Vehicle Detection Mechanism for Traffic Control using Image Processing: Software Implementation

Parichita Basak[1] Sarita Yadav[2] Ghausia Gulrukh[3] Ramandeep Kaur[4]
[1,3,4]Student [2]Assistant Professor
[1,2,3,4]Department of Information Technology
[1,2,3,4]Bharati Vidyapeeth's College of Engineering New Delhi, India

Abstract— Population is increasing day by day exponentially. Number of vehicles on road is also increasing which is resulting to high traffic. In India, traffic management is done statically i.e. time delays between switching of red, green and amber lights are fixed irrespective of the density of traffic i.e. number of vehicles on the lane. So, red light ON for a lane less density traffic is same for another lane with high density traffic. This technique is very ineffective in its functionality. So, our work focuses on implementing a dynamic based traffic control using image processing in which time delays between switching of traffic lights is dependent on the number of vehicles. In our proposed system, number of vehicles on a lane is inversely proportional to duration of red light ON. This paper describes the use of image processing as it is more efficient than other techniques which include use of sensors, microprocessors and microcontrollers which is very expensive.

Key words: Traffic Lights, Density, Image Processing, Traffic Management

I. INTRODUCTION

Traffic congestion is a serious problem in today's era. Various techniques are available in our country to control traffic but none of them are efficient because of certain reasons. Manual controlling of traffic by police at signals requires high manual efforts and time which is quite ineffective[1]. Another methods which includes use of sensors[2], microcontrollers and microprocessors are very expensive to implement and need high maintenance as well.

Our main aim is to design a proposed system for dynamic management of traffic using image processing which is more effective in terms of time and quality constraints. It is less expensive than methods requiring sensors and controllers. Image processing is more flexible, scalable and reliable. Our proposed system is implemented using OpenCV library with Visual Studio which supports various functionalities of image processing. Various methods or algorithms can be implemented for blob detection using OpenCV libraries like edge detection and morphological operations.

Edge detection[3] algorithm deals with extracting of boundaries of objects to measure the density of pixels. Various edge detection techniques are Perwitt, Sobel and Canny but output obtained by using edge detection techniques are very noisy and not efficient.

Morphology[4] is an analytic technique in the field of image processing which plays a vital role in noise reduction as compared to edge detection and improve the image quality after processing with the help of operators dilation, erosion, opening and closing. It is basically developed for binary images and now its functionality has been extended to grayscale images and functions. Binary morphology is implemented to probe(extract) an image which is simple and pre-defined in shape and processed further on how this shape hits and miss the shapes in the image. Grayscale morphology is implemented by treating images as functions by mapping them into an Euclidean space.

II. RELATED WORK

In [5] traffic management system using Background Detection algorithm is implemented which extracts the region of interest from a picture i.e. our blob and controls the switching between traffic lights as per the blob density.

In [6] a system is implemented where vehicles are provided with automatic brake control which will minimize the accidents and damage to the entities, a buzzer which alerts if a traffic rule is broken and a GSM/GPS tracker which helps us in case a vehicle is stolen.

In [7] a system is proposed using image processing with raspberry pi module to control the delays between traffic lights based on the vehicles density. A camera source is being configured at the light posts to capture images and further analyzing it for traffic management.

In [8] Canny edge detection algorithm is implemented to reduce noise and efficiently detect the true pixels to calculate the blob density and control the traffic light time delays.

In [9] a review is being done on controlling traffic lights in different weather conditions using image processing.

In [10] prototype of traffic control system of a city is built to show its functionality using video image processing.

In [11] a smart traffic control system is proposed using image processing which captures images and switch traffic lights according to blobs density. Also, it incorporates a RFID tag which ensures law enforcement and alerts if someone breaks a traffic rule.

III. LITERATURE SURVEY

A. Image Processing

Process of converting RGB to grayscale images and performing some mathematical operations for extracting some useful information is known as image processing[12]. It takes an image as input, various operations are performed on it and we get the required information as output in the form of an image. In this paper, we propose a system where we implement morphological operations[13] along with convex hull algorithm which will detect the blobs, that is, vehicles and help in decreasing traffic congestion to a great extent.

B. Background Subtraction

Background subtraction[14] is also called Foreground Detection. Using this technique in image processing, we extract the region of interest from an image, say cars, humans etc. In our proposed system, our region of interest are cars, so we will extract the vehicles on the road using background subtraction algorithm of image processing. And, further processing can be done on the extracted blobs to get the required result.

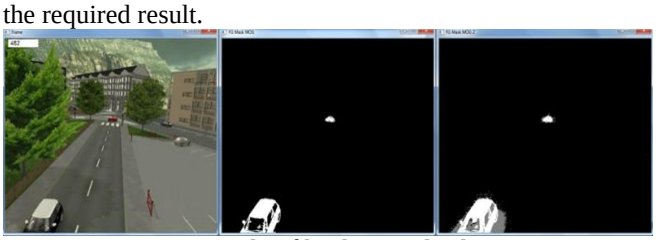

Fig. 1: Example of background subtraction

C. Motion Based and Frame Differencing

Process of finding the moving object by subtracting two consequent frames in image series from the background image is known as Frame Differencing[15] method. Approaches used are motion segmentation, visual based dimensional approximation, multimodal temporal panaroma which are quite complex and costly to implement.

Fig. 2: Example of Motion Detection (very noisy)

D. Edge Detection

Process of finding outlines or boundaries of blobs within images is known as edge detection. Image boundaries are the points where there is an abrupt change in the brightness level. These points have the discontinuities. Edge detection [16] is used in the field of machine vision, computer vision and image processing for feature extraction and feature detection. Various edge detection techniques are Canny edge detector, Sobel edge detector, Prewitt edge detector, Thresholding and linking.

Parameters	Operataors		
	Sobel	Prewitt	Canny
Edge points	Image intensity gradient is maximum	Prewitt approximation used, pixels that are closer to the centre of the masks are given no emphasis	Local maxima of gradient of image
Noise	More susceptible	More susceptible	Less susceptible
Threshold	One threshold value used	One threshold value used	Two threshold values used which will determine strong and weak edges

Table 1: Edge detection techniques[17]

To identify change in intensity Gradient based approach is used which uses the derivative operator.

Fig. 3: Example of edge detection (noisy)

E. Morphological Operations

Processing of images is done on the basis of shapes using morphological[18] operations. Two basic operators are dilation and erosion[19] in morphology. Morphological algorithm provides efficient results in terms of flexibility, accuracy and speed. Foreground shrinking is erosion, foreground expanding is dilation. Other morphological operations include "closing" which means removing holes in the foreground, "opening" which means removing stray foreground pixels in background, finding outline and skeleton of the background.

1) Dilation[20]

ORing is performed between the structuring element and the original binary image. Structuring element controls the extent of "thickening" and is represented by 0s and 1s matrix.

Dilation equation: $A \oplus B \{x|B_{x} \cap A \neq \phi\}$ (1)

"Φ is the empty set and B is the structuring element"

2) Erosion[20]

ANDing is performed between the structuring element and the original image. Structuring element controls the extent of "shrinking" and is represented by 0s and 1s matrix.

Erosion equation: $A \ominus B \{x|B_{x} \cap A^{c} \neq \phi\}$ (2)

"ϕ is the empty set and B is the structuring element"

0s and 1s in the structuring element are the hit and miss transform.

F. Applications

- Texture Decomposition
- Forensics
- Blob Detection
- Retina Recognition
- Fingerprint Recognition
- Pattern Recognition

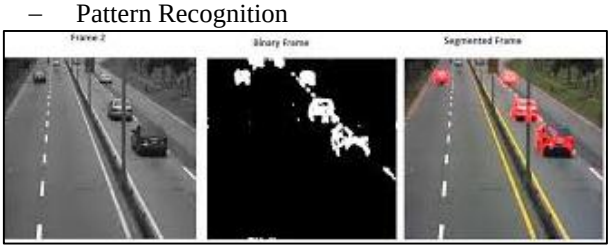

Fig. 4: Example of Morphological operation (good image quality)

IV. OUR PROPOSED SYSTEM

Our system is implemented using Morphological Operations in Visual Studio using OpenCv which supports wide range of libraries for performing image processing operations.

A. Why Morphology?

As discussed in Section II, we used morphological algorithm for implementing our density based traffic control system as it is more effective than other techniques like Motion Based and Frame Differencing, Background Subtraction and Edge Detection.

Steps involved are

1) Image Acquisition

This is the first step in which image is browsed and is loaded into the picture box. Image is converted into Bitmap format before loading as all the image processing operations are performed more effectively in Bitmap classes.

```
OpenFileDialog open = new OpenFileDialog();
        open.Filter = "Image Files(*.jpg; *.jpeg; *.gif;
*.bmp)|*.jpg; *.jpeg; *.gif; *.bmp";
        if (open.ShowDialog() == DialogResult.OK)
        {
                pictureBox1.Image = new
Bitmap(open.FileName);
                path = open.FileName;
        }
        toolStripStatusLabel1.Text = "inverting image....";
        System.Drawing.Bitmap aq =
(Bitmap)pictureBox1.Image;
        Invert a = new Invert();
        aq = a.Apply(aq);
```

Same process is followed to load pictures in four respective picture box.

2) Apply Morphological Algorithm for Blob Detection

This is the second step where morphological operations are performed using libraries of OpenCV to detect blob density and display the blob count.

```
AForge.Imaging.Image.FormatImage(ref aq);
        IFilter filter =
Grayscale.CommonAlgorithms.BT709;
        aq = filter.Apply(aq);
        Threshold th = new Threshold(220);
        aq = th.Apply(aq);
        bl = new BlobCounter(aq);
        int i = bl.ObjectsCount;
        ExtractBiggestBlob fil2 = new
ExtractBiggestBlob();
        fil2.Apply(aq);
        label2.Text = Convert.ToString(i);
```

Output is shown as:

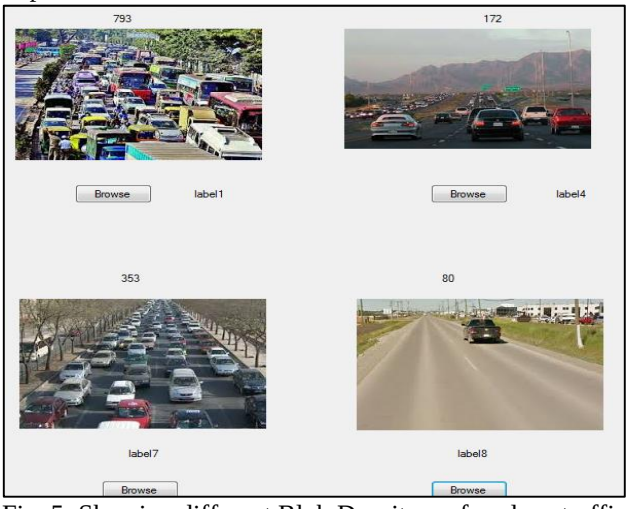

Fig. 5: Showing different Blob Density on four lane traffic

3) Compare densities and control red light duration

In our proposed system, this is the final step where image is compared on the basis of blob density on four lane road and duration of red light is updated according to number of vehicles. More the density less will be the duration of the red light.

Output is shown as :

Fig. 6: Timer control after comparing blob densities

V. CONCLUSION

In this paper, we have compared various methods (Background Subtraction, Motion Based and Frame Differencing, Edge Detection and Morphology) for object detection and we have concluded that morphological operations are less susceptible to noise than all other blob detection algorithms. Morphological operations are more efficient in terms of detecting edges, improving signal to noise ratio and noise susceptibility whereas other techniques inaccurate in terms of timing and quality constraints. We have implemented our system using Morphological algorithms using OpenCV libraries and controlled the timer of red light by comparing traffic densities of four lane road.

REFERENCES

[1] Velastin, S. A., et al. "Automated measurement of crowd density and motion using image processing." Seventh International Conference On Road Traffic Monitoring And Control, 26-28 April 1994. Proceedings. (IEE Conference Publication 391). 1994.

[2] Sharma, Pramod, Akanksha Mishra, and Kaumudi Singh. "Density Based Intelligent Traffic Control System Using IR Sensors." International Journal of Scientific Research 4.5 (2016).

[3] Panda, Santwana, and Anjali M. Patki. "Edge Detection Based Traffic Management System with Route Updates on Android Application." transportation 4.6 (2016).

[4] Gowda, Chandan S., et al. "Intelligent Traffic Signal Monitoring and Management System." Imperial Journal of Interdisciplinary Research 2.6 (2016).

[5] Pukale, D. D., et al. "Density Based Traffic Control System Using Video Processing (Hardware and Software Implementation)." Imperial Journal of Interdisciplinary Research 2.6 (2016).

[6] Badiger, Anil, et al. "SysteMatic and automatic road traffic junction." Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on. IEEE, 2016.

[7] Francis, Sony, and Sunitha Beevi. "Intelligent Traffic Control using Raspberry PI." (2016).
[8] Chauhan, Himanshu, Sandhya Tarar, and P. G. Scholar. "Image Processing Edge Detection Technique using Iterative Enhancement Wavelet used for Traffic Control Problem." International Journal of Engineering Science 4973 (2016).
[9] Ayar Priyanka, M., and Laxmi Saraswat. "International Journal of Advance Research in Engineering, Science & Technology." Traffic 3.5 (2016).
[10] Cao, Jiandong. "Research on Urban Intelligent Traffic Monitoring System Based on Video Image Processing." International Journal of Signal Processing, Image Processing and Pattern Recognition 9.6 (2016): 393-406.
[11] I slam, Md Rokebul, et al. "An efficient algorithm for detecting traffic congestion and a framework for smart traffic control system." 2016 18th International Conference on Advanced Communication Technology (ICACT). IEEE, 2016.
[12] Russ, J,C. The image processing handbook. CRC press, 2016 Jan 5.
[13] Burger, Wilhelm, and M. J. Burge. Digital image processing: an algorithmic introduction using Java. Springer, 2016.
[14] Jadhav, Prashant, "Smart Traffic Control System Using Image Processing." ,2016.
[15] R.A. Hadi, G.Sulong and L.E. George, "Signal Vehicle Detection and Tracking Techniques: A Concise Review", An International Journal(SIPIJ), Vol. 5, Feb 2014, pp. 5-6.
[16] Sharma, A., "Multi-level threshold based edge detector using logical operations.", 2016.
[17] S.K.Katiyar, P.V. Arun, "Comparative analysis of common edge detection techniques in context of object extraction", IEEE TGRS Vol. 50, pp. 68-75.
[18] Koay, J. Ming, et al. "Parallel Implementation of Morphological Operations on Binary Images Using CUDA." Advances in Machine Learning and Signal Processing. Springer International Publishing, 2016, pp. 163-173.
[19] Mardiris, V. Chatzis. "A Configurable Design for Morphological Erosion and Dilation Operations in Image Processing using Quantum-dot Cellular Automata." Journal of Engineering Science and Technology Review 9.1 (2016), pp. 25-30.
[20] S.Das, "Comparison of various edge detection technique", International Journal of Signal Processing, Image Processing and Pattern Recognition Vol. 9, 2016, pp. 152-156.

