IRET Transaction on Image Processing (ITIP), April 2014

[10] Naeem Abbas Sir Syed ,Muhammad Tayyab Sir Syed ,M.Tahir Qadri Sir Syed *Real Time Traffic Density Count using Image Processing* (International Journal of Computer Applications) ,December 2013