

深度學習於醫學影像 DLMI – Final Project

R12725024 資管碩一 蔡鈺驊

這次期末報告我延續之前作業的題目，使用 NeurIPS 在 2023 年於 Grand Challenge 舉辦的競賽當作練習。競賽名稱為 Foundation Model Prompting for Medical Image Classification (MedFM)，連結為 <https://medfm2023.grand-challenge.org>。

I. Introduction

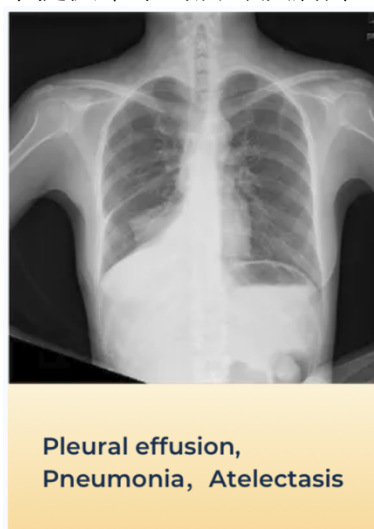
- Data Source

資料來源從他們競賽網站 <https://medfm2023.grand-challenge.org/datasets/>即可下載 6/2 更新公布的資料集，將 train 與 val 的影像資料混合再一起，依任務類別放到 data/medfmc/MedFMC/chest, data/medfmc/MedFMC/colon, data/medfmc/MedFMC/endo 底下。

這個競賽主旨是著重在 few-shot learning 與 prompt engineering，因此不允許參賽者使用全部的 data 訓練，也不允許使用競賽資料及以外的資料。我在這次的期末報告中，會使用他們示範的 baseline 裡面切好的 train, val, test datasets 當作我的訓練以及衡量結果的資料集。挑戰內容也有提到，會有這些限制是希望解決醫學影像中標註不足的問題，希望參與者利用少量私有數據進行初始訓練和驗證，並嘗試提高分類準確性，包括放射學、病理學和結腸鏡影像。

- Dataset

第一項任務：胸部疾病篩檢 (ChestDR)，目的在利用基礎模型，通過少量樣本來提供準確的胸部疾病篩查方法。使用胸部 X 光影像作為日常臨床篩查工具。

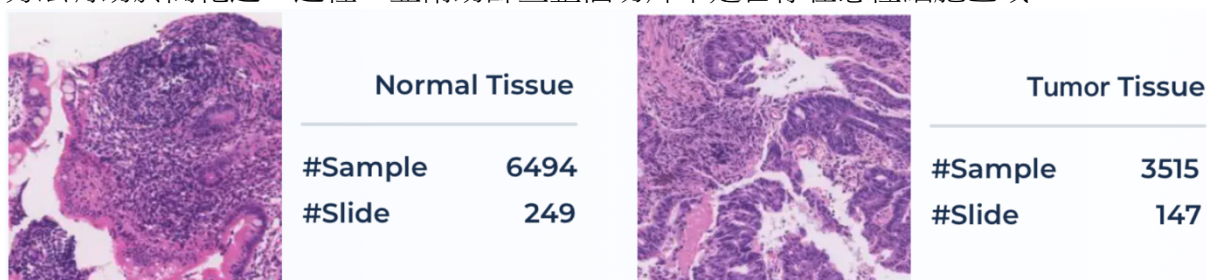


Abnormality	#Sample	Abnormality	#Sample
Cardiomegaly	1300	TB	305
Pleural effusion	1266	Pneumothorax	284
Pneumonia	898	Atelectasis	199
Hilar enlargement	758	Emphysema	159
Nodule	665	Calcification	104
Aortic calcification	645	Pulmonary edema	86
Tortuous aorta	612	Increased lung markings	53
Fibrosis	532	Consolidation	34
Thickened pleura	497	Elevated diaphragm	23
Fracture	429		

(圖一，圖片來源：<https://medfm2023.grand-challenge.org/medfm2023/>)

第二項任務：病理性腫瘤組織分類 (Colon)，目的在於利用基礎模型，通過少量樣本提供準確的病理性腫瘤組織分類方法。在病理學家日常的工作中，他們

需要檢查許多組織切片，這對他們來說是一項繁瑣的工作。分類病理組織塊的方法有助於簡化這一過程，並幫助篩查整個切片中是否存在惡性細胞區域。



(圖二，圖片來源：<https://medfm2023.grand-challenge.org/medfm2023/>)

第三項任務：**病變偵測於結腸鏡影像 (Endo)**，目的是利用基礎模型，在結腸鏡檢查過程中自動檢測病變，以防止漏診並減輕胃腸科醫師的工作量。



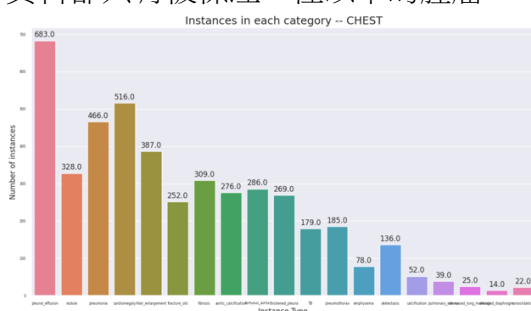
(圖三，圖片來源：<https://medfm2023.grand-challenge.org/medfm2023/>)

- Exploratory Data Analysis (EDA)

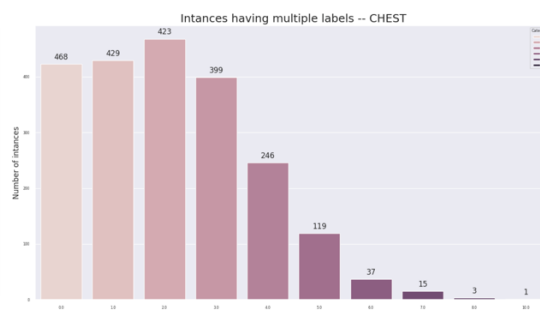
使用 `eda.ipynb` 進行資料視覺化，其中參考了此 blog：

<https://towardsdatascience.com/journey-to-the-center-of-multi-label-classification-384c40229bff>。

從（圖四）可以看到，在第一項任務 CHEST 的訓練資料集當中，以 **pleural effusion** 的佔比最高，有 683 筆，而 **elevated diaphragm** 佔比最低，只有 14 筆。從（圖五）可以看到，有 1 筆資料被標注最多的 10 種腫瘤偵測，其他大多數的資料都只有被標注 3 種以下的腫瘤。



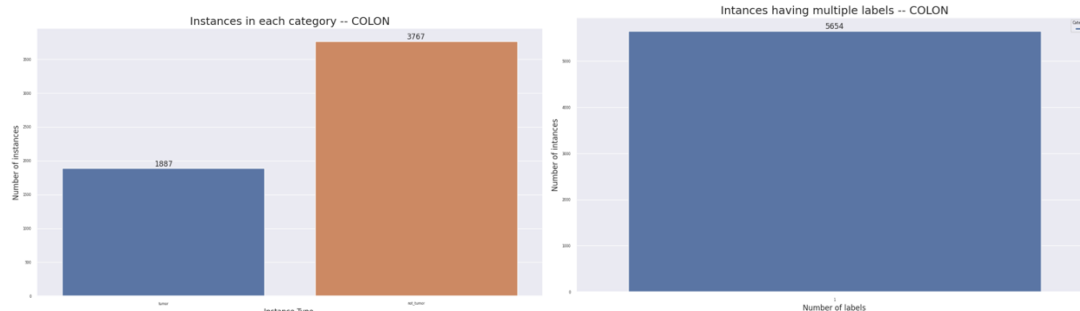
(圖四)



(圖五)

從（圖六）可以看到，在第二項任務 COLON 的訓練資料集當中，被標注為無腫瘤組織比有的大約多一倍。

從（圖七）可以確認此項任務為 **binary classification**。

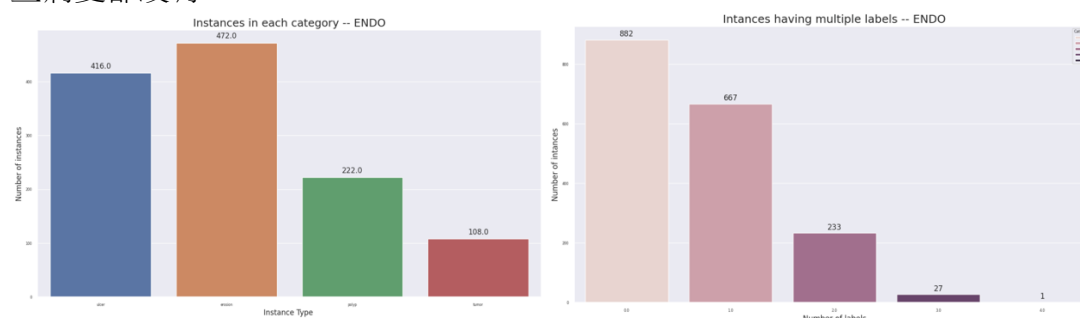


（圖六）

（圖七）

從（圖八）可以看到，在第三項任務 ENDO 的訓練資料集當中，以 **erosion** 的佔比最高，有 472 筆，而 **tumor** 佔比最低，不過整體分佈還算不算偏差太嚴重。

從（圖九）可以看到，有 1 筆資料被標注 4 種病變都有，而有 882 筆被標注這些病變都沒有。



（圖八）

（圖九）

II. Methodology

- Model and Training Method

1. *Transfer Learning on 20% of Training Data by Densenet*

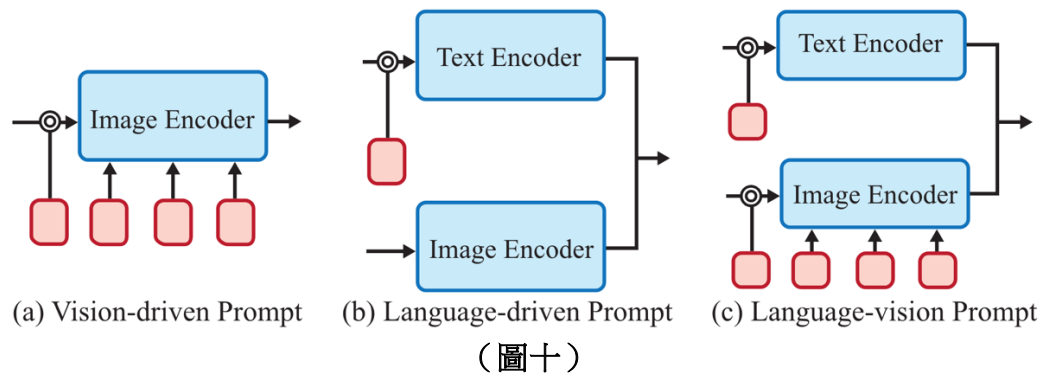
這次報告使用的競賽中，附有他們自己製作的 **baseline model** (<https://github.com/openmedlab/MedFM>)，使用第三方套件 **mmpretrain** 實作。

不過他們是使用較舊的 **mmpretrain** 版本，而在這次報告我是使用最新版本的 **mmpretrain**，其文件配置與 **ground truth annotation file** 的格式都會與原本的範例 **baseline** 不一樣。

2. *Few-shot Learning by CoCoOp*

另外，我尋找了其他 **visual prompt tuning** 的方法，找到了 **CoCoOp** (Conditional Prompt Learning for Vision-Language Models, in CVPR, 2022, cited 883 times)，有別於範例程式碼提供的 **baseline** 使用的是 **VPT** (Visual Prompt

Tuning, in ECCV, 2022, cited 973 times), 根據 Yu et al. 發表的 Visual Tuning (in ACM Computing Surveys, 2024, cited 16 times)的分類, VPT 為 Vision-driven Prompt, 而 CoCoOp 為 Language-driven Prompt。(圖十)為 Visual Tuning 解釋這兩種 prompt method 的差異, 紅色為 tunable block, 藍色為 frozen block。在這次期末報告我想嘗試使用不同種類的 Visual Tuning, 並且觀察其在此任務上的差異, 因此使用 CoCoOp 當作我這次實作 few-shot learning 的方法。



- Data Preprocessing

1. *Transfer Learning on 20% of Training Data by Densenet*

在使用 mmpretrain 方面, 由於第一項任務 **Chest** 與第三項任務 **Endo** 都是 multi-label classification 的任務, 我因此參考了 mmpretrain 的文件: https://mmpretrain.readthedocs.io/en/stable/modules/mmpretrain/datasets/multi_label.html, 製作出符合 mmpretrain 需求的 annotation files, 此文件格式為 pickle 檔, 存取的資料型態是 dictionary, 以下為範例 (圖十一)。

```
{
  "metainfo":
  {
    "classes": ['A', 'B', 'C'...]
  },
  "data_list":
  [
    {
      "img_path": "test_img1.jpg",
      'gt_label': [0, 1],
    },
    {
      "img_path": "test_img2.jpg",
      'gt_label': [2],
    },
    ...
  ]
}
```

(圖十一, 圖片來源:

https://mmpretrain.readthedocs.io/en/stable/modules/mmpretrain/datasets/multi_label.html)

而第二項任務 Colon 則是 binary classification 的任務, 因此只需遵照一般的 annotation, 以下為範例。

```

folder_1/xxx.png 0
folder_1/xyx.png 1
123.png 4
nsdf3.png 3
...

```

(圖十二，圖片來源：https://mmpretrain.readthedocs.io/zh-cn/latest/user_guides/dataset_prepare.html)

另外，configuration 也依照 mmpretrain 配置文件的教學指示，(圖十三) 為官方範例。

```

MMPretrain/
├── configs/
│   ├── _base_/                                # primitive configuration folder
│   │   ├── datasets/                          # primitive datasets
│   │   ├── models/                            # primitive models
│   │   ├── schedules/                        # primitive schedules
│   │   └── default_runtime.py                 # primitive runtime setting
│   ├── beit/                                # BEiT Algorithms Folder
│   ├── mae/                                 # MAE Algorithms Folder
│   ├── mocov2/                              # MoCoV2 Algorithms Folder
│   ├── resnet/                              # ResNet Algorithms Folder
│   ├── swin_transformer/                    # Swin Algorithms Folder
│   ├── vision_transformer/                  # ViT Algorithms Folder
│   └── ...
└── ...

```

(圖十三，圖片來源：https://mmpretrain.readthedocs.io/zh-cn/latest/user_guides/config.html)

2. Few-shot learning by CoCoOp

在 CoCoOp 方面，我先使用其他範例 datasets 說明文件：

<https://github.com/KaiyangZhou/CoOp/blob/main/DATASETS.md>，了解我的該如何配置檔案位置，以及 annotation files 的格式。

在 Caltech101 這個範例資料中，annotation file: split_zhou_Caltech101.json 格式為以下範例 (圖十四) 與 (圖十五)，存取的內容為一個 dictionary。

```

▼ root:
  ► train: [] 4128 items
  ► val: [] 1649 items
  ► test: [] 2465 items

```

(圖十四)

```

▼ root:
  ▼ train: [] 4128 items
    ▼ 0: [] 3 items
      0: "Faces/image_0274.jpg"
      1: 0
      2: "face"

```

(圖十五)

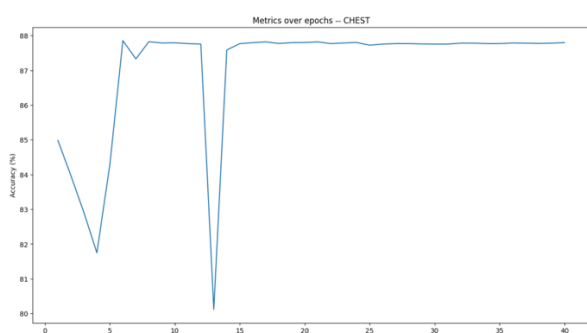
III. Experiment and Result

在之前的作業當中，我是自己以 80%當作 training data，不過在這裡為了配合範例程式碼，我則是使用它已經切好的 train, val, test datasets 當作訓練以及測試資料。Densenet 是以 20% data 去做 transfer learning，而 CoCoOp 則是以 1-shot, 5-shot, 10-shot 去做 few-shot learning。根據競賽的說明，這裡的 few-shot number 是以病患的數量決定，而非影像的數量。另外，我使用兩種 metrics: mAP & Accuracy 去衡量我最後的預測結果。

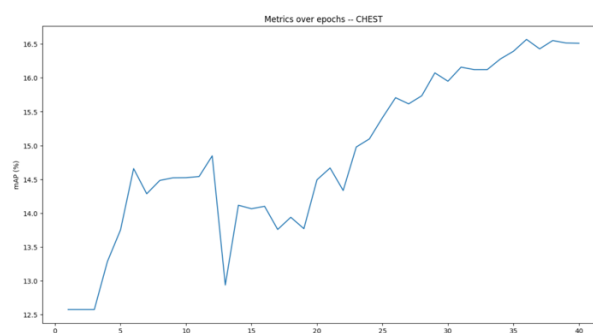
a. 第一項任務：胸部疾病篩檢 (ChestDR)

Number of instances	Densenet	CoCoOp 1-shot	CoCoOp 5-shot	CoCoOp 10-shot
Train	783	19	95	190
Val	196	960	884	789
Test	1161	1161	1161	1161

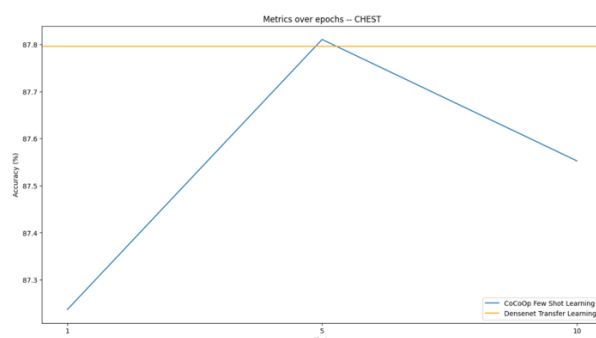
從（圖十六）與（圖十七）可以看到，儘管訓練越多 epochs 可能無法有效提升 Accuracy，不過 mAP 卻持續在增加，而（圖十八）與（圖十九）顯示 CoCoOp 跟 Densenet 的對比，發現 CoCoOp 只能在 Accuracy 方面與 Densenet 有相當的準確率，不過從 mAP 來看卻不如 Densenet。



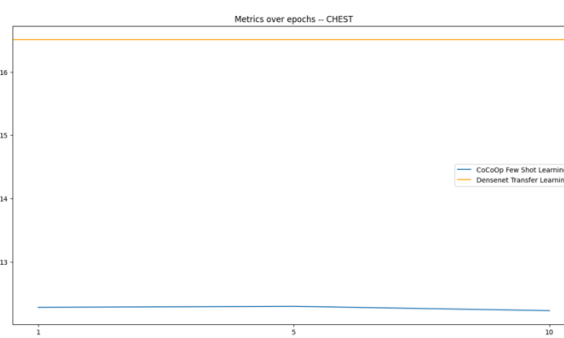
（圖十六）



（圖十七）



（圖十八）

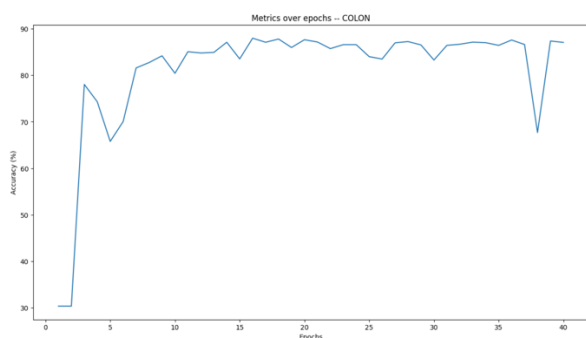


（圖十九）

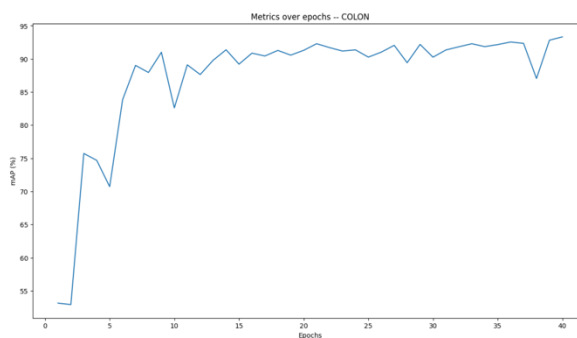
b. 第二項任務：病理性腫瘤組織分類 (Colon)

Number of instances	Densenet	CoCoOp 1-shot	CoCoOp 5-shot	CoCoOp 10-shot
Train	1677	30	234	512
Val	681	2328	2124	1846
Test	3296	3296	3296	3296

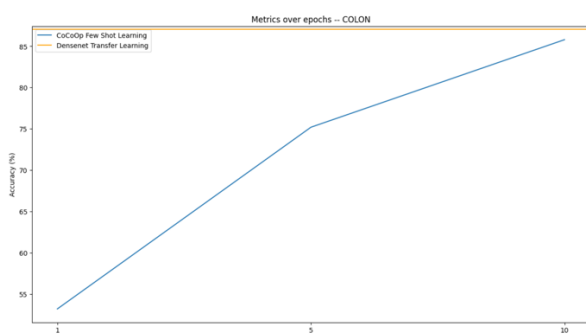
從（圖二十）與（圖二十一）可以看到，Densenet 對於 Colon 資料集的訓練在早期效果就已經飽和了，而（圖二十二）與（圖二十三）顯示 CoCoOp 跟 Densenet 的對比，發現 CoCoOp 只能在 Accuracy 方面與 Densenet 有相當的準確率，不過從 mAP 來看卻遠低於 Densenet。



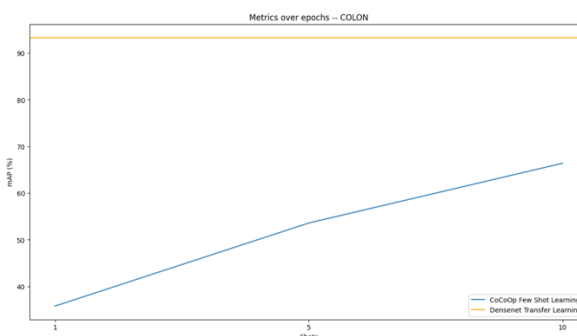
（圖二十）



（圖二十一）



（圖二十二）

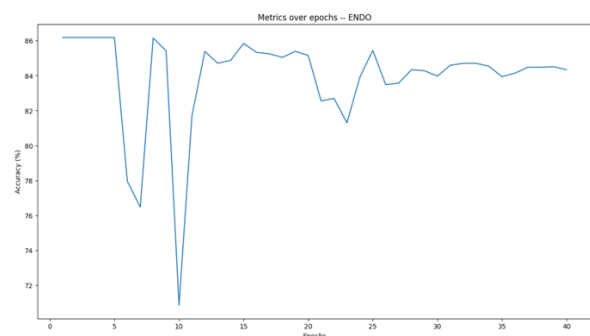


（圖二十三）

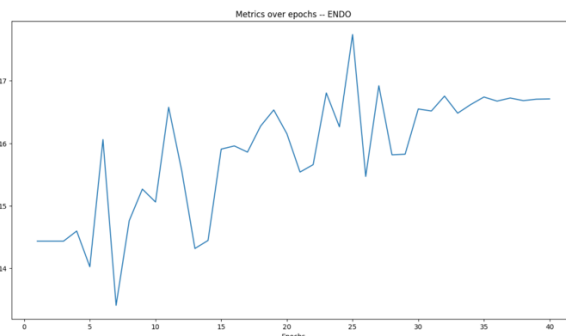
c. 第三項任務：病變偵測於結腸鏡影像 (Endo)

Number of instances	Densenet	CoCoOp 1-shot	CoCoOp 5-shot	CoCoOp 10-shot
Train	743	46	174	422
Val	186	883	755	507
Test	881	881	881	881

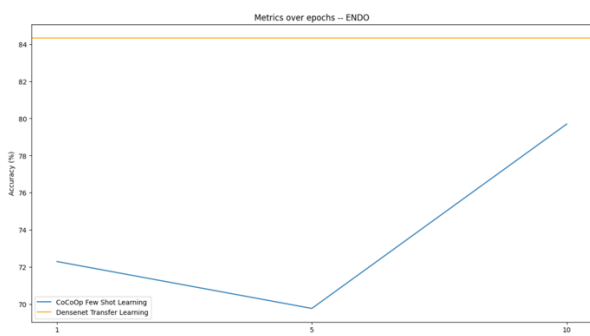
從（圖二十四）與（圖二十五）可以看到，Densenet 對於 Endo 資料集的訓練在早期的 Accuracy 就已經飽和了，不過 mAP 還是有微幅上升，而（圖二十六）與（圖二十七）顯示 CoCoOp 跟 Densenet 的對比，這裡的結果有別於前面 Chest 與 Colon，反而是 CoCoOp 在 mAP 方面有機會高於 Densenet，不過從 Accuracy 來看卻不及 Densenet。



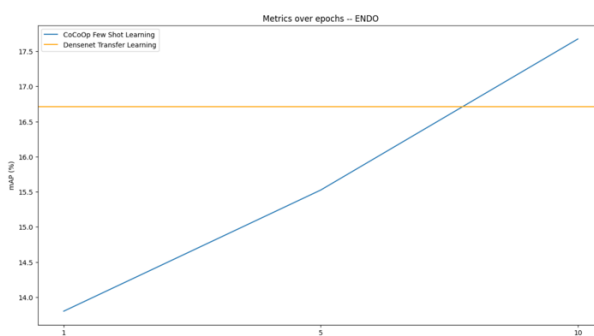
(圖二十四)



(圖二十五)



(圖二十六)

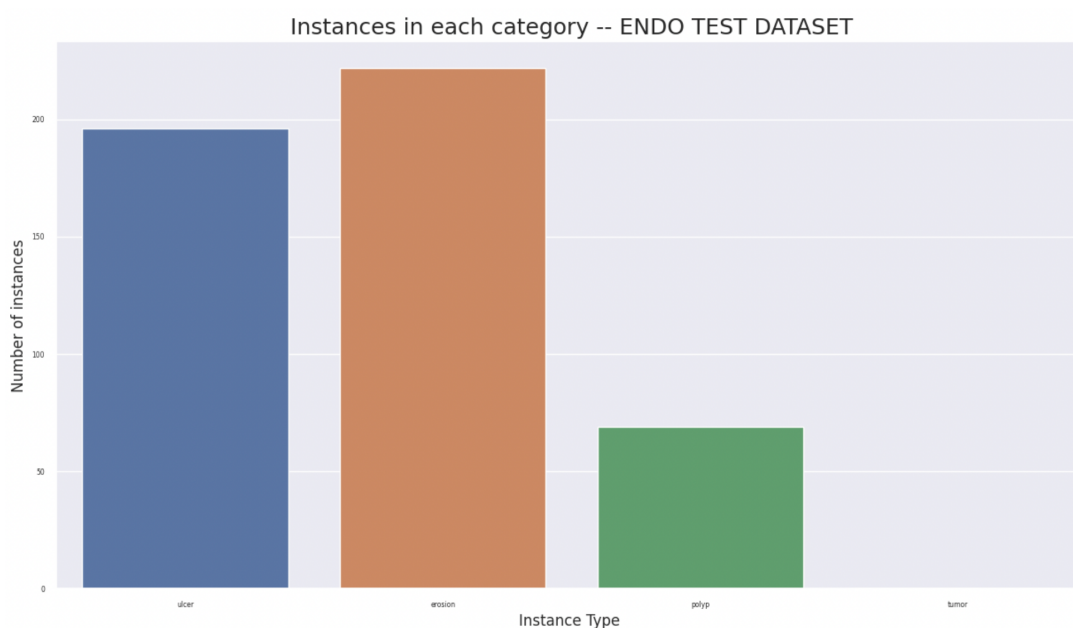


(圖二十七)

IV. Conclusion

- Discussion: Distribution of labels in Endo test data

我在檢視各類別的 mAP 時發現，沒有任何資料有被標注 tumor，代表不管如何 tumor 的 mAP 都會是 0，進而導致 Endo test data 的 mAP 都很低，也可以從前面的 EDA 發現 tumor 的標注數量本來就是最少的，或許範例是希望將少量帶有 tumor 的圖像放在 training 以及 validation data 裡面，去增加模型的效能。



- Future Work

- 使用 DualCoOp 以及其他 multilabel CoOp

這次挑戰的三個資料集當中就有兩個是屬於 multilabel classification，不過其實 CoCoOp 只能執行 binary classification，因此我依照 class 數量把他們拆分成多個 binary classification，這會大大增加運算量以及 overfitting 的可能行，也會降低它的 generalization 能力，可能與原挑戰的宗旨背離，因此應該嘗試其他針對 multilabel classification 的改良，像是 DualCoOp 以及 TAI++。

- 測試 prompt 的有效度

在實作 CoCoOp 時，預設的 prompt 為「a photo of a {}」，{} 會被替換成圖像所屬的 class，不過他們也有針對一些資料集去設計 prompt，像是 OxfordPets 的 prompt 為「a photo of a {}, a type of pet.」，因此我也針對這三個資料集去設計他們的 prompt。

Dataset	Prompt
Chest	A chest X-ray with thoracic disease {}.
Colon	A tissue slide of pathological tissue patches with {}.
Endo	A colonoscopy image with abnormalities {}.

不過不能保證這樣的 prompt 對於預測任務是有幫助的，因此應該先使用預設 prompt 測試一下結果，再使用新的 prompt 去驗證這樣設計的 prompt 是否有比較好。

- 使用 Gradient-weighted Class Activation Mapping 等方法視覺化那些在圖像中造成重要影響的部分。
- 使用 experiment tracking tools 去追蹤我的實驗過程與結果，例如 Neptune AI 與 Comet ML 等等。