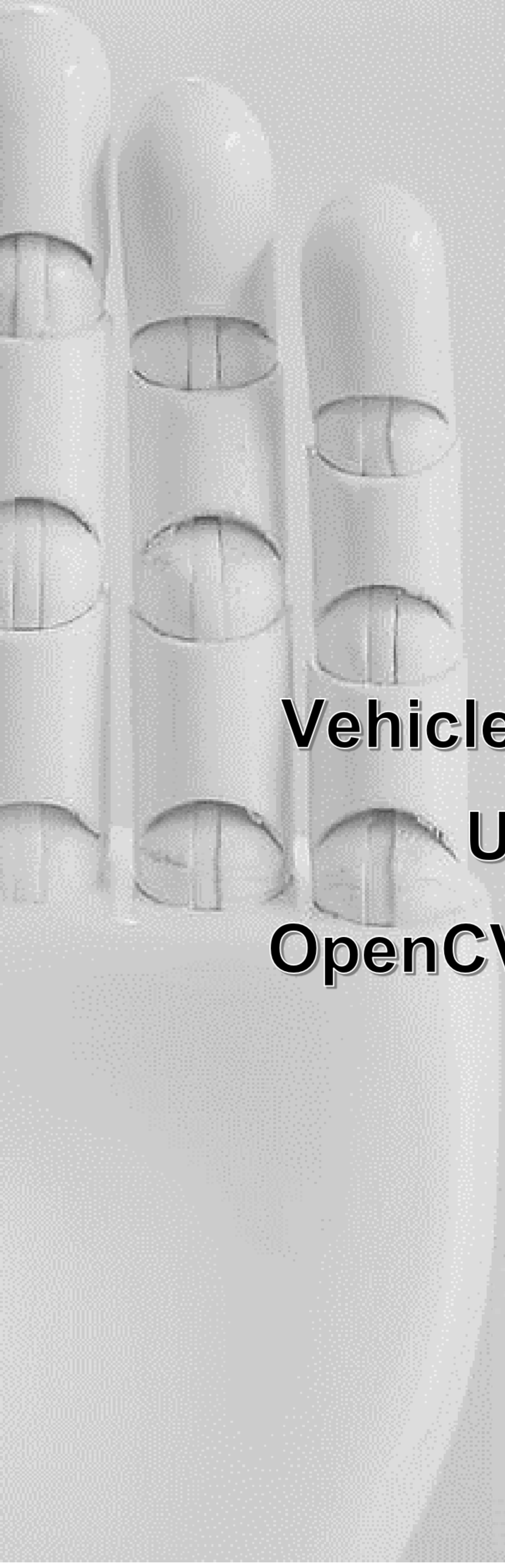


ARTIFICIAL INTELLIGENCE PROJECT (UCS411)

Vehicle detection Using OpenCV & Python

Submitted by :

Jasmine Kaur (102117041)
Jasleen (102117042)
Harmandip Singh (102117040)
Shivam Bector (102117037)

ABSTRACT

This project intends to develop a vehicle detection program using digital image processing technique. Therefore this project needs a video input to make the system work. The system is designed to track the vehicle position. This proposed method is using the image processing technique. This system consists of 4 major steps:

1) Frame differencing

2) Image thresholding

3) Finding Contours

4) Image dilation

The rate of **accuracy** for this system is expected to have **90%.**

Language used: Python

Software used: Pycharm

Libraries used: opencv-python, numpy, opencv-contrib-python

Applications of this model:

Integrating a vehicle detection system in a traffic light camera, we can easily track a number of useful things simultaneously:

1. How many vehicles are present at the traffic junction during the day
2. What time does the traffic build up
3. What kind of vehicles are traversing the junction (heavy vehicles, cars, etc.)
4. Is there a way to optimize the traffic and distribute it through a different street

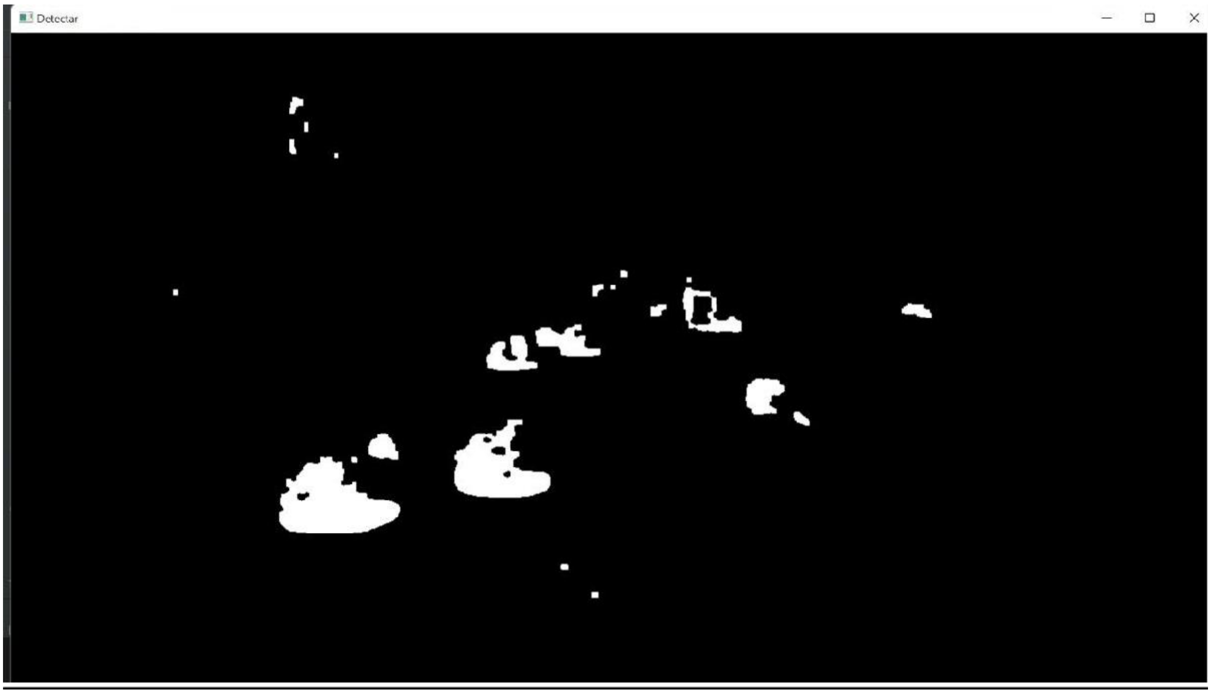
   And so on. The applications are endless!

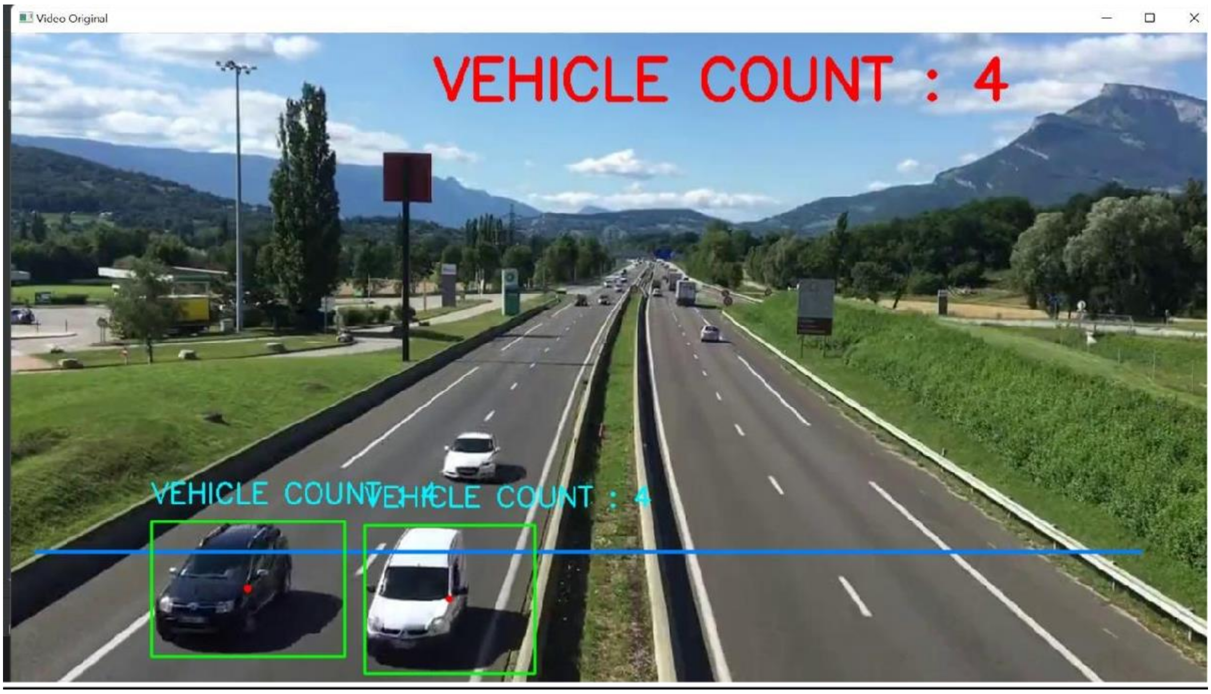
CODE:

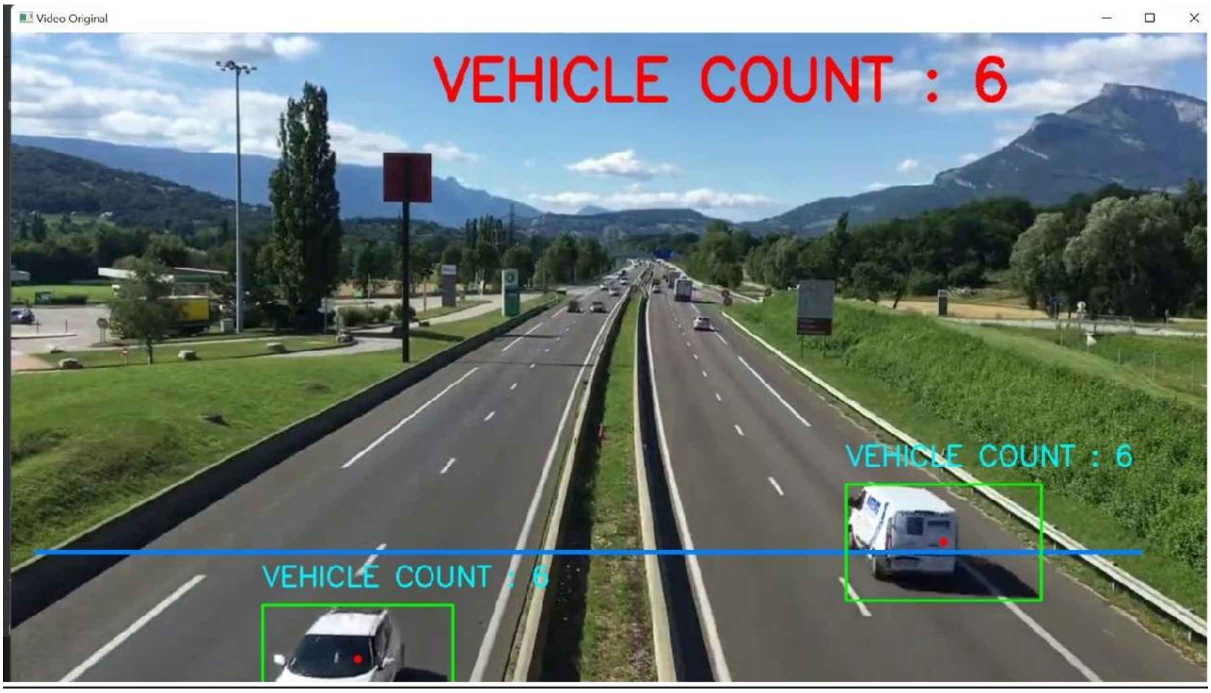
```
import cv2
import numpy as np
from time import sleep

# minimum width of rectangle
width_min = 80
# minimum height of rectangle
height_min = 80

# Error allowable between pixels
offset = 6

# to put the line on frame as when any vechile cross the
line and then it get counted
position_line = 550
delay = 60

# a list if anything detected it can get appended
detec = []
count = 0

# just to find the mid-point of rectangle on the vechile
for red dot
def find_center(x, y, w, h):
    x1 = int(w / 2)
    y1 = int(h / 2)
    cx = x + x1
    cy = y + y1
    return cx, cy


# create a video capture object and help to display the
video
cap = cv2.VideoCapture('video.mp4')

# this is one of algorithm in cv2 and it is known as
subtractor. it is used to substract the background of our
object in video
subtractor = cv2.bgsegm.createBackgroundSubtractorMOG()

while True:
    # return the specific frame and read the video
    ret, frame1 = cap.read()
    tempo = float(1 / delay)
    sleep(tempo)
    # used to convert color of specific frame here it is
frame1
    # cvtColor means convert color
    grey = cv2.cvtColor(frame1, cv2.COLOR_BGR2GRAY)


    # Gaussian blur is the result of blurring an image.
It is commonly used when reducing the size of an image.

    blur = cv2.GaussianBlur(grey, (3, 3), 5)
    # now we create a variable as image subtractor and we
applying the algo on it
    img_sub = subtractor.apply(blur)

    # Dilates an image by using a specific structuring
element. The function dilates the source image using the
specified structuring element that
    # determines the shape of a pixel.

    dilat = cv2.dilate(img_sub, np.ones((5, 5)))

    # Returns a structuring element of the specified size
and shape for morphological operations(Morphology is a
broad set of image
    # processing operations that process images based on
shapes. In a morphological operation, each pixel in the
image is adjusted based on the
    # value of other pixels in its neighborhood).The
function constructs and returns the
    # structuring element that can be further passed to
construct an arbitrary binary mask yourself and use it as
the structuring element.

    # MORPH_ELLIPSE an elliptic structuring element, that
is, a filled ellipse inscribed into the rectangle

    kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE,
(5, 5))

# this is simply give the black and white phase of our
video as in such as a bcknds
    dilatada = cv2.morphologyEx(dilat, cv2.MORPH_CLOSE,
kernel)
    dilatada = cv2.morphologyEx(dilatada,
cv2.MORPH_CLOSE, kernel)

    # just to count the no of objects as in our case here
it thin the image from getting to subtractor
    contorno, h = cv2.findContours(dilatada,
cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
```

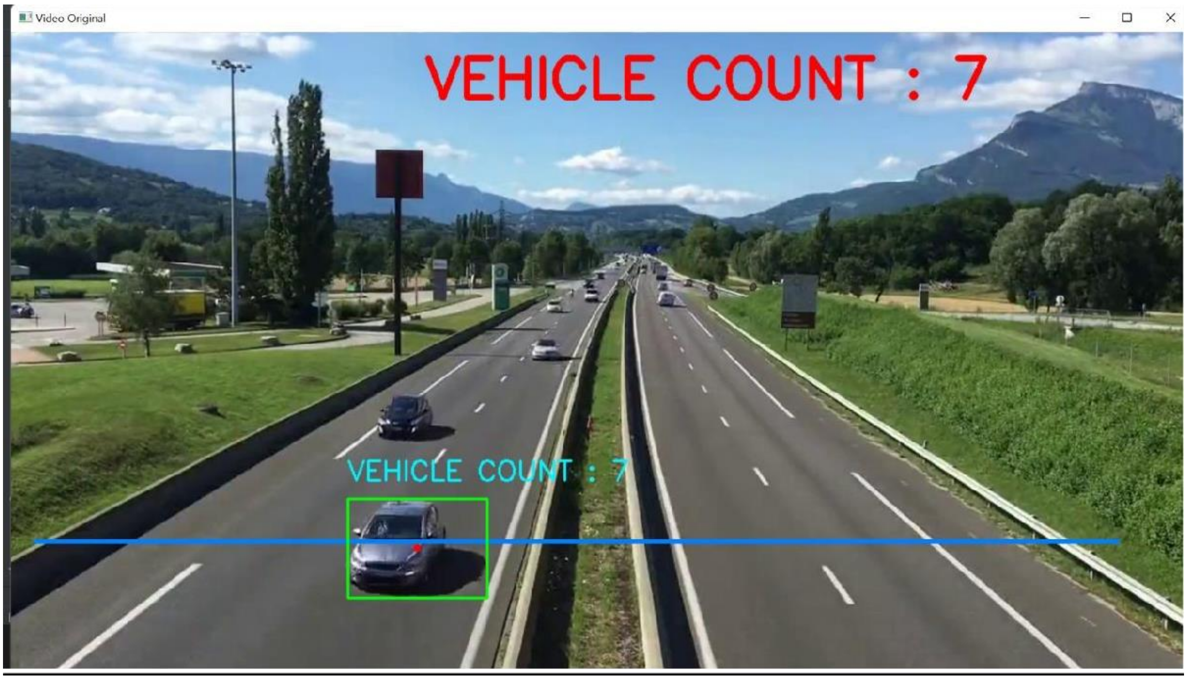
```
# here upto it works as it do all background in black
colour and all vechiles in white ones.


# just giving the specification of line like position or
color or etc etc
    cv2.line(frame1, (25, position_line), (1200,
position_line), (255, 127, 0), 3)

    # for putting rectangle on vechiles
    for (i, c) in enumerate(contorno):


        # here x, y represents the x and y plane
        # rectangle contains length and breadth here I
take w for width and h for height or length

        (x, y, w, h) = cv2.boundingRect(c)
        validar_contorno = (w >= width_min) and (h >=
height_min)
        if not validar_contorno:
            continue

        cv2.rectangle(frame1, (x, y), (x + w, y + h), (0,
255, 0), 2)
        cv2.putText(frame1, "VEHICLE COUNT : " +
str(count), (x, y-20), cv2.FONT_HERSHEY_SIMPLEX, 1, (255,
244, 0), 2)

        centro = find_center(x, y, w, h)
        detec.append(centro)
        # for creating a circle
        cv2.circle(frame1, centro, 4, (0, 0, 255), -1)

# for printing the output and count the no of vechiles on
window
        for (x, y) in detec:
            if y < (position_line + offset) and y >
(position_line - offset):
                count += 1

                # when any vechile pass through the line
the color changes.
                cv2.line(frame1, (25, position_line),
(1200, position_line), (0, 127, 255), 3)
                detec.remove((x, y))
                print("car is detected : " + str(count))

    cv2.putText(frame1, "VEHICLE COUNT : " + str(count),
(450, 70), cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 0, 255), 5)
    cv2.imshow("Video Original", frame1)
    cv2.imshow("Detectar", dilatada)
    # to stop the window

    if cv2.waitKey(1) == 27:
        break
# after completing all task just to release all windows
cv2.destroyAllWindows()
cap.release()
```

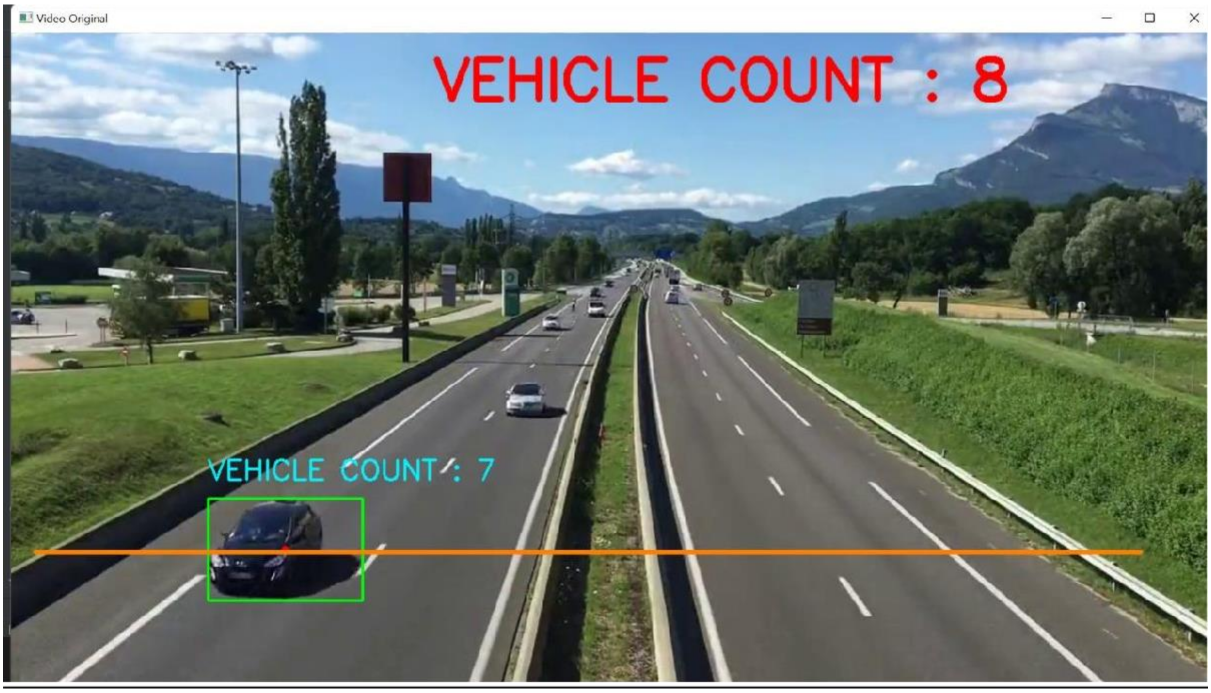
IMPLEMENTATION:

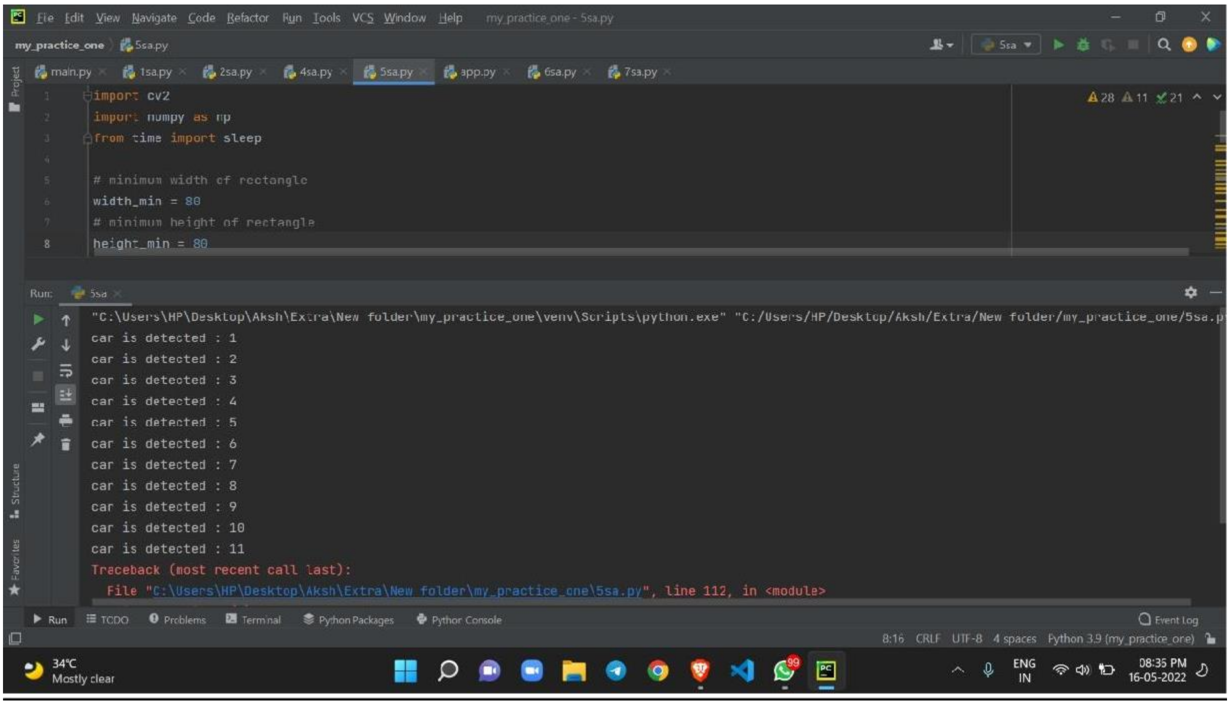





Video Original
VEHICLE COUNT : 6
VEHICLE COUNT : 6
VEHICLE COUNT : 6


Video Original
VEHICLE COUNT : 7
VEHICLE COUNT : 7





```python
import cv2
import numpy as np
from time import sleep

# minimum width of rectangle
width_min = 80
# minimum height of rectangle
height_min = 80
```

```
"C:\Users\HP\Desktop\Aksh\Extra\New folder\my_practice_one\venv\Scripts\python.exe" "C:/Users/HP/Desktop/Aksh/Extra/New folder/my_practice_one/5sa.p
car is detected : 1
car is detected : 2
car is detected : 3
car is detected : 4
car is detected : 5
car is detected : 6
car is detected : 7
car is detected : 8
car is detected : 9
car is detected : 10
car is detected : 11
Traceback (most recent call last):
  File "C:\Users\HP\Desktop\Aksh\Extra\New folder\my_practice_one\5sa.py", line 112, in <module>
```