

source code : https://github.com/Aeocp/Traffic_counting

รันผ่าน : <https://colab.research.google.com/drive/19HcjuVgkj1mLdxrmZbeLpolvfywaJtEX?usp=sharing>

วิธีการรันโค้ด Traffic Counting

- เข้า <https://colab.research.google.com/drive/19HcjuVgkj1mLdxrmZbeLpolvfywaJtEX?usp=sharing>

Step1

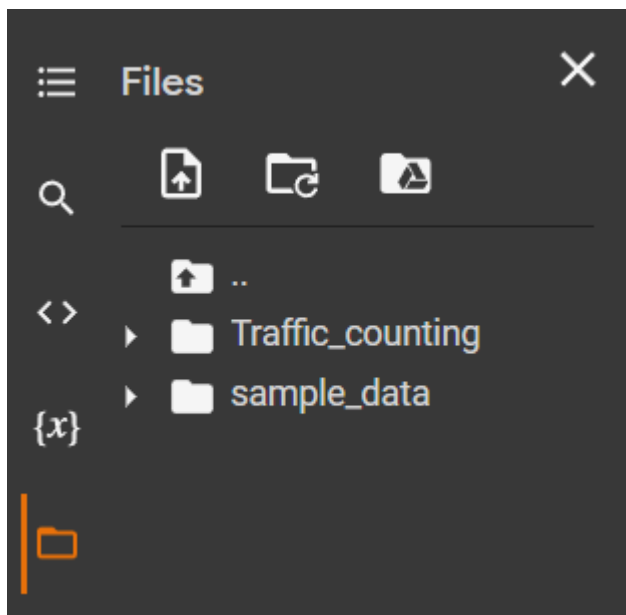
1. รัน 2 โค้ดดังภาพด้านล่างเพื่อ clone code จาก github และดาวน์โหลด requirements

```
Step 1 : Cloning code for Github and Google drive

[1] !git clone https://github.com/Aeocp/Traffic_counting

[3] %cd /content/Traffic_counting/
    !pip install -r requirements.txt
```

จะได้ folder ที่มีชื่อว่า “Traffic_counting” ดังภาพ



2. นำเข้าข้อมูล (วิดีโอและข้อมูลที่ได้จาก Yolo)

เนื่องจากใน github ลงคลิปวิดีโอในขนาดที่จำกัดจึงต้องดึงข้อมูลจาก google drive โดยสามารถโหลดตัวอย่างได้จาก https://www.dropbox.com/sh/26yoftfgxx5hcbr/AADCoZmdosS1-ZgMHB_ac6q3a?dl=0



```
%cd /content/  
from google.colab import drive  
drive.mount('/content/drive')
```

Step2A และ Step2B

เป็นการหา input เพื่อกำหนดตำแหน่งของเส้นนับรถบนถนน โดย

Step2A : เมื่อใช้ตัวอย่างวิดีโอที่มีอยู่หรือมุมกล้องตัวเดียวกับที่มีอยู่

เช่น หากต้องการ test ไฟล์ hlp-01-20210610-170005-cut.mp4

กล้องมุมนี้จะมีถนนที่วัดได้ 2 เส้น ซึ่งมี input เป็น

0.05,0.6,0.2,0.65

และ

0.2,0.8,0.6,0.9

ซึ่ง input ทั้งสองชุดนี้จะนำไปใช้ใน Step 3

Step2B : เมื่อใช้มุมกล้องอื่นๆนอกเหนือจากตัวอย่าง

1. แกะไขบรรทัดบริเวณสี่เหลี่ยม

```
vidcap = cv2.VideoCapture("/content/Traffic_counting/video/hlp/080841-03.mp4")
```

ให้เป็น path ของวิดีโอที่ต้องการหาตำแหน่งเส้น

2. รันโค้ดด้านล่าง

```
[ ] %cd /content/
import cv2
from google.colab.patches import cv2_imshow
vidcap = cv2.VideoCapture("/content/Traffic_counting/video/hlp/080841-03.mp4")
success,image = vidcap.read()
frameY = image.shape[0]
frameX = image.shape[1]
numLine = int(input("Enter number of counting line :"))
x = []
y = []
s_line = []
l = 0
print("Enter the beginning and ending of the line (0-1) (xb,yb,xe,ye) :")
for l in range(numLine):
    print("Line",l+1)
    Bp = input("Enter line position :").split(",")
    x.append((Bp[0]))
    y.append((Bp[1]))
    x.append((Bp[2]))
    y.append((Bp[3]))
    line = [(int(float(x[l*2]) * frameX), int(float(y[l*2]) * frameY)), (int(float(x[l*2+1]) * frameX), int(float(y[l*2+1]) * frameY))]
    cv2.line(image, line[0], line[1], (0, 0, 255), 2)
cv2.imwrite("pic_output.jpg", image)
img = cv2.imread("/content/pic_output.jpg")
cv2_imshow(img)
cv2.waitKey(0)
```

3. เมื่อรันแล้วจะต้องได้ input ซึ่งได้แก่จำนวนเส้นและตำแหน่งของเส้นนับจำนวนตามจำนวนเส้นที่กำหนด

* โดย input ตำแหน่งเส้นจะอยู่ในรูปแบบของ (x1,y1,x2,y2) โดยที่ x,y จะอยู่ระหว่าง 0-1

```
/content
Enter number of counting line :2
Enter the beginning and ending of the line (0-1) (xb,yb,xe,ye) :
Line 1
Enter line position :0.05,0.6,0.2,0.65
Line 2
Enter line position :0.1,0.2,0.3,0.4
```

จะได้ผล

เป็นภาพที่มีเส้นสีแดงปรากฏ(ดังภาพตัวอย่าง) เราจะต้องปรับเปลี่ยนตัวเลขเพื่อให้ได้ตำแหน่งของเส้นนับจำนวน
รถที่เหมาะสม และจดบันทึกแยกเอาไว้



Step3 การรันโปรแกรม

1. คล้ายกับวิธีการใน **Step2B** ในคำสั่ง

```
!python main.py
```

```
--video /content/drive/MyDrive/test/hlp-01-20210610-170005-cut.mp4
```

```
--text /content/drive/MyDrive/test/hlp-01-20210610-170005-cut.mp4.txt
```

```
--output /content/hlp-01.avi
```

เราจะต้องเปลี่ยน path ของไฟล์วิดีโอและtxtที่ได้จาก yolo ให้เป็น path ที่ต้องการ รวมถึงสามารถเปลี่ยนตำแหน่งที่จะเก็บวิดีโอ output และชื่อได้อีกด้วย

```
!python main.py
```

```
--video โฟลเดอร์ที่ไฟล์วิดีโออยู่
```

```
--text โฟลเดอร์ที่ไฟล์outputที่ได้จากYolo(.txt)
```

```
--output โฟลเดอร์ที่ต้องการให้แสดงวิดีโอ
```

ตัวอย่าง

```
!python main.py --video /content/Traffic_counting/video/hlp/080841-03.mp4 --text
```

```
/content/Traffic_counting/input_txt/hlp/080841-03.txt --output /content/Traffic_counting/080841-03-s1.avi
```

2. เมื่อรันโค้ดแล้วจะต้องใส่ input ซึ่งได้แก่จำนวนเส้นและตำแหน่งของเส้นนับจำนวนตามจำนวนเส้นที่กำหนด

* โดย input ตำแหน่งเส้นจะอยู่ในรูปแบบของ (x1,y1,x2,y2) โดยที่ x,y จะอยู่ระหว่าง 0-1 ซึ่งได้มาจาก

Step2A หรือ Step2B

```
[ ] %cd /content/Traffic_counting/
```

```
[ ] !python main.py --video /content/drive/MyDrive/test/hlp-01-20210610-170005-cut.mp4 --text /content/drive/MyDrive/test/hlp-01-20210610-170005-cut.mp4.txt --output /cd
```

3. รอนรันผลเสร็จจะได้บรรทัดท้ายๆมาดังนี้

```
Total 1 : 30
frame 1480
Total 1 : 30
frame 1481
Total 1 : 30
frame 1482
Total 1 : 30
frame 1483
Total 1 : 30
frame 1484
Total 1 : 30
frame 1485
Total 1 : 30
frame 1486
imutils FPS: 9.261583426866393
```

ซึ่งหมายถึงนับรถได้ 30 คัน

Step4 Show Video

1. แก้ไขโค้ด %cd /content/ ให้เป็นตำแหน่ง path ที่ iutput อยู่
2. แก้ชื่อใน !ffmpeg -i hlp-01.avi output.mp4 ให้เป็นชื่อไฟล์ iutput ที่ตั้งไว้

จะแสดงวิดีโอผลลัพธ์ที่ได้ขึ้นมา และสามารถดาวน์โหลดผลลัพธ์ได้จากไฟล์ output.mp4

เพิ่มเติม การปรับค่า theshold สำหรับการเว้นเฟรม

** โค้ดปัจจุบันถูกตั้งค่า theshold เป็นแบบ เว้นสองเฟรม (10 frame/sec) และอ่านค่าแบบเว้นสองเฟรม
หากต้องการแก้ไขสามารถแก้ไขโดย*

1. การเว้นเฟรม : /content/Traffic_counting/main.py

```
* บรรทัดที่ 118 : if(frame_index%3 == 1):
    - if(frame_index%1 == 0) : ไม่เว้นเฟรม
    - if(frame_index%2 == 1) : เว้นเฟรมหนึ่งเฟรม
    - if(frame_index%3 == 0) : เว้นเฟรมสองเฟรม
    - if(frame_index%4 == 0) : เว้นเฟรมสามเฟรม
```

2. ปรับค่า threshold :

* กรณีไม่เว้นเฟรม แก้ไขดังนี้

/content/Traffic_counting/main.py

- บรรทัดที่ 40 : `max_cosine_distance = 0.3`
- บรรทัดที่ 42 : `nms_max_overlap = 1.0`

/content/Traffic_counting/deep_sort/tracker.py

- บรรทัดที่ 37 : แก้
`max_iou_distance= 0.7`
`max_age = 30`
`n_init = 3`
`adc_threshold = 0.5`

* กรณี เว้นเฟรมหนึ่งเฟรม แก้ไขดังนี้

/content/Traffic_counting/main.py

- บรรทัดที่ 40 : `max_cosine_distance = 0.3`
- บรรทัดที่ 42 : `nms_max_overlap = 1.0`

/content/Traffic_counting/deep_sort/tracker.py

- บรรทัดที่ 37 : แก้
`max_iou_distance= 0.95`
`max_age = 50`
`n_init = 3`
`adc_threshold = 0.5`

* กรณี เว้นเฟรม**สอง**เฟรม แก้ไขดังนี้

/content/Traffic_counting/main.py

- บรรทัดที่ 40 : $\text{max_cosine_distance} = 0.75$
- บรรทัดที่ 42 : $\text{nms_max_overlap} = 1.0$

/content/Traffic_counting/deep_sort/tracker.py

- บรรทัดที่ 37 : แก้
 $\text{max_iou_distance}=0.95$
 $\text{max_age} = 50$
 $\text{n_init} = 1$
 $\text{adc_threshold} = 0.5$

* กรณี เว้นเฟรม**สาม**เฟรม แก้ไขดังนี้

/content/Traffic_counting/main.py

- บรรทัดที่ 40 : $\text{max_cosine_distance} = 1.3$
- บรรทัดที่ 42 : $\text{nms_max_overlap} = 1.0$

/content/Traffic_counting/deep_sort/tracker.py

- บรรทัดที่ 37 : แก้
 $\text{max_iou_distance}= 0.95$
 $\text{max_age} = 50$
 $\text{n_init} = 1$
 $\text{adc_threshold} = 0.5$