Computer Vision

LAB 5

Submitted By:

Gaurav Agarwal	BT17CSE001
Abhibha Gupta	BT17CSE020
Thogaru Himabindu	BT17CSE056

Submitted to:

Dr. Mayur Parate

Task:

TRACK AN OBJECT BY ESTIMATING ITS MOTION IN VIDEO SEQUENCE

भारतीय सूचना प्रौद्योगिकी संस्थान, नागपूर

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, NAGPUR

(An Institution of National Importance by Act of Parliament)

BSNL RTTC, Near TV Tower, Besides Balaji Temple, Seminary Hills, Nagpur-440006

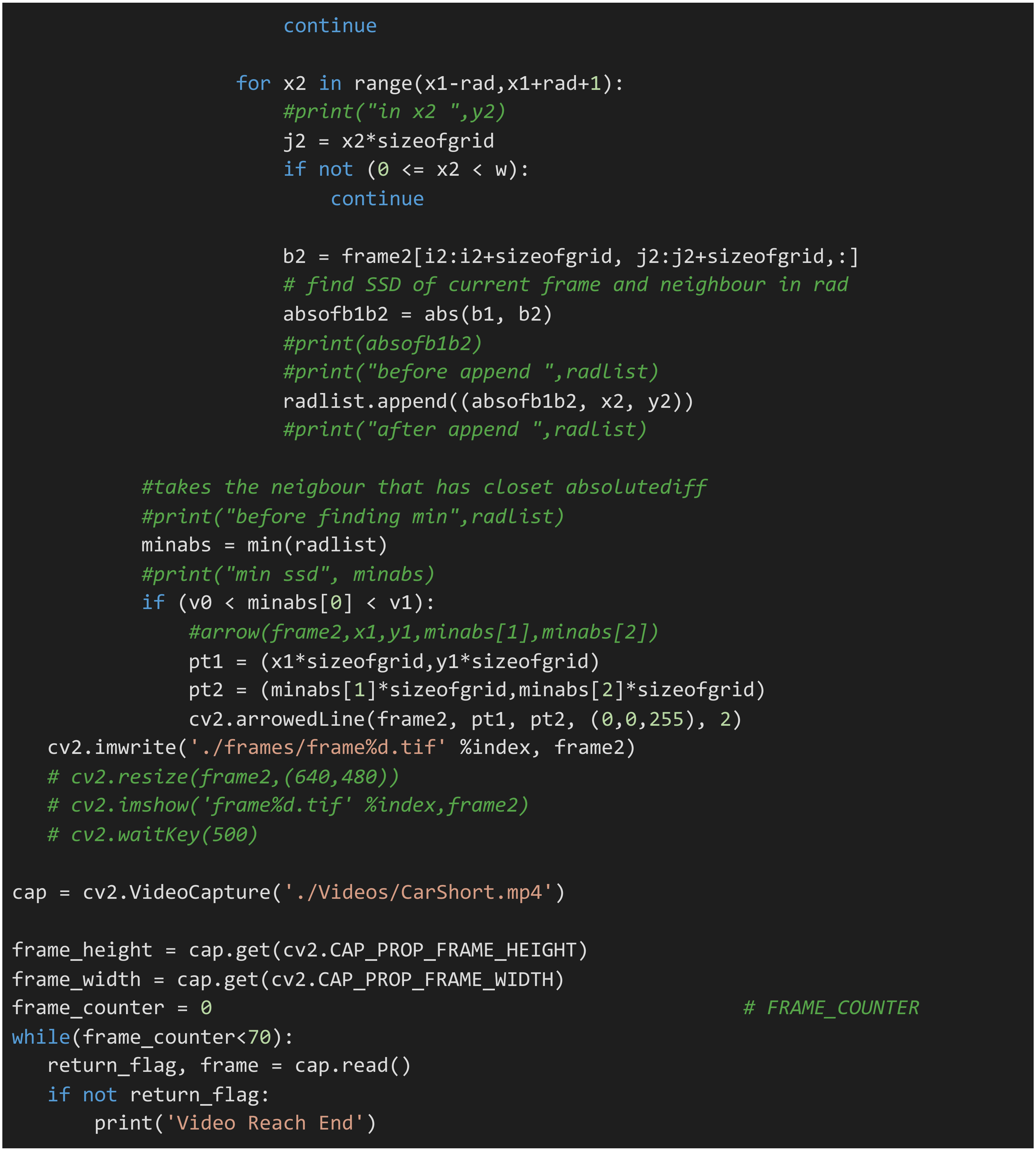
- To track the motion, we wrote the code from scratch where we take the video frames and we consider a grid of specific size and use the blocks and find the startpoint and appended location of it (whose difference is more than a threshold value) and draw an arrowed line on the appended frame from the start point to the appended point. Then we merge all the frames with arrows drawn to output video.
- Open CV functions used:
    - cv2.arrowedLine(frame,image, start_point, end_point, color, thickness)
    - cv2.imwrite() and cv2.imshow()

- The Block matching algorithm did not use any prebuilt methods of opencv.

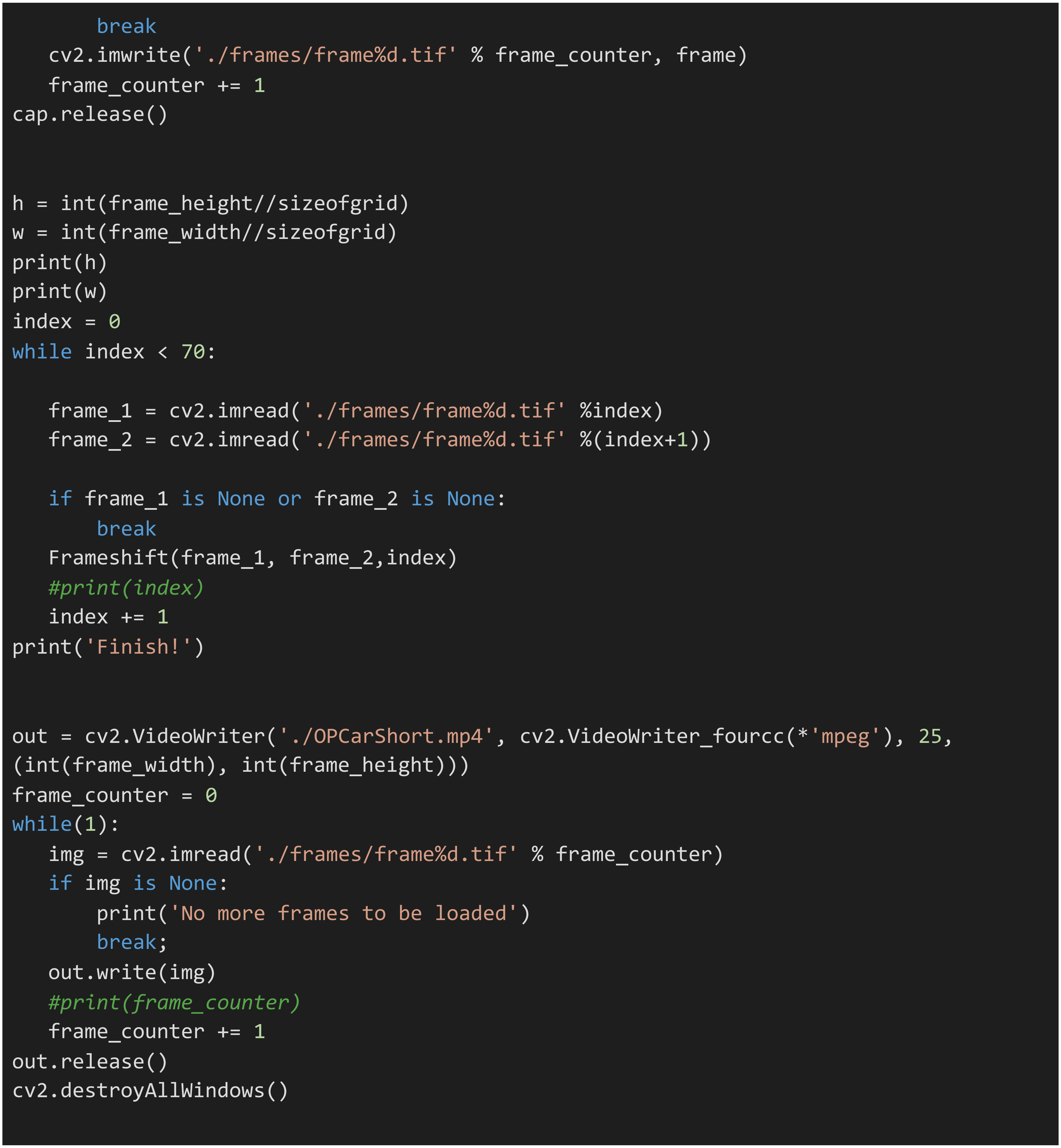
It was written from scratch

Code and output are shown below:

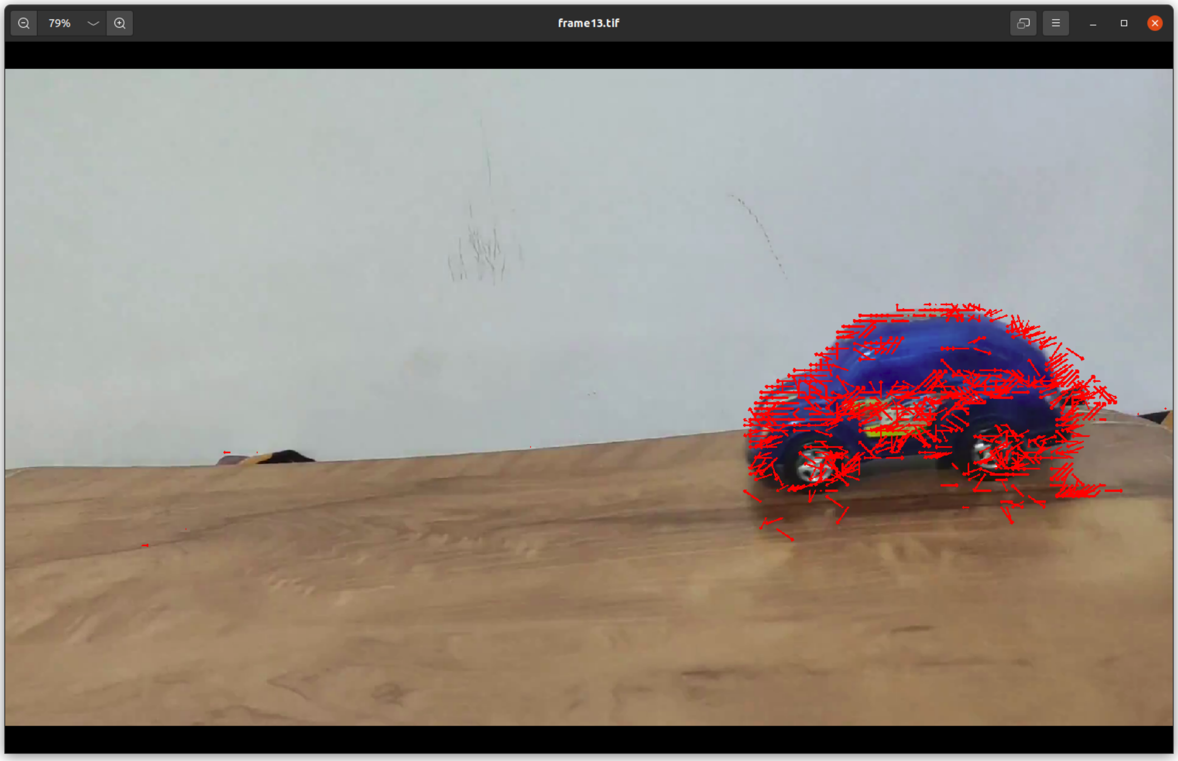
```
import cv2
sizeofgrid = 9
rad= 3
v0 = 100
v1 = 150
def Frameshift(frame1 , frame2, index):
   for y1 in range(h):
       #print("in y1", y1)
       i = y1*sizeofgrid
       for x1 in range(w):
           #print("in x1")
           j = x1*sizeofgrid
           b1 = frame1[i:i+sizeofgrid, j:j+sizeofgrid,:]
           radlist = []

           for y2 in range(y1-rad,y1+rad+1):
                   #print("in y2 ",y2)
                   i2 = y2*sizeofgrid
                   if not (0 <= y2 < h):
```

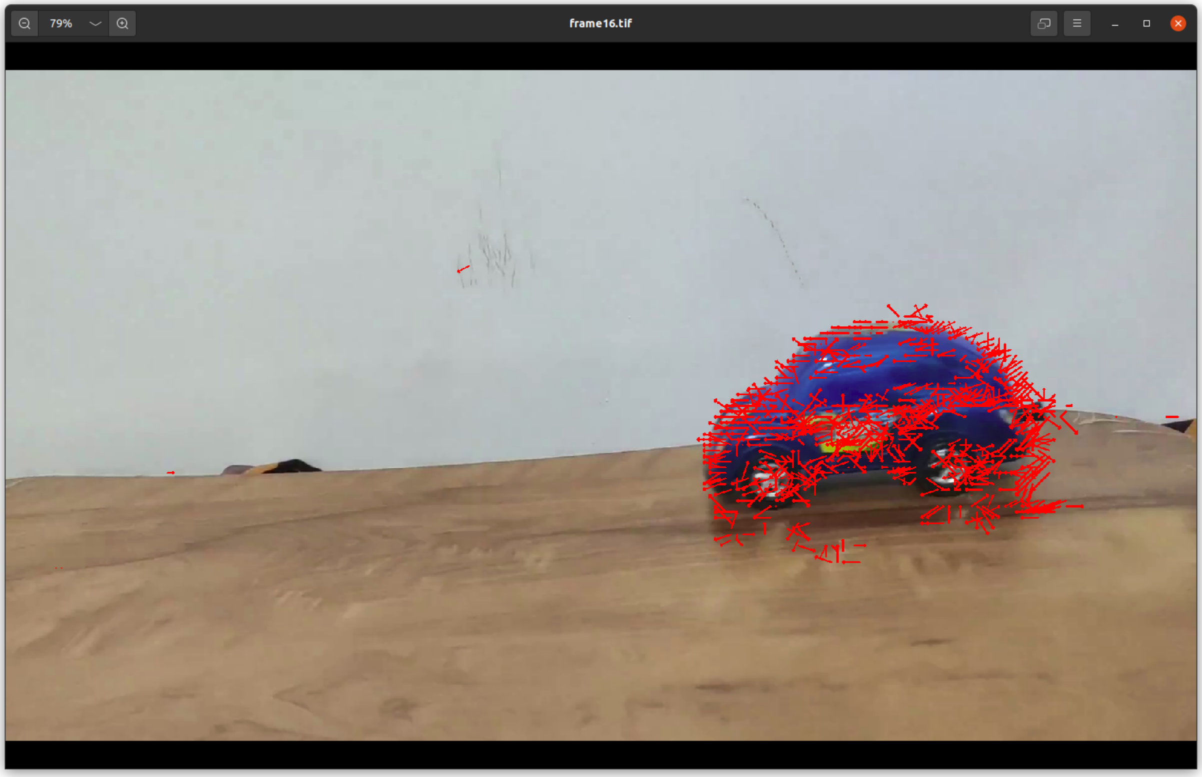


```
                        continue

                    for x2 in range(x1-rad,x1+rad+1):
                        #print("in x2 ",y2)
                        j2 = x2*sizeofgrid
                        if not (0 <= x2 < w):
                            continue

                        b2 = frame2[i2:i2+sizeofgrid, j2:j2+sizeofgrid,:]
                        # find SSD of current frame and neighbour in rad
                        absofb1b2 = abs(b1, b2)
                        #print(absofb1b2)
                        #print("before append ",radlist)
                        radlist.append((absofb1b2, x2, y2))
                        #print("after append ",radlist)

            #takes the neigbour that has closet absolutediff
            #print("before finding min",radlist)
            minabs = min(radlist)
            #print("min ssd", minabs)
            if (v0 < minabs[0] < v1):
                #arrow(frame2,x1,y1,minabs[1],minabs[2])
                pt1 = (x1*sizeofgrid,y1*sizeofgrid)
                pt2 = (minabs[1]*sizeofgrid,minabs[2]*sizeofgrid)
                cv2.arrowedLine(frame2, pt1, pt2, (0,0,255), 2)
    cv2.imwrite('./frames/frame%d.tif' %index, frame2)
    # cv2.resize(frame2,(640,480))
    # cv2.imshow('frame%d.tif' %index,frame2)
    # cv2.waitKey(500)

cap = cv2.VideoCapture('./Videos/CarShort.mp4')

frame_height = cap.get(cv2.CAP_PROP_FRAME_HEIGHT)
frame_width = cap.get(cv2.CAP_PROP_FRAME_WIDTH)
frame_counter = 0                                             # FRAME_COUNTER
while(frame_counter<70):
    return_flag, frame = cap.read()
    if not return_flag:
        print('Video Reach End')
```



```
        break
    cv2.imwrite('./frames/frame%d.tif' % frame_counter, frame)
    frame_counter += 1
cap.release()


h = int(frame_height//sizeofgrid)
w = int(frame_width//sizeofgrid)
print(h)
print(w)
index = 0
while index < 70:

    frame_1 = cv2.imread('./frames/frame%d.tif' %index)
    frame_2 = cv2.imread('./frames/frame%d.tif' %(index+1))

    if frame_1 is None or frame_2 is None:
        break
    Frameshift(frame_1, frame_2,index)
    #print(index)
    index += 1
print('Finish!')


out = cv2.VideoWriter('./OPCarShort.mp4', cv2.VideoWriter_fourcc(*'mpeg'), 25,
(int(frame_width), int(frame_height)))
frame_counter = 0
while(1):
    img = cv2.imread('./frames/frame%d.tif' % frame_counter)
    if img is None:
        print('No more frames to be loaded')
        break;
    out.write(img)
    #print(frame_counter)
    frame_counter += 1
out.release()
cv2.destroyAllWindows()
```





Input video and Output video are attached with this file
Some of the output frames are included here