VEHICLE DETECTION USING IMAGE

PROCESSING TECHNIQUES

**Varun Buddha**

19BCE1168

**Arjun PJ**

19BCE1116

**MA Ram**

19BCE1162

# Abstract

*Nowadays, detection of the vehicles and their*

*classification is very essential and also it has a lot of importance because of its use in many applications. One of the main applications is controlling and managing traffic. Vehicle*

*detection and tracking plays a major role in the project of preventing road accidents using image processing. Tracking of moving objects is important in monitoring surveillance videos and capturing human motion. Taking its importance into consideration, an efficient algorithm is proposed to detect vehicles in an image using image processing. The image is captured from the front view of the vehicles. So, this algorithm detects vehicles using the front view. Each vehicle is detected based on its size. The two major techniques used in this algorithm are edge detection and morphological processing.*

*Edge detection as well as morphological processing are important applications in image processing because of their wide range of uses. Edge detection is used to enhance the*

*objects in image. Morphological operations are used to remove noise and as well as to adjust image in such a way to detect objects in an image. The simulation work of this algorithm is done in OpenCv which is a very strong scientific tool.*

# Introduction

Today road fatalities and traffic management are the two

major issues faced by many citizens all over the country

especially in urban areas. According to the survey, in past

few years around 4 lakh accidents happened each year. And 40-50 percent of the unnatural deaths are contributed by these road accidents. Among these accidents, metropolitan cities has major contribution. The important fact is that 90 percent of the accidents are done due to lack of traffic monitoring . Over speeding vehicles by youth, rash driving, drunk and drive, red light jumping, using phone while driving are some of the important reasons which causes accidents due to driver’s carelessness. These accidents had ruined many lives and sometimes government property is also damaged. So, this is the major challenge for the people

in India especially the people who are leading urban life. To

overcome these challenges, many methodologies have been

proposed using emerging technologies . For most of the

methods, the first and foremost important step is to detect

vehicles and their tracking. So, vehicle detection has a lot of

importance in today’s life. With the help of vehicle detection, vehicles can be tracked time to time which avoids all the problem mentioned above . Vehicle detection helps in video surveillance application . In past few years, many algorithms have been developed for vehicle detection and tracking. To overcome all these drawbacks, an efficient algorithm is designed to detect vehicles in an image using image processing. As Image processing is a vast domain which helps us to process the image as per our convenience and requirement by using various applications of this domain.

# Problem Statement

Due to the heavy population in the cities as well as villages traffic problems like heavy traffic jams, violation of traffic rules, improper traffic management and traffic congestion increased enormously that results in long waiting times, loss of life, wastage of fuel and money, accidents,etc.,. Vehicle detection and counting is very important in traffic congestion, keep tracking of vehicles and to control the traffic signal duration. Enormous growth of population results in the improper management of traffic and thus it results in th lack of control of traffic by human power.As a result fines are imposed on those who viloate traffic rules but became invain in some cases. In order to reduce the traffic problem, we need the smart solution and it can be possible through “SMATRACON- Smart Traffic Control Using Image Processing”

# Literature Survey:

**Image Processing:**

Image processing is a method to perform some operations on an image, to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image.

Image Processing basically needs following three steps:

1. Importing the image via image acquisition tools.
2. Analysing and manipulating the range.
3. Report that is based on image analysis.

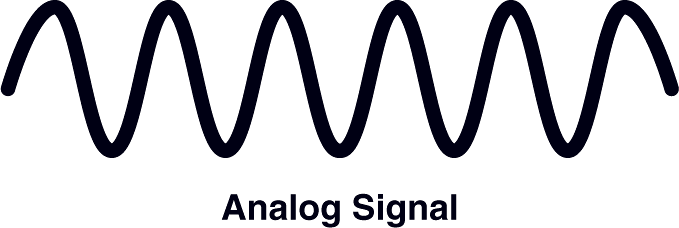
Image processing refers to quantitative analyses and/or algorithms applied to digital image data. It allows generation of 3D parametric maps and implies calculation of values that should be ultimately replicable and rather-independent. Generally, 2 types of image processing are used.

They are:

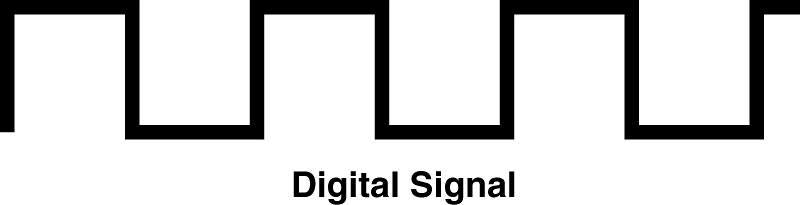
1. ANALOGUE IMAGE PROCESSING
2. DIGITAL IMAGE PROCESSING

# Analogue Image Processing:

Analog image processing is any image processing task conducted on two-dimensional (2-D) analog signals and if the pictorial representation of the data represented in analog wave formats that can be named as analog image. Mainly, analogue image processing can be used for the hard copies like printouts and photographs.



# Digital Image processing:

Digital image processing deals with manipulation of digital images through a digital computer. It is a subfield of signals and systems but focus particularly on images. The input of that system is a digital image and the system process that image using efficient algorithms and gives an image as an output by using computers. The three general phases that all types of data undergo while using digital technique are pre-processing, enhancement, and display, information extraction.

# OpenCV-Python:

Python is a general-purpose programming language started by Guido van Rossum, which became very popular in short time mainly because of its simplicity and code readability. Compared to other languages like C/C++, Python is slower but can be easily extended with C/C++. This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives us two advantages: ﬁrst, our code is as fast as original C/C++ code (since it is the actual C++ code working in background) and second, it is very easy to code in Python.



Besides that, several other libraries like SciPy, Matplotlib which supports Numpy can be used with this. So OpenCV-Python is an appropriate tool for fast prototyping of computer vision problems. OpenCV-Python works with python wrapper around original C++ implementation. All the OpenCV array structures are converted to-and-from Numpy arrays. So whatever operations you can do in Numpy, you can combine it with OpenCV.

**OBJECTIVES**

**1.**To develop the system which can be efficient using artificial intelligence, such that the concept image processing technique is implemented for estimation of actual road traffic in the particular area by counting the number of vehicles in the traffic up to the particular distance

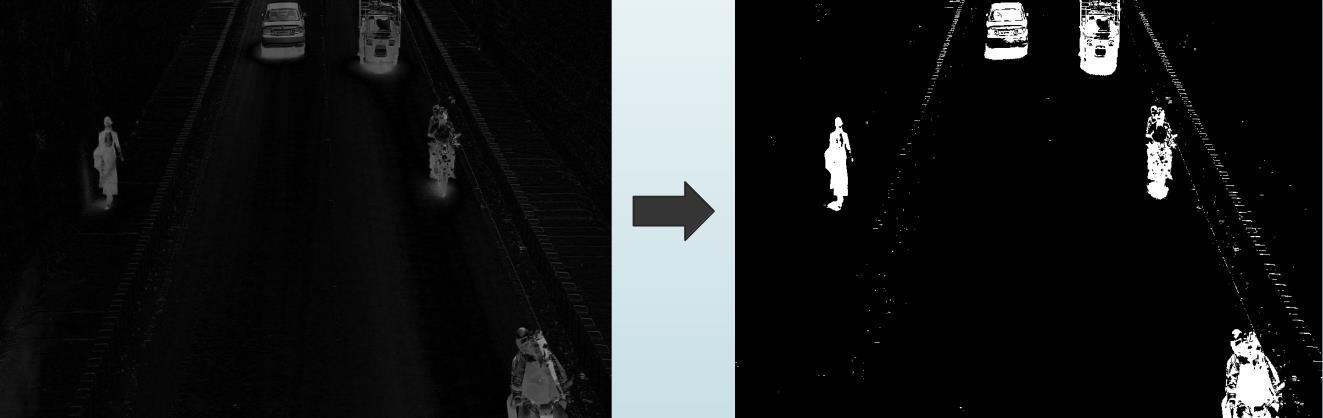
1. To monitor the system in such a way that the count of vehicles is displayed in digital way
2. Developing the server to handle previous data in automatic way.
3. To develop a vision-based surveillance system capable of identifying vehicles in the scene.
4. To track the vehicles as they progress along the image sequence.
5. To count the number of vehicles passing.

# IMPLEMENTATION:

**Background Registration:**

Background registration is the most important job in vision-based surveillance systems. The background is created by taking the continuous average of accumulated weight of first few frames and is gray scaled.

# Background Registration algorithm:

1. procedure get background (alpha , no of frames)
2. result = empty frame
3. for 1 : no of frames do
4. Retrieve new frame from the camera
5. result = ( 1 - alpha )result + ( alpha )new frame
6. end for
7. result = grayscale(result)
8. return result
9. end procedure

**Frame Extraction and Grayscaling:**

The video to be processed in nothing but a series of images or frames. Each frame is accessed individually and processed. OpenCV provides us with methods to extract individual frames with ease. Before processing the frame, we convert it to grayscale. Converting image to grayscale is a simple task in OpenCV. It provides a function” cvtColor()” for the same.It uses the following formula.

**Y = 0.299R + 0.587G + 0.114B**

**Background Subtraction:**

Background subtraction is done to obtain the objects in the fore-ground i.e for detection of vehicles. Here we take the absolute difference between theregistered background and the current frame. Since both the images are grayscaled it isvery easy to calculate the difference and for further processing. OpenCV provides us witha function called “absdiff()” to carry

out the process.



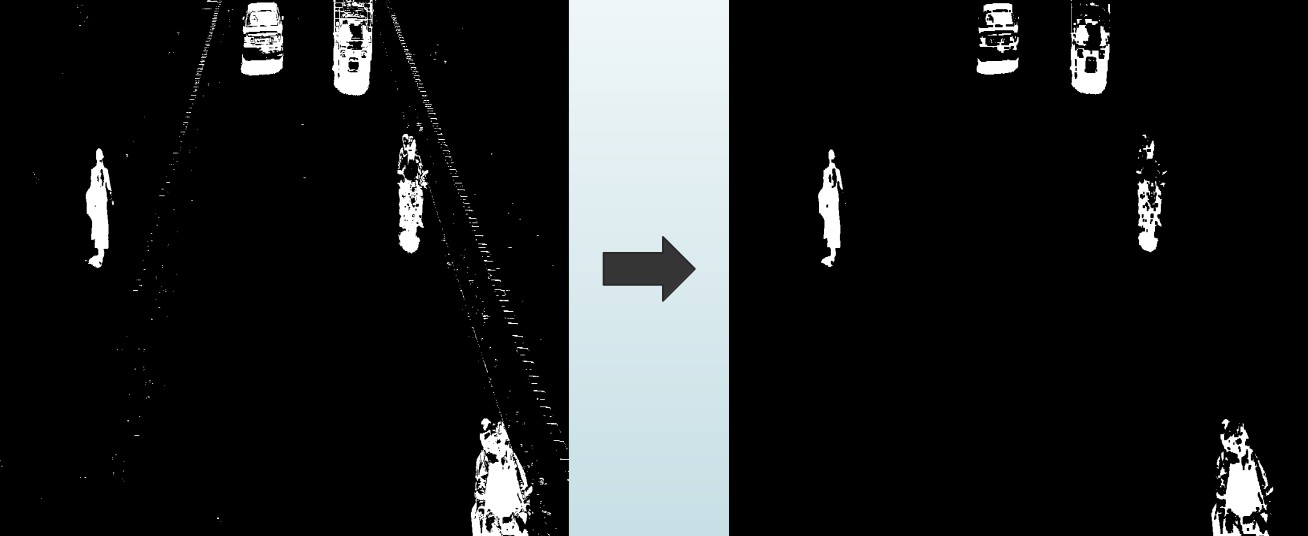
# Thresholding:

After obtaining the subtracted image,the image has to be segmented into regions containing vehicle and the regions containing a non-vehicle. This is decided on the intensity of the region. If the region has an intensitygreater than a specified threshold, it is converted to white and the rest is black. OpenCV provides a function “threshold()” for this purpose.

**Erosion:**

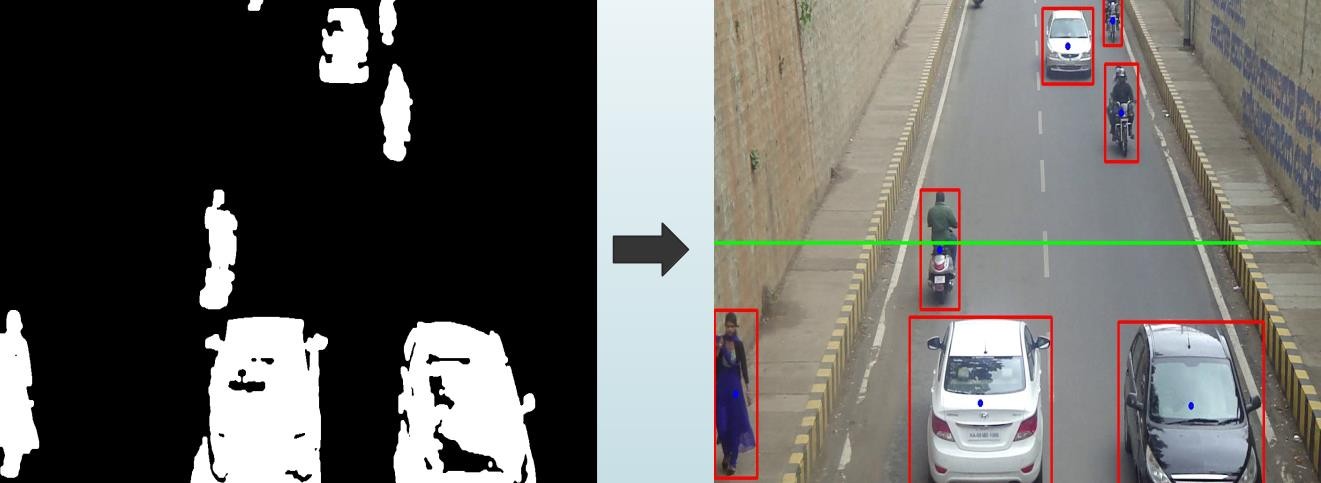
Erosion is one of two fundamental operations in morphological image processing from which all other morphological operations are based. The erosion operator takes two pieces of data as inputs. The first is the image which is to be eroded. The second is a (usually small) set of coordinate points known as a structuring element (also known as a kernel). It is this structuring element that determines the precise effect of the erosion on the input image. The mathematical definition of erosion for binary images is as follows:

We consider each of the foreground pixels in the input image in turn. For each foreground pixel 10 (which we will call the input pixel) we superimpose the structuring element on top of the input image so that the origin of the structuring element coincides with the input pixel coordinates. If for every pixel in the structuring element, the corresponding pixel in the image underneath is a foreground pixel, then the input pixel is left as it is. If any of the corresponding pixels in the image are background, however, the input pixel is also set to background value. OpenCV provides “erode()” function for this process.



**Vehicle Detection :**

After the noise removal, the frame is then ready for vehicle detection. All the regions that are white ( or 1) are considered as vehicles and the rest as non- vehicles. We take contours of all the vehicle regions, calculate the smallest rectangle that will fit the contour and draw bounding boxes. OpenCV provides the functions “findContours()” which finds the contours for all the white regions in the frame.

Another function “boundingRect()” calculates the coordinates and size of smallest rectangle to fit the contour. These rectangles are drwn on the frame. The following figure shows drawing bounding boxes around the vehicles. After this the vehicles are tracked using the centroids of these bounding boxes.

**Vehicle Counting:**

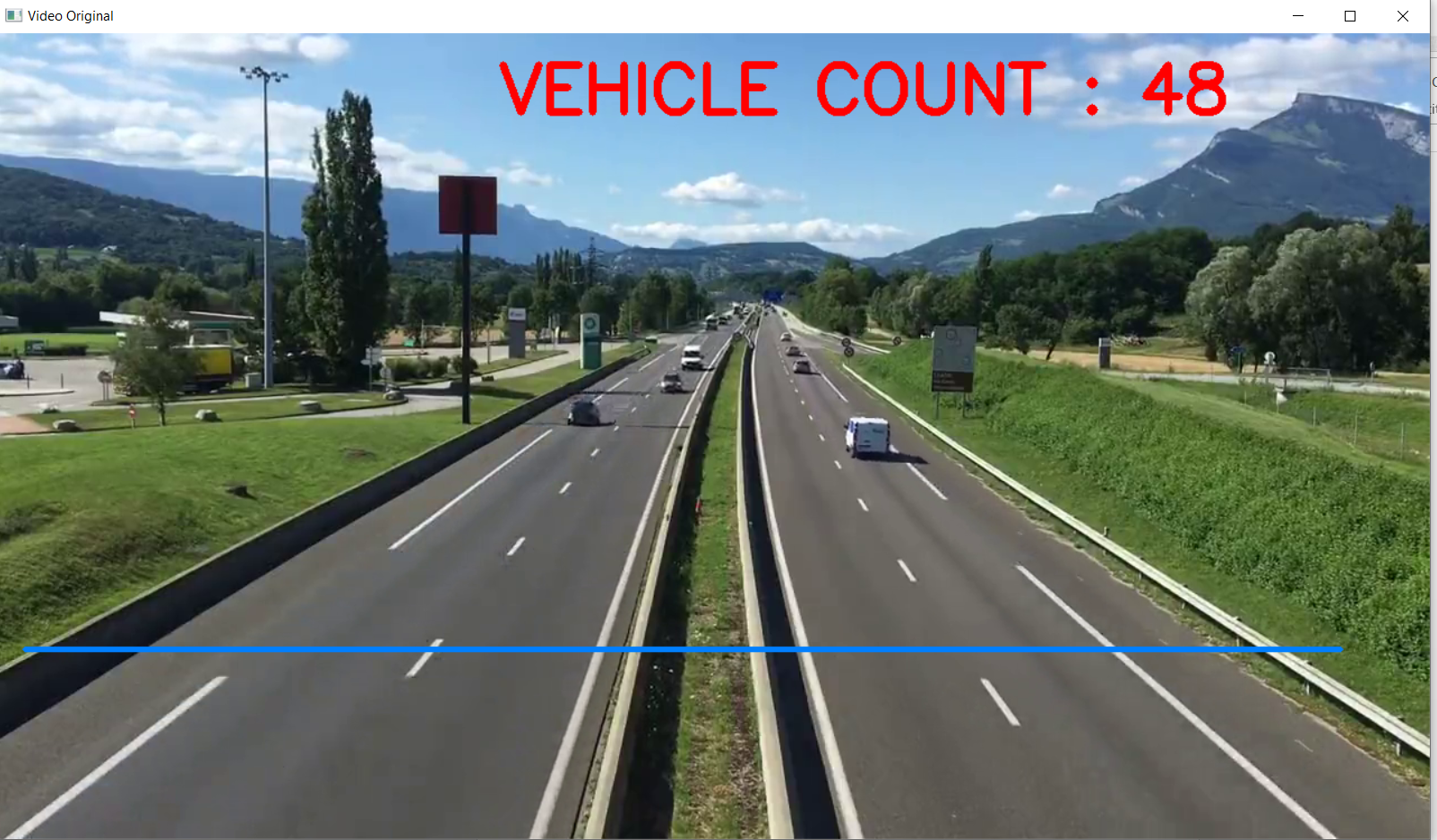
We employ a small tricky algorithm for counting.A vehicle is counted only when the centroid of its bounding box passes through the virtual line.To make sure that a vehicle is counted only once , we also store the centroids of previous frame and use them comparison.A vehicle will be counted if its centroid of current frame has crossed the virtual line and the centroid of previous frame has not crossed the line.

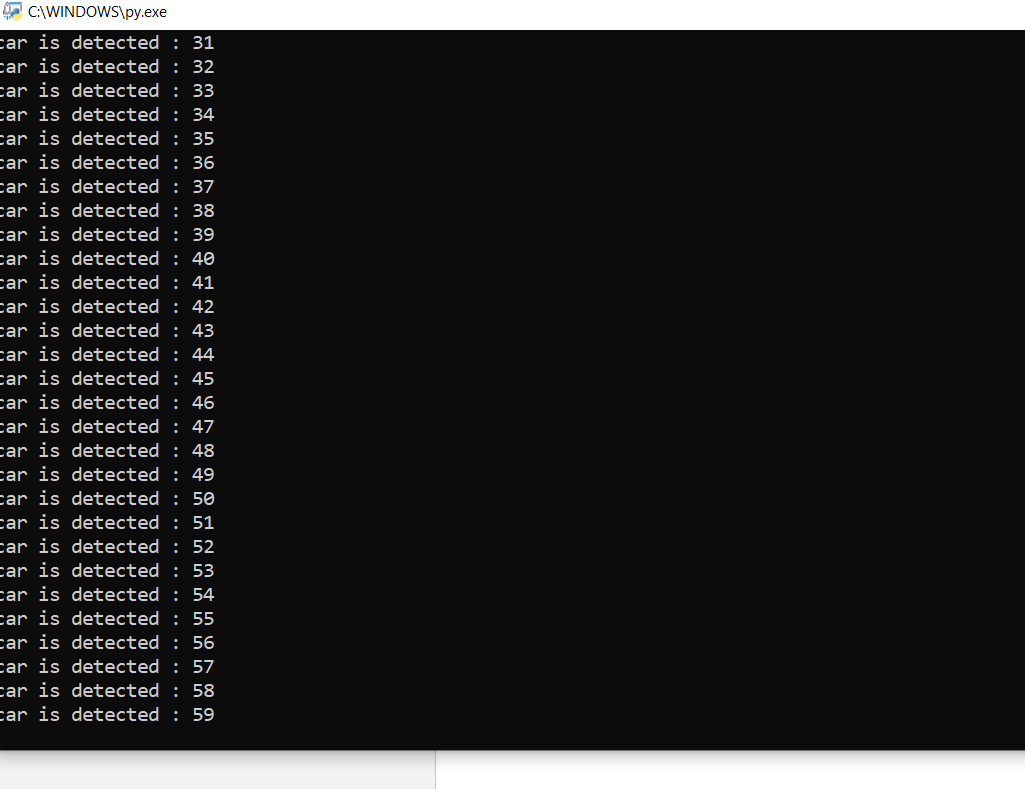
**Algorithm Check if the vehicle crossed the line:**

1. procedure Check crossing(count , old centroids , current centroids
2. for each centroid c1 in current centroids do
3. c2 = position of centroid in previous frame
4. if c1 is above the virtual line and c2 is below the virtual line then
5. count = count + 1
6. end if
7. if c1 is below the virtual line and c2 is above the virtual line then
8. count = count + 1
9. end if
10. end for
11. return count
12. end procedure

# RESULTS

# 





**CONCLUSION**

Vehicle Detection and Counting is necessary to establish an enriched in formation platform and improve the quality of intelligent transportation systems. This solution for Vehicle Detection, Classification and Counting can be used in traffic monitoring, parking area allocation etc. A system has been developed to detect and count dynamic vehicles on highways efficiently. The experimental results show that the accuracy of counting vehicles was 90% with all the false positives involved. Complex algorithms can be employed in the future which focus on feature-based detection of vehicles to handle occlusion and light related issues. Researchers are also working for detection of vehicles using the headlights and the tail lights.

**REFERENCES:**

1. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=6112843>
2. htt[ps://www.behanc](http://www.behance.net/gallery/Vehicle-Detection-Tracking-and-Counting/4057777)e[.net/gallery/Vehicle-Detection-Tracking-and-Counting/4057777](http://www.behance.net/gallery/Vehicle-Detection-Tracking-and-Counting/4057777)
3. F. Bardet, et al., "Unifying real-time multi-vehicle tracking and categorization," in Intelligent Vehicles Symposium, 2009 IEEE, 2009, pp. 197-202.
4. Huei-Yung Lin and Juang-Yu Wei, “A Street Scene Surveillance System for Moving Object Detection, Tracking and Classification”, Proceedings of the IEEE Intelligent Vehicles Symposium Istanbul, Turkey, June 13-15, 2007.