

視訊串流與追蹤HW4
309553012 黃建洲

1. Experiment setup:

步驟1: webcam

webcam的部分我使用openCV提供的VideoCapture(0), 由於MP4Box的部分要使用到gpac套件, 而該套件在linux上使用較為方便, 故需要安裝虛擬機的extention package並將內置相機連接至虛擬機(以USB的方式)。

步驟2: object tracking

我的作法是在讀取完webcam後, 將讀取的畫面寫入avi檔, 累積數個frame之後統一將avi檔送進上次作業實作的yolov3 + sort的object tracking模型, 將所有影格轉換為經過tracking的影格。

步驟3: ffmpeg

利用套件ffmpeg-python, 可以直接將轉換後的avi檔轉碼為mp4檔案

步驟4: MP4Box

在這次作業中我在虛擬機安裝了gpac的套件, 並在ffmpeg轉碼結束後用subprocess在shell中使用gpac提供的MP4Box功能將MP4檔案轉為mpd檔以及一個fragment mp4

步驟5: HTML

在專案資料夾中建立一個index.html, 根據各方資料建立出一個videoplayer後, 遇到了CORS的問題, 導致影片不能被html檔所用。在這部分我使用Node.js在專案資料夾建立了虛擬伺服器解決這個問題。

2. Briefly explain your code:

步驟1:

```
cap = cv2.VideoCapture(0)
```

```

# Capture frame-by-frame
ret, frame = cap.read()

# Our operations on the frame come here
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

# Display the resulting frame
cv2.imshow('frame',gray)
#cv2.imwrite("output/frame-{}.jpg".format(frameIndex), frame)
print(frameIndex)
if frameIndex > 1000:
    break
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

if writer is None:
    # initialize our video writer
    fourcc = cv2.VideoWriter_fourcc(*"MJPG")
    writer = cv2.VideoWriter("input.avi", fourcc, 30,
        (frame.shape[1], frame.shape[0]), True)

    # some information on processing single frame

    #if total > 0:
    #    elap = (end - start)
    #    #print("[INFO] single frame took {:.4f} seconds".format(elap))
    #    #print("[INFO] estimated total time to finish: {:.4f}".format(
    #        elap * total))

# write the output frame to disk
writer.write(frame)

# increase frame index
frameIndex = frameIndex + 1

```

將網路攝像頭的影像擷取下來，並逐幀存成一個avi檔。

步驟2同上次作業，不再描述。

步驟3:

```

stream = ffmpeg.input('output.avi')
stream = ffmpeg.output(stream, 'output.mp4')
stream = ffmpeg.overwrite_output(stream)

```

利用ffmpeg將步驟2輸出的影片檔案轉換為mp4格式。

步驟4:

```

subprocess.call(["MP4Box", "-dash", "10", "output.mp4"], shell=False)

```

呼叫shell使用MP4Box指令，將mp4檔案轉換為mpeg dash格式。

步驟5:

```

<!doctype html>
<html>
<head>
<title>DASH</title>
<script src="https://cdn.dashjs.org/latest/dash.all.min.js"></script>
</head>
<body>
<video id="videoPlayer" autoplay="true" controls="true" muted="muted"></video>
<script>
    (function() {
        var url = "output_dash.mpd";
        var player = dashjs.MediaPlayer().create();
        player.initialize(document.querySelector("#videoPlayer"), url, true);
    })();
</script>
</body>
</html>

```

利用dash.js建立網頁的mpeg dash格式影片撥放器。

3. Contribution:

黃建洲: 100%

3. discussion:

這次作業讓我們實作了MPEG dash，由於沒有找到組員以及期末專題眾多，僅將基本功能做出，並且在效能上還有待加強。這次我認為我遇到最難的問題在MP4Box的部份以及CORS的問題。由於我一開始使用的環境為conda以及windows的ubuntu subsystem，前者卡在gpac不支援，後者則卡在webcam無法掛接，因此最後使用虛擬機進行。CORS的問題則是在查詢了很久之後才發現有Node.js的方法