Vehicle Detection and Counting
(Traffic surveillance)

Abstract-
Computer Vision techniques are frequently more appropriate because of the fact that environment, such as traffic in this case, is not disturbed and modification is achieved more easily.
A portable and inexpensive method is discussed for vehicle detection, counting and in some extent classification. A video consists of frame sequences, which are later taken to perform analysis. Besides, in order to detect the vehicles in motion, a technique called adaptive background subtraction is carried out.
Moreover, in this paper, differential morphology closing profile is used for extracting the vehicle more effectively from the traffic scene.

Contents

Introduction

Traffic counting and vehicle classification are fundamental data for a diverse amount of transportation projects; for instance, transportation planning or modern intelligent transportation systems. Nevertheless, traffic surveillance still relies on sensors for finding traffic parameters. On the other hand, there has been relatively little work carried out in the field of vehicle classification due to the fact that this is a challenging problem. Furthermore, a great idea to start with is mounting a static camera on a tall structure looking down on the traffic scene.

Applications

Monitoring activities at traffic intersections, which allow the detection of congestions.

Assisting in regulating traffic.

The need for highway usage statistics in large metropolitan areas.

Playing an important role for civilian and military applications.

Objectives

Detection of multiple moving vehicles in a video sequence
Counting the number of vehicles passing through a defined line point
Classifying each vehicle by using its dimensions

Problem Statement

Developing a system being able to count the number of vehicles passing through a defined line point and classifying them by dimensions through a multi line road by using a computer vision based approach.

Besides, there are some parameters that have to be identified:

Number of vehicles
Kinds of vehicles

Tools

- Implemented on python 2.7.9 and Opencv library
- Using videos and extracting their frames.
- Computer vision techniques such as background subtraction, pre-processing, post-processing, etc
- Running on GNU/Linux

System overview

Main steps:

- Capturing the video in order to get the video frames

Capturing traffic flow video either from a camera or a pre-recorded source and converting it into various frames.

- Pre-processing

The first task of vehicle detection is to convert true color input frame into gray
scale. Input frame $I = [I_R\ I_G\ I_B]^T$, is a combination of the three color channels: Red, Green and Blue. Furthermore a gaussian filter is used.

- Background extraction

In the adaptive background subtraction algorithm, the first frame is considered as the reference one, which the future frames are going to be subtracted from, but it has to be updated by following some criteria, thus, adaptive background subtraction. After subtraction, only pixels not changed are not going to be taken into account and the ones that are can be accepted according to some threshold.

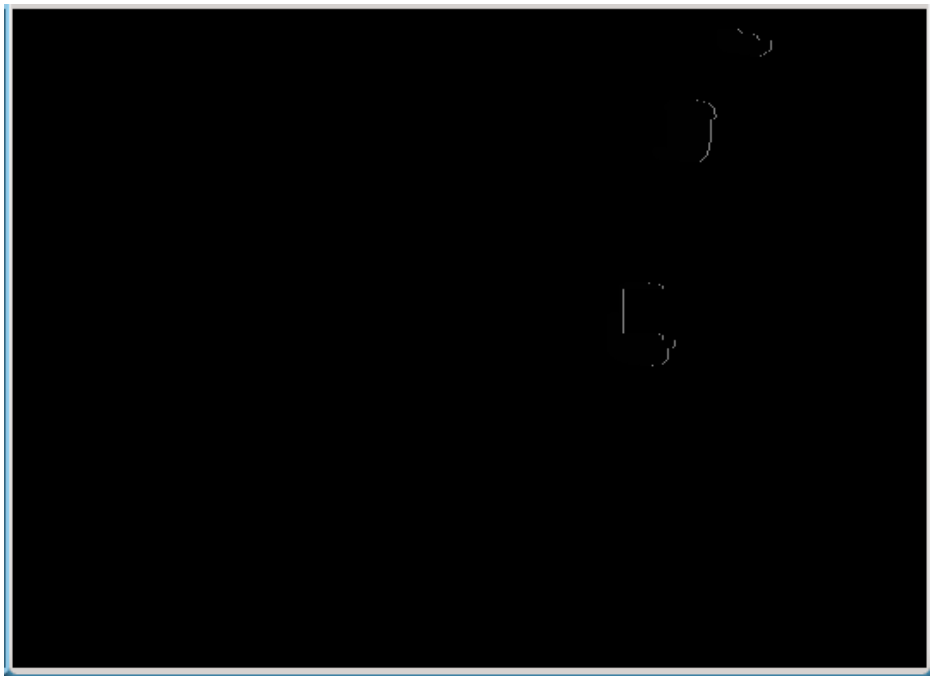
- Post-processing

Smoothing the image, applying a threshold to it in order to get rid of some shadows or some occlusion.

- Segmentation

There was a segmentation step called **Mathematical morphology,** which is used for analyzing vehicle shape characteristics such as size and connectivity, which are not easily accessed by linear approaches.

- The closing operation was carried out and it consists of erosion and dilation.
- The main task achieved by this operation is closing small holes inside the foreground objects, or small black points on the object.



- Foreground detection to get blobs (objects in motion)

Using the subtracted background image in order that all the moving vehicles/objects can be tracked and counted.

In a video sequence an object is said to be moving, when it is changing its location with respect to its background

- Vehicle classification and counting

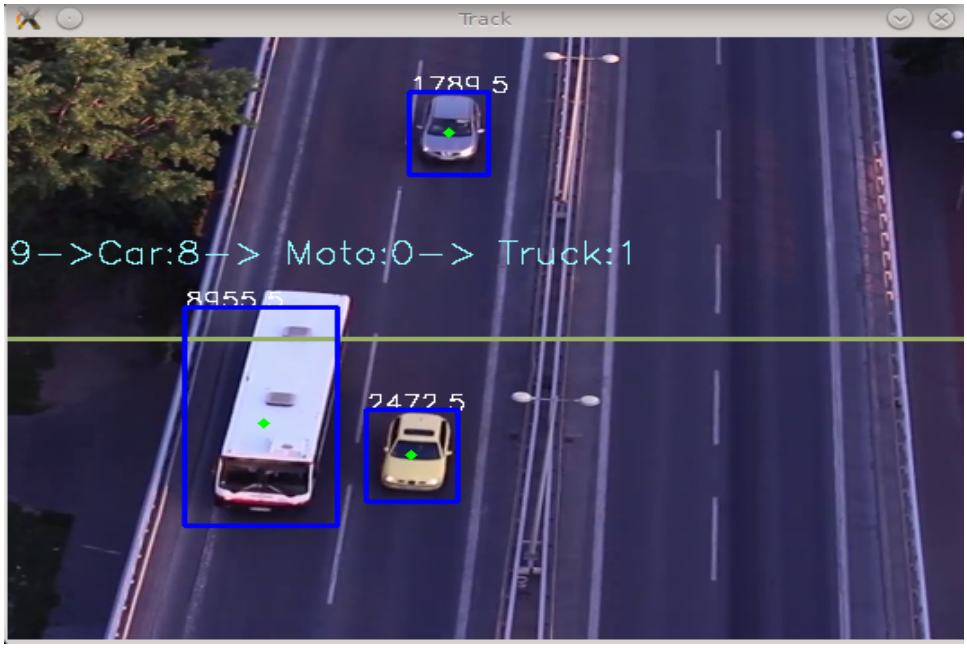
In this project, the detected vehicle regions will be
classified as cars, heavy vehicles like trucks or buses and

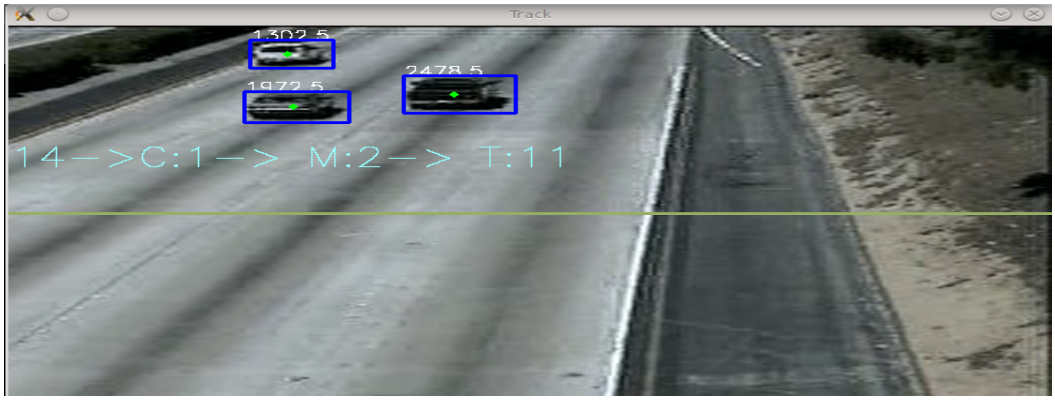
motorcycles.

• Number of Detected cars (represented by letter **C**)

• Number of Detected Heavy Vehicles (Trucks, buses and so on represented by letter **T**)

• Number of Detected Motor-Cycles (represented by letter **M**)





Difficulties

- Occlusion either by vehicles or background obstacles such as road signals, trees, weather, pedestrians, etc.
- Noise in the images.

- Complexity in vehicle motion.
- Complexity in vehicle shapes since different vehicles have different shapes.

Some Implementation details

Counting and classifying

- To carry this out, a region of the image was cropped (the one where the road is) and therefore avoiding to process unnecessary image areas. Line detection was performed to get the lines from the highway and then the minor and major x coordinate were gotten from the leftmost line to the rightmost line.

```
def detect_lines(gray): #getting to detect lines for finding the area intended for analysis
    edges = cv2.Canny(gray,100,300) #canny edge detector
    lines = cv2.HoughLines(edges, 2, np.pi/180, 40)[0]
    lines1=[]
    for (rho, theta) in lines[:5]:
        # lines (only choose the 5 strongest ones)
        a = np.cos(theta)
        b = np.sin(theta)
        x0 = a*rho
        x1 = int(x0 + 1000*(-b))
        x2 = int(x0 - 1000*(-b))
        if x1 and x2:
            lines1.append((x1,x2))
    x_m=min(lines1,key=lambda item:item[0])
    x_M=max(lines1,key=lambda item:item[1])
    return (x_m[0]+80,x_M[1])

def crop_area(framex): #get the area to be cropped
    framex=cv2.resize(framex,(900,400),interpolation = cv2.INTER_CUBIC) # using cubic interpolationx
    framex=cv2.cvtColor(framex,cv2.COLOR_RGB2GRAY)
    return detect_lines(framex)
```

- In order to achieve counting, a line was defined within the center of the image and all the blobs crossing this line were counted

- In order to accomplish classifying, the area of each object was obtained and then compared to a threshold to distinguish among: cars, motorcycles and other kinds of cars.

```
def bounding_boxes(frame,contours,y_linea,x_linea):
    counter=0
    counter_carro=0
    counter_moto=0
    counter_camion=0

    for contour in contours:
        (x,y,w,h) = cv2.boundingRect(contour)
        if w > 10 and h > 15: #restricting bounding boxes to at least some h and w
            M = cv2.moments(contour)
            c_x = int(M['m10']/M['m00']) #getting coordinates for mass center
            c_y = int(M['m01']/M['m00'])
            area=M["m00"] #filtering kinds of cars by area
            diff=c_y-y_linea
            if diff>=0 and diff<=8 and c_x<=x_linea and c_x>=x_linea//2: #right lane
                if area<=450:
                    counter_moto+=1
                if area>450 and area<=3000:
                    counter_carro+=1
                if area>3000:
                    counter_camion+=1
                counter+=1
            else:
                cv2.rectangle(frame, (x,y), (x+w,y+h), (255, 0, 0), 2) #Drawing bounding box
                cv2.circle(frame,(int(c_x),int(c_y)),1,(0,255,0),4)
            diff2=y_linea-c_y
            if diff<0 and diff2<=8 and c_x<=x_linea//2 and c_x>=0: #left lane
                if area<=450:
                    counter_moto+=1
                if area>450 and area<=3000:
                    counter_carro+=1
                if area>3000:
                    counter_camion+=1
                counter+=1
```

The process in brief

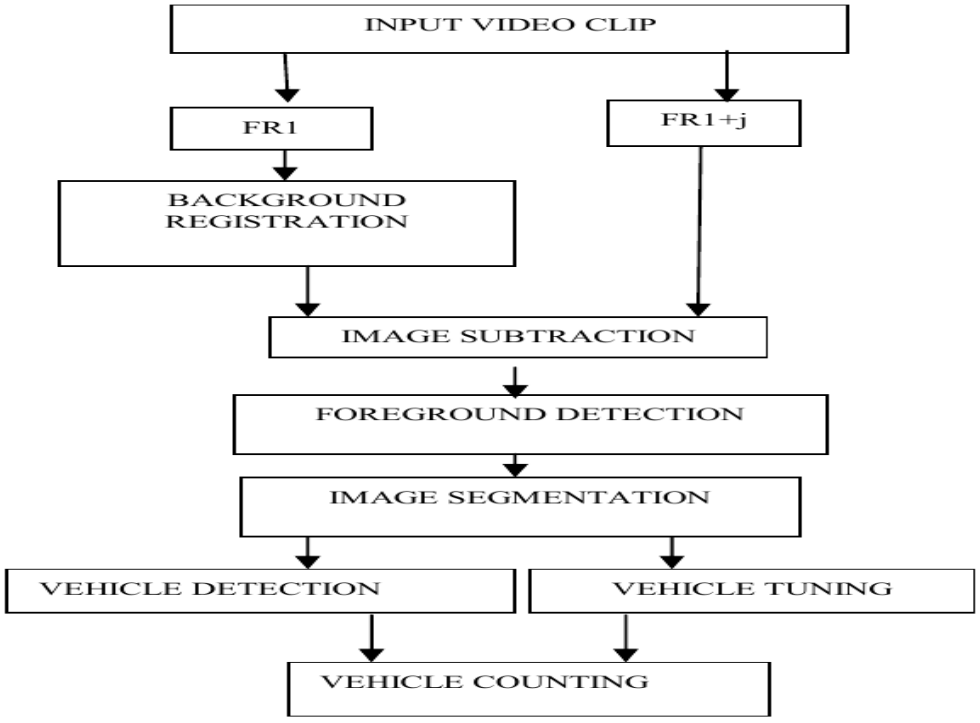


Fig.1.Architecture and modelling

[1] Mrs. Prema M. Daigavane, Dr. Preeti Bajaj, “Real Time Vehicle Detection and Counting Method for Unsupervised Traffic Video on Highways”

Future Scope

- Improving the system classification technique.
- Reporting live statistics online about the behaviour.
- Improving the detection even on constant occlusions.
- Improving the counting and classification techniques

The project is functional so far, but due to noise or inconsistencies, there are some issues with counting and classifying.

Bibliography

[1] Mrs. Prema M. Daigavane, Dr. Preeti Bajaj, “Real Time Vehicle Detection and Counting Method for Unsupervised Traffic Video on Highways”, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.8, August 2010.

[2] ravikiran js,http://www.academia.edu/8517960/Vehicle_detection_and_counting_in_traffic_video_on_highway_CHAPTER, India

[3] Raad Ahmed Hadi, Ghazali Sulong and Loay Edwar George, “VEHICLE DETECTION AND TRACKING TECHNIQUES : A CONCISE REVIEW”, Signal & Image Processing : An International Journal (SIPIJ) Vol.5, No.1, February 2014

[4] Alexander Mordvintsev & Abid K. Last updated on Oct 31, 2014,http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_imgproc/py_filtering/py_filtering.html