**國立清華大學**

**計算機視覺**

**Computer Vision**

一張含有 文字, 筆跡, 字型, 圓形 的圖片

自動產生的描述

**Homework 4**

**系所級:電子所二年級**

**學號:111063548**

**姓名:蕭方凱**

**指導老師:孫民教授**

**目錄**

[Problems 3](#_Toc152598265)

[1. Find the split train val path function in data/preprocess.py and write the code to generate the training YAML file using the Q1 dataset. 3](#_Toc152598266)

[(1) WAY1(random indices to control paths) 3](#_Toc152598267)

[(2) WAY2(sklearn.model\_selection) 5](#_Toc152598268)

[2. finish the select images function in data/preprocess.py to generate the training YAML file. 6](#_Toc152598269)

[(1) WAY1(random shuffle) 6](#_Toc152598270)

[(2) WAY2(equally distributed path) 7](#_Toc152598271)

[(3) WAY3(Distribute in proportion) 8](#_Toc152598272)

[3. In the final question, besides utilizing the dataset selected in Q2, you can incorporate another 1,200 unlabeled images. In the following rules, you can try any method to improve performance. 10](#_Toc152598273)

[Discussion 14](#_Toc152598274)

[1. Precision and Recall 14](#_Toc152598275)

[2. Q1 16](#_Toc152598276)

[3. Q2 17](#_Toc152598277)

[4. Q3 18](#_Toc152598278)

[(1) Freeze backbone 18](#_Toc152598279)

[(2) Positive weights 18](#_Toc152598280)

[(3) Focal loss 19](#_Toc152598281)

# Problems

## Find the split train val path function in data/preprocess.py and write the code to generate the training YAML file using the Q1 dataset.

在split\_train\_val\_path function中，分別使用兩種方法去將資料分成training data及validation data，下面分別介紹兩種方法及其對應結果。

### WAY1(random indices to control paths)

一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述

先將all\_image\_path的數量透過指令len求出，即num\_samples，接著將num\_samples乘上比重train\_val\_ratio，得到train\_image\_paths的數量，本次作業num\_samples固定為200，假設乘上train\_val\_ratio = 0.9，最終則會將180個paths丟入training\_image\_paths，其餘20筆paths則丟入validation\_image\_paths。

Indicies部分是將all\_image\_paths順序打散用，先使用np.arange產生一0-199的一維陣列([0 1 2 … 198 199])，再用rng.shuufle將順序打亂，並按照前面算好的比重數量分配給train\_indicies和val\_indicies，最後將打亂後的順序分別對應到all\_image\_paths，產生隨機順序的train\_image\_paths及val\_image\_paths。

|  |  |  |
| --- | --- | --- |
|  | **Train result** | **Test result** |
| **Ratio = 0.7** | 一張含有 文字, 圖表, 行, 繪圖 的圖片  自動產生的描述 | 一張含有 圖表, 行, 繪圖, 斜率、斜坡 的圖片  自動產生的描述 |
| **Ratio = 0.8** | 一張含有 文字, 圖表, 行, 繪圖 的圖片  自動產生的描述 | 一張含有 圖表, 行, 繪圖 的圖片  自動產生的描述 |
| **Ratio = 0.9** | 一張含有 文字, 圖表, 行, 繪圖 的圖片  自動產生的描述 | 一張含有 圖表, 行, 繪圖 的圖片  自動產生的描述 |

上表為三個ratio下的PR Curve result，左邊column為train result，右邊column為test result，在ratio=0.8時，test result有最好的表現，mAP@0.5為0.632。

### WAY2(sklearn.model\_selection)

一張含有 文字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述

引用sklearn.metrics的train\_test\_split功能，輸入all\_image\_paths，random\_state=777是為了始終產生相同的隨機分割， train\_size則直接定義ratio，會將all\_image\_path內的path數量乘ratio並將此數量的path指定給train\_image\_paths，其餘指定給val\_image\_paths，suffle=True則與WAY1一樣達到打亂path順序的效果。

|  |  |  |
| --- | --- | --- |
|  | **Train result** | **Test result** |
| **Ratio = 0.7** |  |  |
| **Ratio = 0.8** |  |  |
| **Ratio = 0.9** |  |  |

上表為三個ratio下的PR Curve result，左邊column為train result，右邊column為test result，在ratio=0.9時，test result有最好的表現，mAP@0.5為0.617。

**總結WAY1及WAY2，使用WAY1去split train validation path，ratio設0.8，會有最好的test result。**

## finish the select images function in data/preprocess.py to generate the training YAML file.

在select images function中，使用三種方式去對Q2 datasets進行挑選，Q2 datasets共有6個資料夾(170, 173, 398, 410, 595, 511)，每個資料夾有200張照片，共1200張。如何從1200張照片中選擇200張，以下介紹三種方法及其分別對應的結果。

### WAY1(random shuffle)

一張含有 螢幕擷取畫面, 字型, 文字, 行 的圖片

自動產生的描述

輸入image\_paths，這裡為Q2六個資料夾所有的paths，並直接打亂image\_paths的順序，打亂之後取前200個paths(images\_num=200)，並將每個資料夾提供的paths數量print出來，如下:

一張含有 文字, 字型, 螢幕擷取畫面, 黑與白 的圖片

自動產生的描述

每個資料夾提供的paths數量不相等，為隨機取出，共200個selected\_image\_paths，而train.txt數量為160，val.txt數量為40是因為在split\_train\_val\_path function中train\_val\_ ratio為0.8。

### WAY2(equally distributed path)

一張含有 文字, 螢幕擷取畫面, 字型, 軟體 的圖片

自動產生的描述

先將image\_paths全部輸入到folder\_paths，folder\_path是一個dict的形式，有6個key分別為Q2 datsets六個資料夾路徑，每個key對應到的200個value為每個資料夾內中的image\_paths。

遍歷folder paths中的key和value，將每個key隨機挑選33個value出來，意即將六個資料夾中，每個資料夾隨機挑選33個image\_paths出來，並加入進selected\_image\_paths，最後第46行代表從六個資料夾中隨機選擇兩個並個抓一個image\_patht加入至selected\_image\_paths。

總結來說此方法做到在6個資料夾中盡量取出平均數量，且是隨機的，不會特別偏袒某個資料夾。Print出的結果如下:

一張含有 文字, 字型, 螢幕擷取畫面, 黑與白 的圖片

自動產生的描述

### WAY3(Distribute in proportion)

一張含有 文字, 螢幕擷取畫面, 字型, 功能表 的圖片

自動產生的描述

此方法先對Q2中六個資料夾進行訓練及測試，並對其結果表現進行排序，實測結果表現由好到壞為:495→398→173→410→511→170，將這六個資料夾按照比例提供不等的path數量，我採用比例0.9去按照比例分配path數量，假設495提供x個path，計算公式為:x + 0.9x + 0.81x + 0.73x + 0.63x + 0.55x = 200，x算出來約為44，所以從資料夾495→398→173→410→511→170，分別提供44、39、35、31、27、24筆image\_paths，print出的結果如下:

一張含有 文字, 字型, 螢幕擷取畫面, 黑與白 的圖片

自動產生的描述

|  |  |  |
| --- | --- | --- |
| **Method** | **Train result** | **Test result** |
| WAY1(random select) | 一張含有 文字, 圖表, 行, 地圖 的圖片  自動產生的描述 | 一張含有 圖表, 行, 繪圖 的圖片  自動產生的描述 |
| WAY2(equally distributed) | 一張含有 文字, 圖表, 行, 繪圖 的圖片  自動產生的描述 | 一張含有 圖表, 行, 繪圖, 文字 的圖片  自動產生的描述 |
| WAY3(distributed in proportion) | 一張含有 文字, 圖表, 行, 繪圖 的圖片  自動產生的描述 | 一張含有 圖表, 行, 繪圖, 文字 的圖片  自動產生的描述 |

在WAY1, WAY2, WAY3三種方法中，我實際各測試了三次，上圖是各取最佳的表現，test result的mAP @0.5由way1, way2, way3依序為0.759、0.754、0.75，way1(random select)表現最好。

造成此結果的原因，我推測是因random select方法透過隨機選取200張，造成其test result最不穩定，變動幅度很大，偶爾會出現比其他方法還要好的結果，但也會出現低於平均值的表現。way2採平均選取，每個資料夾選取33或34張，此方法的表現最穩定，估計是因為每個資料夾貢獻的量都差不多所導致。Way3採比例分配，先獨立選取每個資料夾的200張照片作為selected images去觀察每個資料夾的表現，讓表現最好的資料夾貢獻最多照片，表現最差的資料夾貢獻最少照片，並按照比例去分配每個資料夾的數量，六個資料夾加起來共200張照片，此方法下test result的變動幅度介於中間。

**雖然way1使用random方式擁有最高的mAP @0.5，但在選擇輸入資料時穩定度也是很重要的一部份，故本次作業採用way2去作為Q2及Q3的照片選擇。**

## In the final question, besides utilizing the dataset selected in Q2, you can incorporate another 1,200 unlabeled images. In the following rules, you can try any method to improve performance.

在Q3中，除了Q2的selected\_images，還加入了Q3的照片，與Q2依樣有六個資料夾共1200張照片，但差別是這些照片皆為unlabeled，在選擇Q3照片時方法與選擇Q2時一樣，採用平均數量方法去選擇照片。

除Q2 selected images之外，還加入Q3的200張照片後，train.txt及val.txt如下圖Fig1、Fig2:

一張含有 文字, 紙張, 字型, 數字 的圖片

自動產生的描述

Fig 1 train.txt

一張含有 文字, 紙張, 字型, 數字 的圖片

自動產生的描述

Fig 2 val.txt

首先，執行沒有任何improvement method的code，純粹輸入400張照片，其中來自於Q3的200張照片沒有label。



Test result如下圖:

一張含有 圖表, 行, 繪圖 的圖片

自動產生的描述

Fig 3 no improvement method result

mAP@0.5遠低於Q1及Q2的test result，推測是因為輸入照片中有一半的照片是沒有label的狀態，接下來會嘗試透過設定freeze及weights，freeze設定為2，weights則是沿用Q2所產生的weights去優化訓練結果，以及調整train.py內的positive weight部分去優化訓練結果。

加入Q2 weights及freeze後的code:





Test result如下圖:

一張含有 圖表, 行, 繪圖, 文字 的圖片

自動產生的描述

Fig 4 with Q2 weights

mAP@0.5有顯卓的提升，從原本的0.57增加0.753，甚至比Q2的一些方法還要高，由此可見就算沒有每張照片都label還是能夠將其準確度、精確度等指標提高。

調整train.py內的positive weight:

一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述

更改部分

Test result如下圖:

一張含有 圖表, 行, 繪圖 的圖片

自動產生的描述

Fig 5 調整positive weight code

# Discussion

## Precision and Recall

在機器學習中，評估模型好壞的常見指標有Accuracy、Precision、Recall 與 F1-Measure，每個指標評估的模型能力方向都有各自的意義及使用時機，在介紹這些指標前有幾個名詞(TP、TN、FP、FN)須先了解。

True Positive (TP): 預測為Positive且預測準確

True Negative (TN): 預測為Negative且預測準確

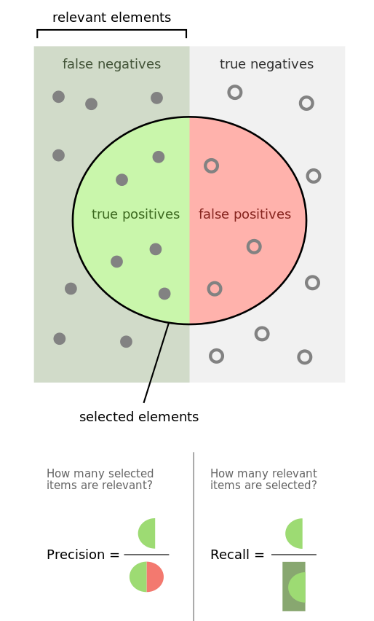
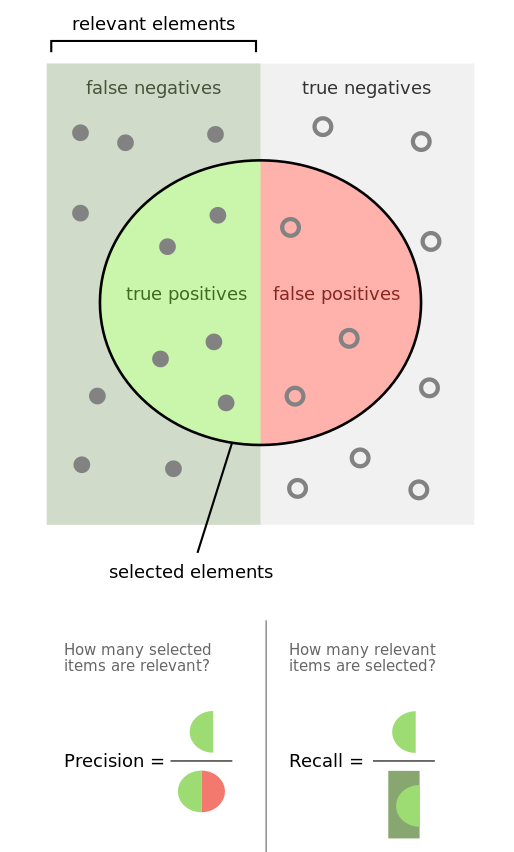
False Positive (FP): 預測為Positive但預測錯誤

False Negative (FN): 預測為Negative但預測錯誤

Accuracy為模型預測正確數量所佔整體的比例，即正確分類的樣本數與總樣本數之比。

Precision為精確率，這個指標在意的是在預測為 Positive 的結果中，其精確性是多少，即被預測為 Positive 的資料中，有多少是真的 Positive。

Recall為召回率，這個指標在意的是否觸及了所有的 Positive case，在所有positive樣本中(TP、FN)被準確預測的比例是多少。



由precision和recall可組成PR Curve，將模型評估視覺化，如下圖:

一張含有 圖表, 行, 繪圖, 文字 的圖片

自動產生的描述

右上角的mAP@0.5，0.5是IOU(Intersection Over Union)閥值，閥值越高代表對於準確度或精確度等指標要求更高，條件更嚴苛，示意圖如下圖:

一張含有 圖表, 文字, 螢幕擷取畫面, Rectangle 的圖片

自動產生的描述

由左至右，閥值越來越高，預測準確的難度更高，只有在非常確信的情況才會被視為預測正確，就會導致FP很少，FN很多，進而造成precision很高recall很低的狀況。從PR Curve大致可推斷P與R之間的關係，而mAP@0.5是閥值為0.5下，PR Curve(上方藍線)與Recall(X軸)所圍的面積，此數值越高越好。

## Q1

Q1是實現train\_val\_split功能，輸入的image path為兩個資料夾173、398，是由兩個不同視角的鏡頭所拍攝的畫面，兩資料夾共200張照片，在train\_val\_ratio為0.8時，有最佳test result(mAP@0.5最高)，推測原因是因為datasets輸入本身就來自兩個鏡頭而已，拍攝到車子的特徵不夠多元，若此時又使更多datasets當作training datasets，可能會導致overfitting，training result非常優秀，接近100%，但test result不盡理想。

不同train\_val\_ratio下的train/test mAP@0.5:

|  |  |  |
| --- | --- | --- |
| **train\_val\_ratio** | **Train mAP@0.5** | **Test mAP@0.5** |
| Ratio = 0.7 | 0.958 | 0.602 |
| Ratio = 0.8 | 0.952 | 0.632 |
| Ratio = 0.9 | 0.960 | 0.618 |

ratio = 0.9的training result雖然比ratio = 0.8優秀，但其testing result卻不是，可推斷是overfitting導致。

## Q2

Q2是實現selected\_images功能，Q2 dataset比起Q1多了許多資料夾，亦即擁有更多元視角下的車子照片，共六個資料夾(六個視角的鏡頭)，每個資料夾街友200張照片，Q2共計有1200張照片，Q2要實現的便是如何從1200張照片挑選200張照片作為訓練及驗證集。

我使用三種方法去實現照片選擇功能，分別為隨機選取、平均選取、按照資料夾有效程度進行比例分配，並搭配不同train\_val\_ratio去觀察結果，在train\_val\_ratio為0.9時有較佳的結果，與Q1結果不同，造成此差異的原因我認為是Q2 datasets的相機視角較多元，不會這麼容易造成overfitting的問題，理論上給予更多有效資料去做訓練確實也會提高預測模型的準確度。

Train\_val\_ratio = 0.9下，不同selected\_images的方式之對應結果(各別實測3次):

|  |  |  |
| --- | --- | --- |
| **Method** | **Train mAP@0.5** | **Test mAP@0.5** |
| WAY1(random select) | 0.872~0.882 | 0.724~0.759 |
| WAY2(equally distributed) | 0.87~0.96 | 0.733~0.754 |
| WAY3(distributed in proportion) | 0.805~0.927 | 0.7~0.75 |

## Q3

Q3中的400張照片包含Q2的200張labeled images以及Q3的200張unlabeled images，這雖然可以節省人力成本(不需要花時間去人工標籤)，但會對訓練模型造成影響，甚至導致訓練結果變差。

本次作業主要介紹幾種方式去優化訓練結果，positive weights、freeze backbone及focal loss，下面簡單說明三種方式及其目的。

1. Freeze backbone

主幹網絡通常指的是深度學習模型的前幾層，用於提取特徵。在訓練過程中，凍結主幹網絡可以防止其權重被更新，本次作業在train.py的code選擇沿用Q2所訓練的權重(best.pt)，為避免在訓練過程中此權重被更新，可使用freeze backbone，除了防止權重更新之外，還可增加訓練速度。

1. Positive weights

資料集的正樣本和負樣本數量不平衡，有時會影響模型的性能。透過賦予正樣本較大的權重，使模型更加關注對正樣本的準確預測，進而提高正樣本的召回率。

一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述

更改前的code計算了各個類別的權重，現在在計算完後進行歸一化，確保類別權重總和為1，這樣的好處在於平衡不同類別之間的影響，這樣每個類別的權重都在相同的比例範圍內，如果某些類別的權重過大，模型可能會更加關注那些權重大的類別，而對其他類別忽視。這可能導致模型的不穩定性，也可能因模型可能會傾向於過擬合該類別的訓練而造成過擬合的現象。

1. Focal loss

解決類別不平衡和困難樣本問題，將模型更集中地學習對難以分類的樣本進行準確預測，此方法主要針對交叉損失函數在處理大量易分類樣本時的問題，降低了對易分類樣本的關注，同時加強了對難以分類樣本的關注，這樣可以使模型更專注於那些難以區分的樣本，提高對這些樣本的預測準確性。