Brishti Roy

25MML0027

Video processing Lab-2

import pandas as pd # Used for loading and manipulating CSV label data  
import numpy as np # Used for numerical operations  
import matplotlib.pyplot as plt # Used for plotting and displaying images  
from glob import glob # Used for file pattern matching (not heavily used here)  
import IPython.display as ipd # Used for displaying media in notebook environments  
from tqdm.notebook import tqdm # Used for displaying progress bars in loops

#import subprocess

ipd.Video("026c7465-309f6d33.mp4",width = 200)

<IPython.core.display.Video object>

Open the Video and Read Metadata

!pip install opencv-python

Requirement already satisfied: opencv-python in /usr/local/lib/python3.12/dist-packages (4.13.0.92)  
Requirement already satisfied: numpy>=2 in /usr/local/lib/python3.12/dist-packages (from opencv-python) (2.0.2)

# Load in video capture  
import cv2  
cap = cv2.VideoCapture('026c7465-309f6d33.mp4')

# Retrieves total number of frames in the video  
cap.get(cv2.CAP\_PROP\_FRAME\_COUNT)

2398.0

# Retrieves height and width of video frames  
height = cap.get(cv2.CAP\_PROP\_FRAME\_HEIGHT)  
width = cap.get(cv2.CAP\_PROP\_FRAME\_WIDTH)  
print(f'Height {height}, Width {width}')

Height 720.0, Width 1280.0

# Retrieves frames per second (FPS) of the video  
fps = cap.get(cv2.CAP\_PROP\_FPS)  
print(f'FPS : {fps:0.2f}')

FPS : 59.94

# Releases video capture resource after metadata extraction  
cap.release()

Pull the image from the first frame

# Reads one frame from the video  
# 'ret' is True if frame is successfully read  
  
cap = cv2.VideoCapture('026c7465-309f6d33.mp4')  
ret, img = cap.read()  
print(f'Returned {ret} and img of shape {img.shape}')

Returned True and img of shape (720, 1280, 3)

Loading the First Image

plt.imshow(img) #opencv loads color in BGR format, matplot lib expects RGB format

<matplotlib.image.AxesImage at 0x780495e4eb70>



## Helper function for plotting opencv images in notebook  
def display\_cv2\_img(img, figsize=(10, 10)):  
 img\_ = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)  
 fig, ax = plt.subplots(figsize=figsize)  
 ax.imshow(img\_)  
 ax.axis("off")

display\_cv2\_img(img)



cap.release()

Display Multiple Frames from the Video

import cv2  
fig, axs = plt.subplots(5, 5, figsize=(30, 20))  
axs = axs.flatten()  
  
cap = cv2.VideoCapture("026c7465-309f6d33.mp4")  
n\_frames = int(cap.get(cv2.CAP\_PROP\_FRAME\_COUNT))  
  
img\_idx = 0  
for frame in range(n\_frames):  
 ret, img = cap.read()  
 if ret == False:  
 break  
 if frame % 100 == 0: #frame is divisible by 100 - Note the modulo operator  
 axs[img\_idx].imshow(cv2.cvtColor(img, cv2.COLOR\_BGR2RGB))  
 axs[img\_idx].set\_title(f'Frame: {frame}')  
 axs[img\_idx].axis('off')  
 img\_idx += 1  
  
plt.tight\_layout() #automatically adjusts the spacing between the plots  
plt.show()  
cap.release()



labels = pd.read\_csv('mot\_labels.csv',low\_memory=False)  
labels.head()  
video\_labels = (labels.query('videoName == "026c7465-309f6d33"').reset\_index(drop=True).copy())  
video\_labels["video\_frame"] = (video\_labels["frameIndex"] \* 11.9).round().astype("int") #Video is at 60Hz but the labellers did only 5 Hz

print(video\_labels.shape)

(0, 14)

print(labels["category"].value\_counts())

category  
car 1141935  
pedestrian 195536  
truck 75712  
bus 30907  
bicycle 15278  
rider 10971  
other vehicle 10060  
motorcycle 6686  
trailer 2181  
other person 1654  
train 795  
Name: count, dtype: int64

cap = cv2.VideoCapture("026c7465-309f6d33.mp4")  
n\_frames = int(cap.get(cv2.CAP\_PROP\_FRAME\_COUNT))  
  
img\_idx = 0  
for frame in range(n\_frames):  
 ret, img = cap.read()  
 if ret == False:  
 break  
 if frame == 1035:  
 break  
cap.release()

display\_cv2\_img(img)

