Filename: C:\Users\Avinash\memscript.py

```
Line #    Mem usage    Increment  Occurences   Line Contents
=============================================================
     8     73.5 MiB     73.5 MiB           1   def detect():
     9     75.0 MiB      1.4 MiB           1       capture=cv2.VideoCapture(r'C:\Users\Avinash\Downloads\laser.mkv') #we use 0 for webcam and file name of
    10     75.0 MiB      0.0 MiB           1       start = time.time()
    11     75.0 MiB      0.0 MiB           1       kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(3,3))
    12                                             #fgbg = cv2.bgsegm.createBackgroundSubtractorMOG()
    13     75.0 MiB      0.0 MiB           1       fgbg = cv2.createBackgroundSubtractorMOG2(history=100, varThreshold=50, detectShadows=True)
    14     75.0 MiB      0.0 MiB           1       countt=0
    15     75.0 MiB      0.0 MiB           1       totaltime=0
    16     75.0 MiB      0.0 MiB           1       printt=True #used as a key to print the execution time once
    17     75.0 MiB      0.0 MiB           1       while True:
    18    211.9 MiB   -803.1 MiB        1393           e1 = cv2.getTickCount()
    19    211.9 MiB   -803.1 MiB        1393           countt+=1
    20    211.9 MiB   -764.3 MiB        1393           ret, frame = capture.read()
    21
    22    211.9 MiB   -801.5 MiB        1393           if ret:
    23
    24    211.0 MiB  -1527.2 MiB        1393               fgmask = fgbg.apply(frame)
    25    211.0 MiB    -74.3 MiB        1393               color_mask = cv2.morphologyEx(fgmask, cv2.MORPH_OPEN, kernel)
    26
    27    211.9 MiB    384.0 MiB        1393               contourparts, _ = cv2.findContours(color_mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE) #used to
    28    211.9 MiB -24904.3 MiB       31341               contourparts = sorted(contourparts, key=lambda x:cv2.contourArea(x), reverse=True) #selects the
    29
    30    211.9 MiB   -804.0 MiB        1393               a=[] #used as a key to enter the if statement.
    31    211.9 MiB   -804.0 MiB        1393               xaxis_medium=0 #this variable is used as output to draw a vertical line passing through the cent
    32    211.9 MiB   -804.0 MiB        1393               yaxis_medium=0
    33
    34    211.9 MiB   -804.0 MiB        1393               if contourparts!=a:
    35    211.9 MiB   -796.9 MiB        1374                   for cnt in contourparts:
    36    211.9 MiB   -796.3 MiB        1374                       (leftmost_end, bottommost_end, width, height) = cv2.boundingRect(cnt) #to get the dimen
    37    211.9 MiB   -796.3 MiB        1374                       xaxis_medium = int((leftmost_end +(leftmost_end+width)) /2) #to get the x_axis value of
    38    211.9 MiB   -796.3 MiB        1374                       yaxis_medium = int((bottommost_end + (bottommost_end+height)) /2)
    39    211.9 MiB   -796.3 MiB        1374                       break
    40
    41    211.9 MiB   -803.4 MiB        1393               e2 = cv2.getTickCount()
    42    211.9 MiB   -804.0 MiB        1393               t = (e2 - e1)/cv2.getTickFrequency()
    43    211.9 MiB   -804.0 MiB        1394               while printt==True:
    44    194.9 MiB      0.0 MiB           1                   print("The time of execution of this function:{}".format(t))
    45    194.9 MiB      0.0 MiB           1                   end=time.time()
    46    194.9 MiB      0.0 MiB           1                   print('Time consumed in seconds is {}'.format(end-start))
    47    194.9 MiB      0.0 MiB           1                   printt=False
    48    211.9 MiB   -804.0 MiB        1393               totaltime=totaltime+t
    49
    50    211.9 MiB   -804.0 MiB        1393               cv2.line(frame,(xaxis_medium,0),(xaxis_medium,1000),(0,255,0),2) #used to draw a vertical line p
    51    211.9 MiB   -804.0 MiB        1393               cv2.line(frame,(0,yaxis_medium),(1700,yaxis_medium),(0,255,0),2)
    52    211.9 MiB   -801.3 MiB        1393               cv2.imshow("Frame",frame)
    53    211.9 MiB   -801.3 MiB        1393               cv2.imshow("Frame2",color_mask)
    54
    55
    56    211.9 MiB   -804.0 MiB        1393               key = cv2.waitKey(1) #used to terminate the program by pressing 'esc' key
    57    211.9 MiB   -804.0 MiB        1393               if key==27:
    58    211.0 MiB     -0.9 MiB           1                   break
    59    211.0 MiB      0.0 MiB           1       print(totaltime)
    60    211.0 MiB      0.0 MiB           1       print(countt)
    61    211.0 MiB      0.0 MiB           1       print(totaltime/countt)
    62
    63    211.0 MiB      0.0 MiB           1       print(cv2.useOptimized()) #checks if OpenCV is running optimized code or not
    64    174.0 MiB    -37.1 MiB           1       capture.release()
    65    168.7 MiB     -5.3 MiB           1       cv2.destroyAllWindows()
```