Chapter 3

Working with Videos

This chapter deals with basics of video processing using Python 3.8.13. I hope you will like it.

Contents:

1. Accessing Camera

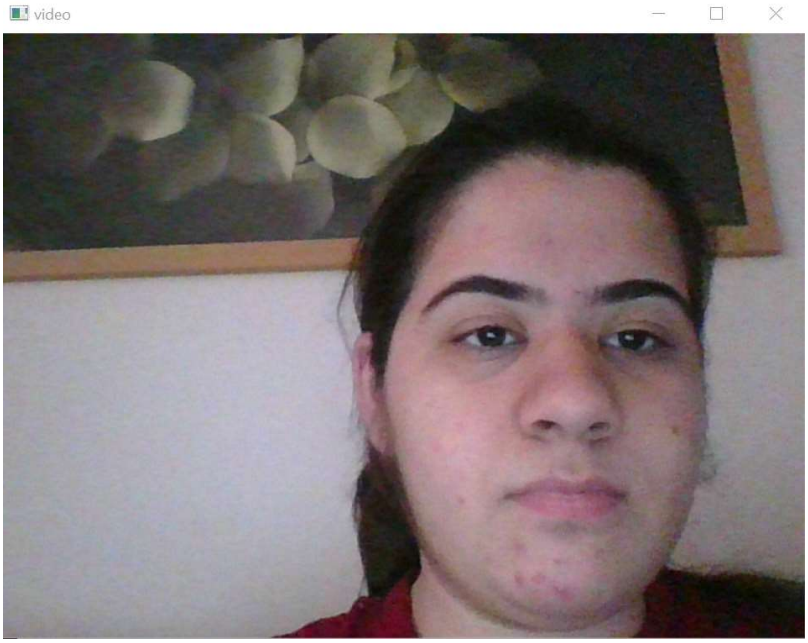
Write the following code for accessing camera and video frames:

Code:

Accessing Camera.py

```python
# Code to Access Camera

import cv2

cam = cv2.VideoCapture(0)

if not cam.isOpened():
    print("Cannot open camera")
    exit()

while True:
    # Capture frame-by-frame
    check, frame = cam.read()
    # if frame is read correctly ret is True
    if not check:
        print("Can't receive frame (stream end?). Exiting ...")
        break
    # Display the resulting frame
    cv2.imshow('video',frame)
    key = cv2.waitKey(1)
    if (key==27):
        break
# When everything done, release the capture
cam.release()
cv2.destroyAllWindows()
```

Output:



For More Information Please Refer To The Links Given Below:

https://www.geeksforgeeks.org/python-opencv-capture-video-from-camera/ (https://www.geeksforgeeks.org/python-opencv-capture-video-from-camera/)

https://docs.opencv.org/4.x/dd/d43/tutorial_py_video_display.html (https://docs.opencv.org/4.x/dd/d43/tutorial_py_video_display.html)

2. Basic Operations on Videos

Writing a video using OpenCV

While building applications, it becomes important to save demo videos of your work as well as many applications themselves might require saving a video clip. For example, in a surveiallance application, you might have to save a video clip as soon as you see something unusual happening.

In this section, we will describe how to save a video in **avi** and **mp4** formats using openCV.

Read Video from the Source

In [10]:
```
# import the library
import cv2
import matplotlib.pyplot as plt
%matplotlib inline


source = './race_car.mp4'  # source = 0 for webcam

cap = cv2.VideoCapture(source)
```

In [11]:
```
if (cap.isOpened()== False):
    print("Error opening video stream or file")
```

Read and Display One Frame

In [12]:
```
ret, frame = cap.read()
plt.imshow(frame[...,::-1])
plt.axis('off')
plt.show()
```

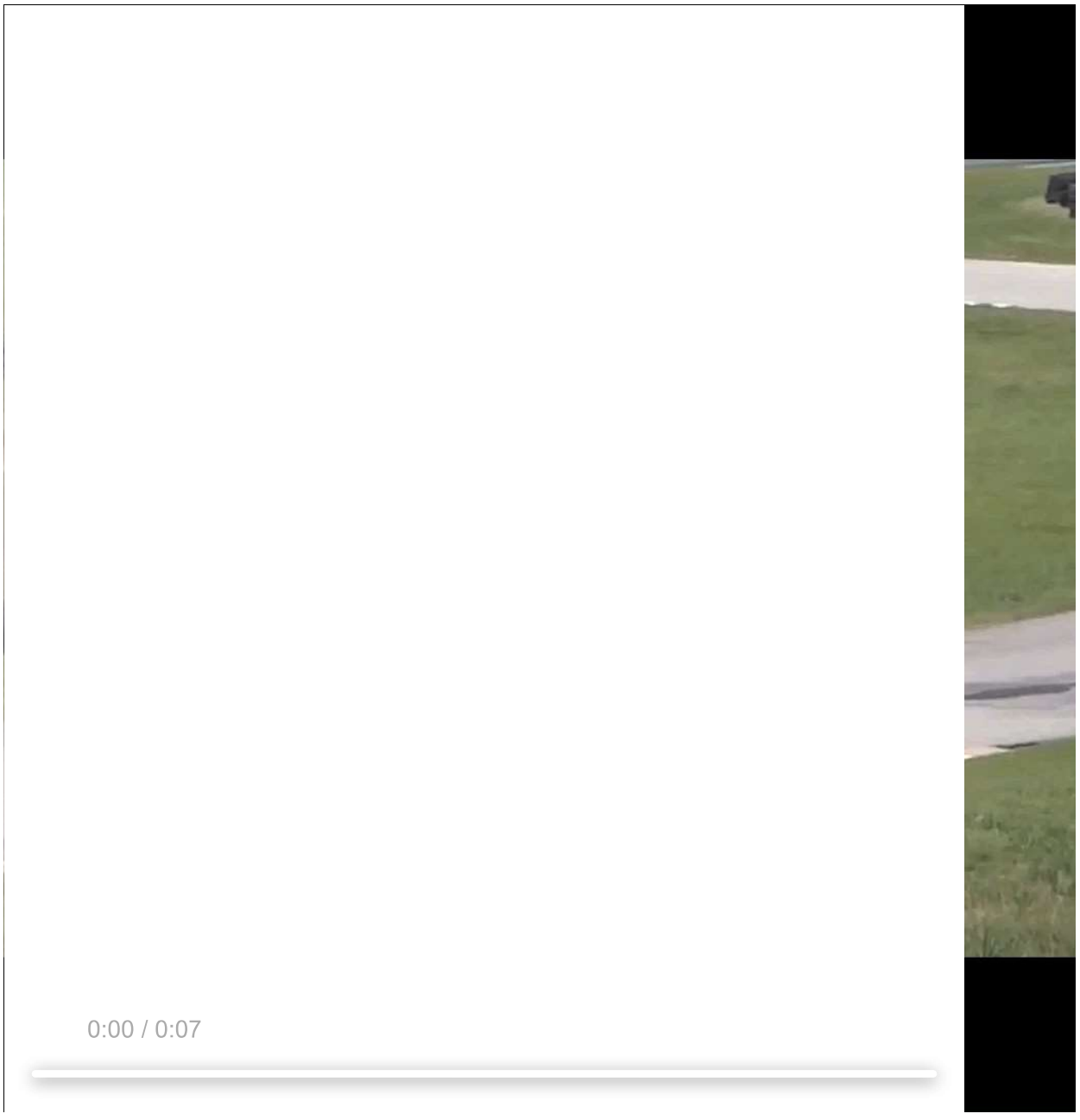
In [13]:
```
print(frame.shape)
```

```
(1080, 1728, 3)
```

Display the Video from File

In [14]:
```
from IPython.display import HTML
HTML("""
<video width=1024 controls>
  <source src="race_car.mp4" type="video/mp4">
</video>
""")
```

Out[14]:



Write Video using OpenCV

For writing the video, you need to create a videowriter object with the right parameters.

Function Syntax

```
VideoWriter object=   cv.VideoWriter( filename, fourcc, fps, frameSize
 )
```

where, **Parameters**

1. `filename` : Name of the output video file.
2. `fourcc` : 4-character code of codec used to compress the frames. For example, VideoWriter::fourcc('P','I','M','1') is a MPEG-1 codec, VideoWriter::fourcc('M','J','P','G') is a motion-jpeg codec etc. List of codes can be obtained at Video Codecs by FOURCC page. FFMPEG backend with MP4 container natively uses other values as fourcc code: see ObjectType, so you may receive a warning message from OpenCV about fourcc code conversion.
3. `fps` : Framerate of the created video stream.
4. `frameSize` : Size of the video frames.

In [15]:
```
# Default resolutions of the frame are obtained.
# Convert the resolutions from float to integer.
frame_width = int(cap.get(3))
frame_height = int(cap.get(4))
# Define the codec and create VideoWriter object.
out_avi = cv2.VideoWriter('race_car_out.avi',cv2.VideoWriter_fourcc('M','J','P','
out_mp4 = cv2.VideoWriter('race_car_out.mp4',cv2.VideoWriter_fourcc(*'XVID'), 10,
```

Read Frames and Write to File

We will read the frames from the race-car video and write the same to the two objects we created in the previous step. We should release the objects after the task is complete.

In [16]:
```
# Read until video is completed
while(cap.isOpened()):
    # Capture frame-by-frame
    ret, frame = cap.read()
    if (ret == True):
        # Write the frame to the output files
        out_avi.write(frame)
        out_mp4.write(frame)
    # Break the loop
    else:
        break
```

In [17]:
```
# When everything done, release the VideoCapture and VideoWriter objects
cap.release()
out_avi.release()
out_mp4.release()
```

The End of Chapter 3 !!!