Vehicle Detection and Counting (Traffic surveillance)

Contents

Introduction

- Due to the high vehicle growth, traffic monitoring has become a great research topic.
- Detecting and counting vehicles is a challenging problem in computer vision.
- Essential for road usage and management.
- Using a camera mounted 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

Video capturing -> Background modelling and subtraction -> Vehicle detection -> Counting and classifying

Main steps:

Capturing the video in order to get the video frames

Pre-processing

Background extraction

Post-processing

Segmentation

Getting to detect blobs in foreground

Vehicle classification and counting

Difficulties

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

- Noise in the images.

- Complexity in vehicle motion.

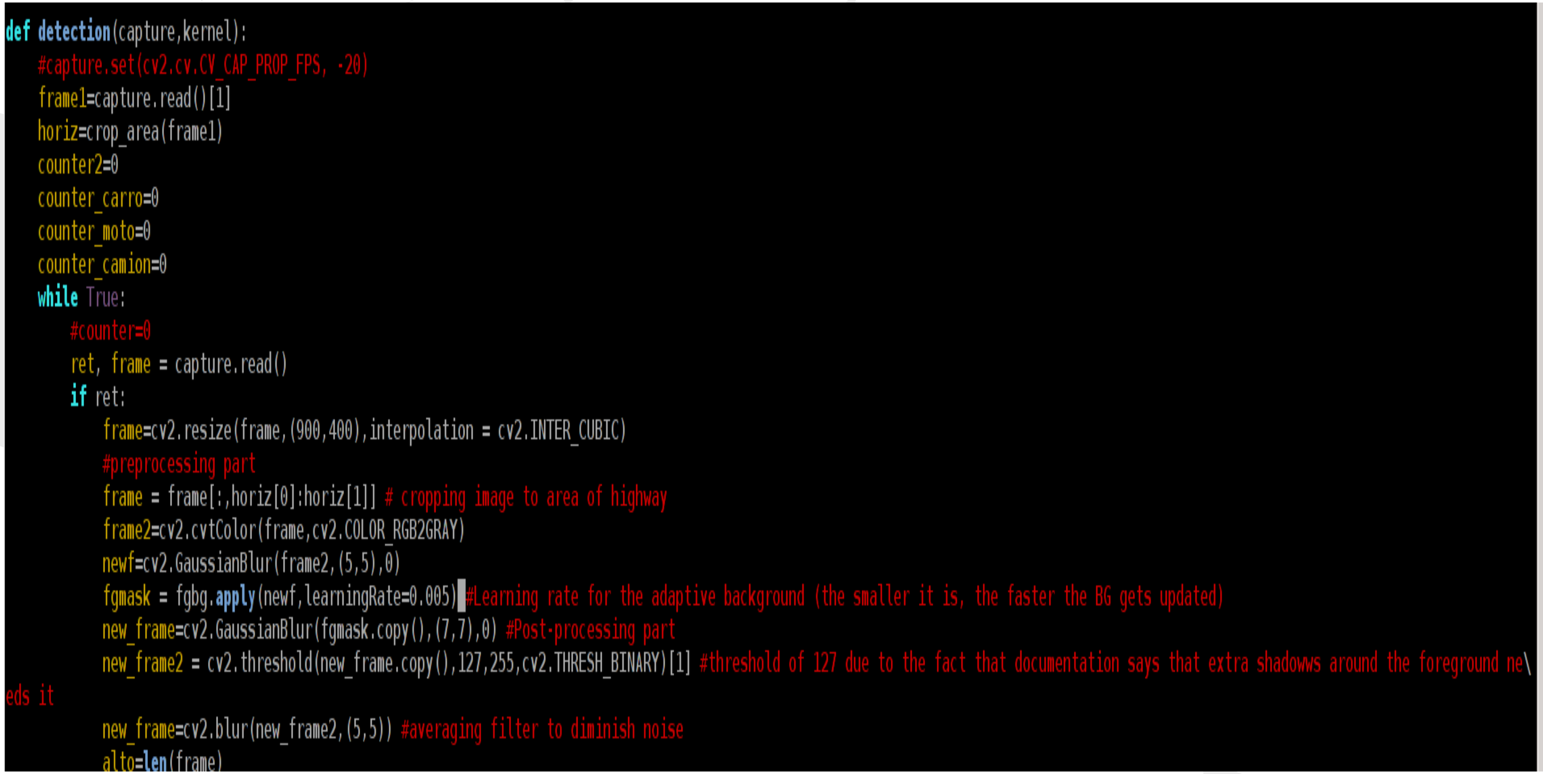
- Complexity in vehicle shapes since different vehicles have different shapes.

Implementation

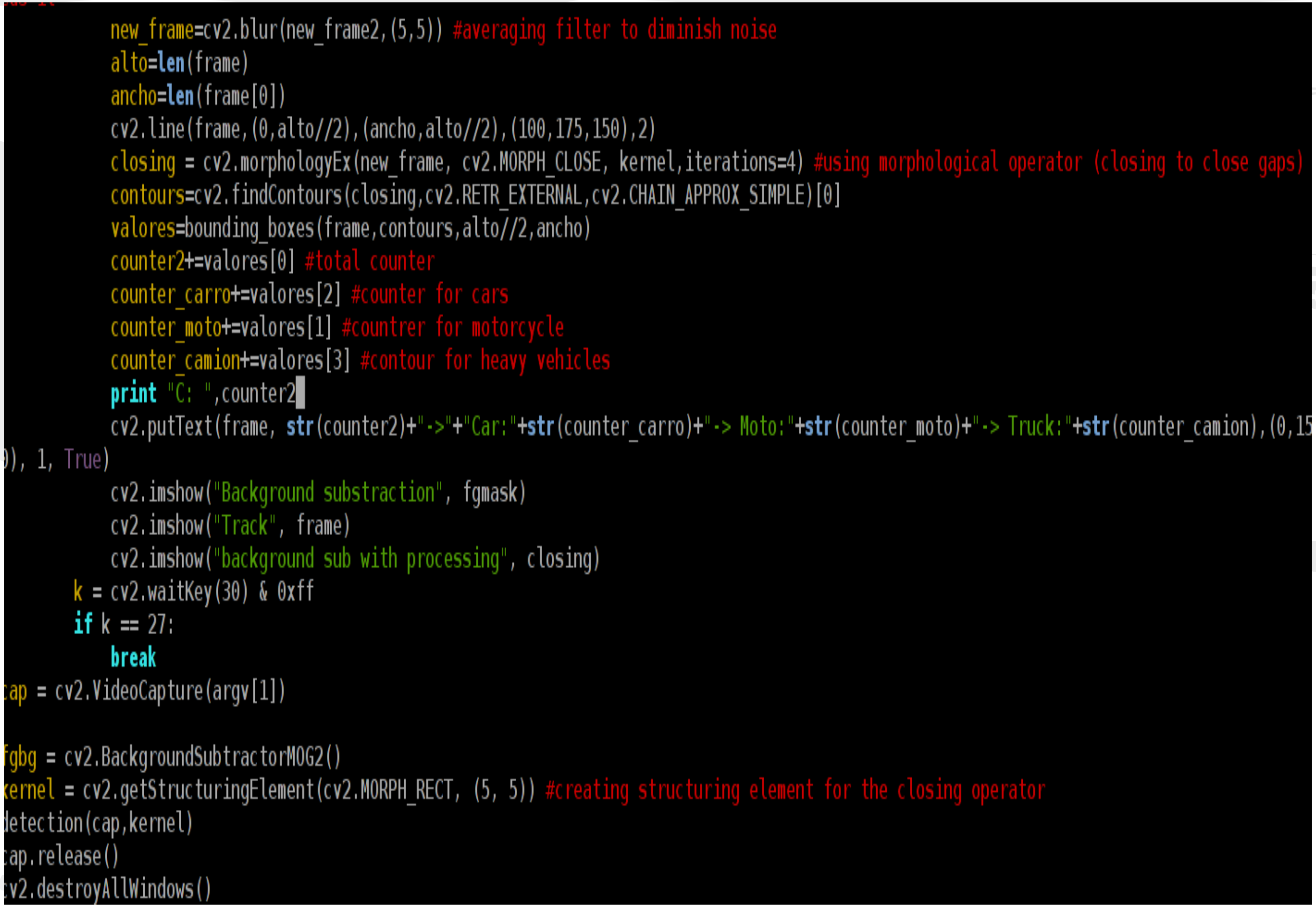
Segmentation:

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.



```
def detection(capture,kernel):
    #capture.set(cv2.cv.CV_CAP_PROP_FPS, -20)
    frame1=capture.read()[1]
    horiz=crop_area(frame1)
    counter2=0
    counter_carro=0
    counter_moto=0
    counter_camion=0
    while True:
        #counter=0
        ret, frame = capture.read()
        if ret:
            frame=cv2.resize(frame,(900,400),interpolation = cv2.INTER_CUBIC)
            #preprocessing part
            frame = frame[:,horiz[0]:horiz[1]] # cropping image to area of highway
            frame2=cv2.cvtColor(frame,cv2.COLOR_RGB2GRAY)
            newf=cv2.GaussianBlur(frame2,(5,5),0)
            fgmask = fgbg.apply(newf,learningRate=0.005)#Learning rate for the adaptive background (the smaller it is, the faster the BG gets updated)
            new_frame=cv2.GaussianBlur(fgmask.copy(),(7,7),0) #Post-processing part
            new_frame2 = cv2.threshold(new_frame.copy(),127,255,cv2.THRESH_BINARY)[1] #threshold of 127 due to the fact that documentation says that extra shadowws around the foreground ne\
eds it
            new_frame=cv2.blur(new_frame2,(5,5)) #averaging filter to diminish noise
            alto=len(frame)
```



```
        new_frame=cv2.blur(new_frame2,(5,5)) #averaging filter to diminish noise
        alto=len(frame)
        ancho=len(frame[0])
        cv2.line(frame,(0,alto//2),(ancho,alto//2),(100,175,150),2)
        closing = cv2.morphologyEx(new_frame, cv2.MORPH_CLOSE, kernel,iterations=4) #using morphological operator (closing to close gaps)
        contours=cv2.findContours(closing,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)[0]
        valores=bounding_boxes(frame,contours,alto//2,ancho)
        counter2+=valores[0] #total counter
        counter_carro+=valores[2] #counter for cars
        counter_moto+=valores[1] #countrer for motorcycle
        counter_camion+=valores[3] #contour for heavy vehicles
        print "C: ",counter2
        cv2.putText(frame, str(counter2)+"->"+"Car:"+str(counter_carro)+"-> Moto:"+str(counter_moto)+"-> Truck:"+str(counter_camion),(0,15
)), 1, True)
        cv2.imshow("Background substraction", fgmask)
        cv2.imshow("Track", frame)
        cv2.imshow("background sub with processing", closing)
    k = cv2.waitKey(30) & 0xff
    if k == 27:
        break
ap = cv2.VideoCapture(argv[1])

fgbg = cv2.BackgroundSubtractorMOG2()
ernel = cv2.getStructuringElement(cv2.MORPH_RECT, (5, 5)) #creating structuring element for the closing operator
detection(cap,kernel)
ap.release()
v2.destroyAllWindows()
```

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.

- 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.

A general overview

- A vehicle can be simply defined as a set of pixels moving in a coherent manner, either as a darker background over a lighter region or vice versa.

[2]

[1]

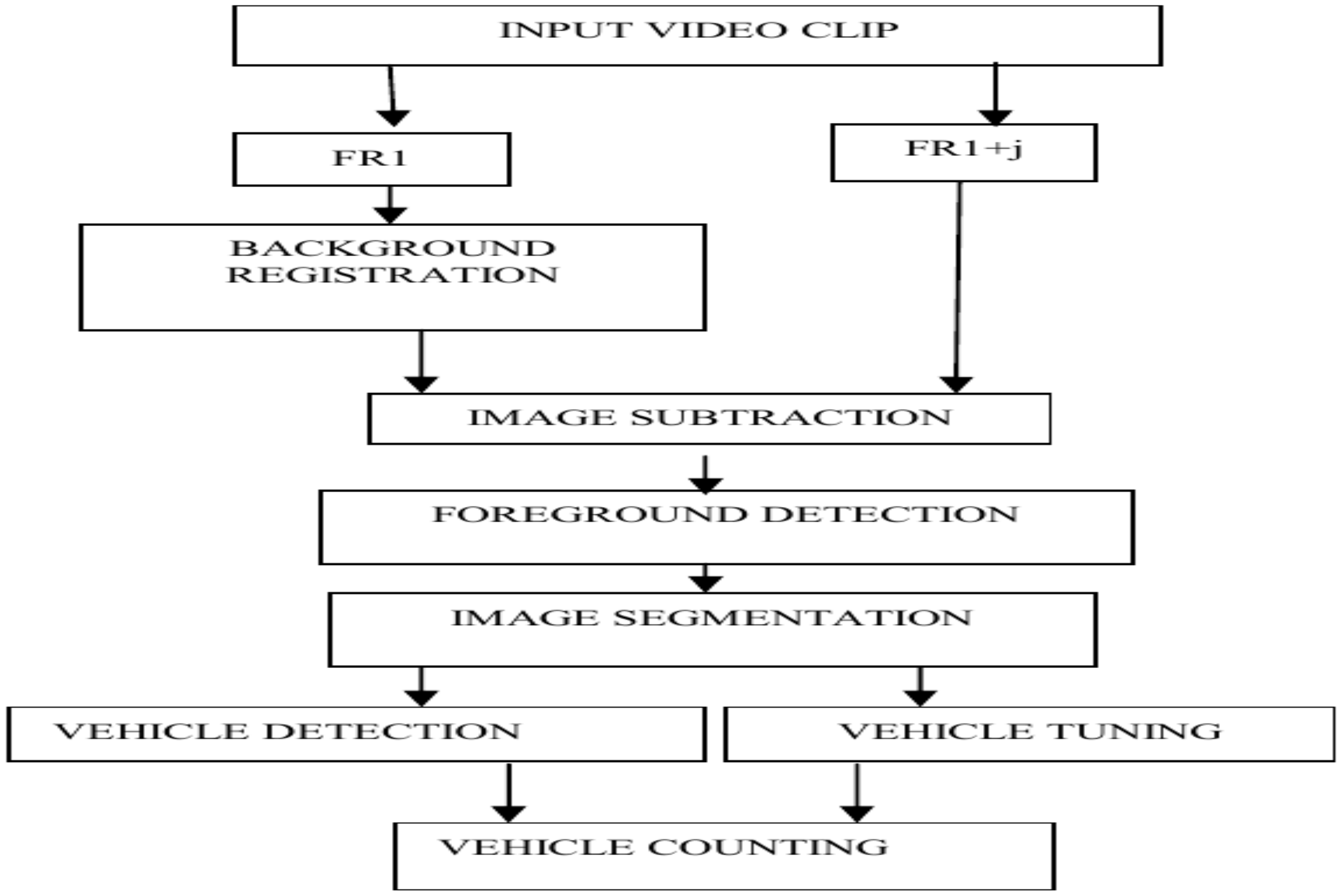
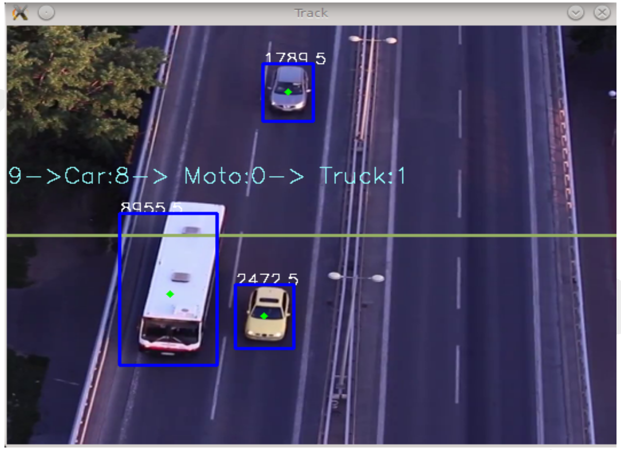


Fig.1.Architecture and modelling

Samples

Background subtraction involves computing a reference image and subtracting each new frame from this image and thresholding the result. Adaptive BG subs. was used.



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