

深度學習於醫學影像 DLMI -- Homework

R12725024 資管碩一 蔡鈺驊

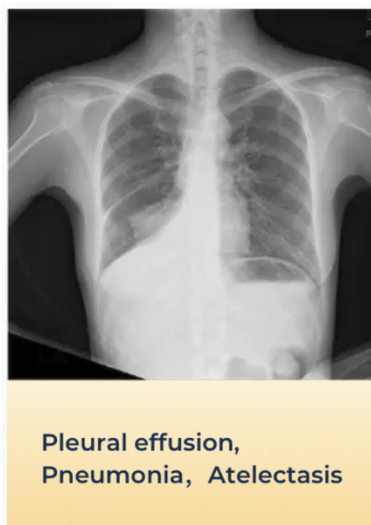
Data

- Dataset

這次作業我使用 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>。

這個競賽主旨是著重在 few-shot learning 與 prompt engineering，因此原本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，不過我在這次的作業中，會先使用全部的資料做訓練當作練習，而之後在做 final project 也希望使用額外的資料達到更完善的 foundation model。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，希望參與者將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。

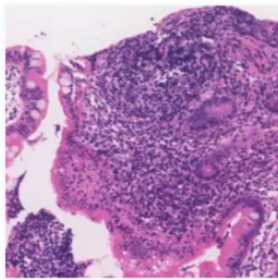
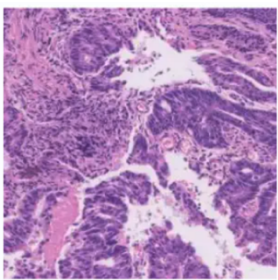
第一項任務：**胸部疾病篩檢 (ChestDR)**，目的在利用基礎模型，通過少量樣本來提供準確的胸部疾病篩查方法。使用胸部 X 光影像作為日常臨床篩查工具。



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		

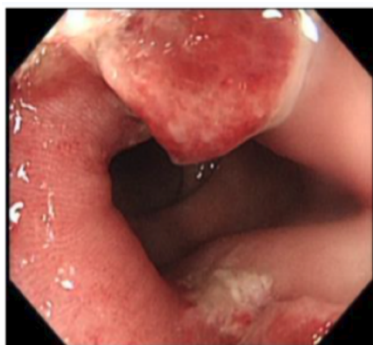
(圖一，圖片來源：<https://medfm2023.grand-challenge.org/medfm2023/>)

第二項任務：**病理性腫瘤組織分類 (Colon)**，目的在在利用基礎模型，通過少量樣本提供準確的病理性腫瘤組織分類方法。在日常病理學家的工作中，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。

	Normal Tissue <hr/> #Sample 6494 #Slide 249		Tumor Tissue <hr/> #Sample 3515 #Slide 147
---	--	--	---

(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)**，目的在利用基礎模型，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。

	Leison	#Sample	#Study
Ulcer, Tumor	Ulcer	790	69
	Erosion	951	70
	Polyp	390	62
	Tumor	136	11

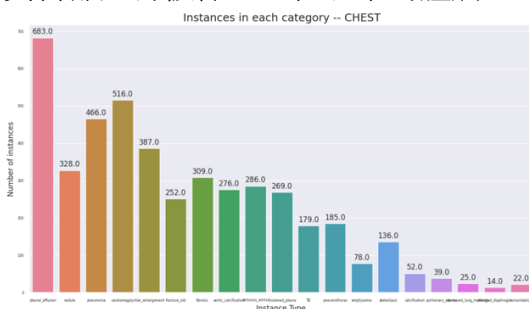
(圖三，圖片來源：<https://medfm2023.grand-challenge.org/medfm2023/>)

- **Exploratory Data Analysis (EDA)**

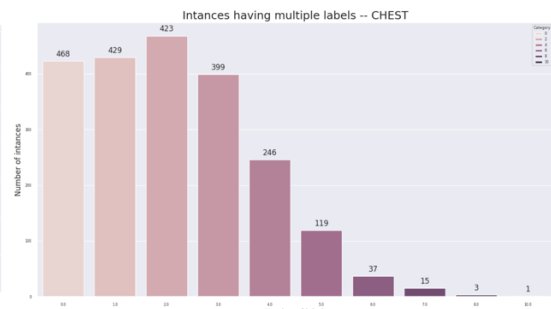
使用 data/MedFMC/eda.ipynb 進行資料視覺化，其中參考了此 blog：
<https://towardsdatascience.com/journey-to-the-center-of-multi-label-classification-384c40229bff>。

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。



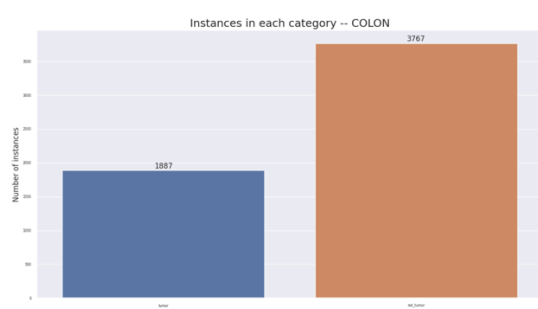
(圖四)



(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。



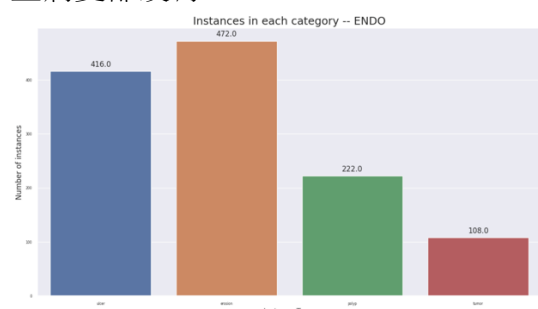
（圖六）



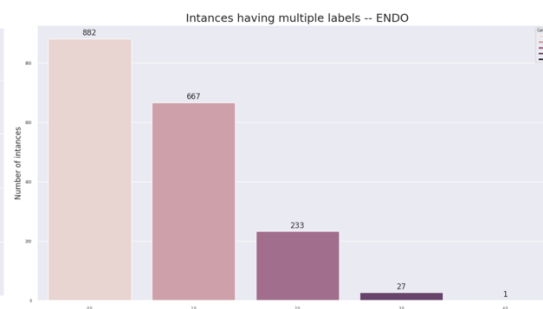
（圖七）

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 683 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）

Methodology

這次作業使用的競賽中，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 few-shot learning method 與 Visual Prompt Tuning，backbone 則是有嘗試 DenseNet, EfficientNet, Vision Transformer 以及 Swin Transformer。

不過由於他們是使用較舊的 mmpretrain 版本，若我想嘗試較新的 pretrained model，像是 BEiT v2 等其他 state-of-the-art 模型的話，就需要更新的版本，其文件配置與 ground truth annotation file 的格式都會不一樣，因此在此作業我先熟悉練習 mmpretrain 的使用方式，在 final project 中再實作 few-shot learning 與 visual prompt tuning。

- 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/chest, data/MedFMC/colon, data/MedFMC/endo 底下。原有標注資料則放在 data/MedFMC/

- Data Preprocessing

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，使用 data/MedFMC/generate_custom_label.ipynb 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，大致上長得像以下範例（圖十）。

```
{
  "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，圖片來源：

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

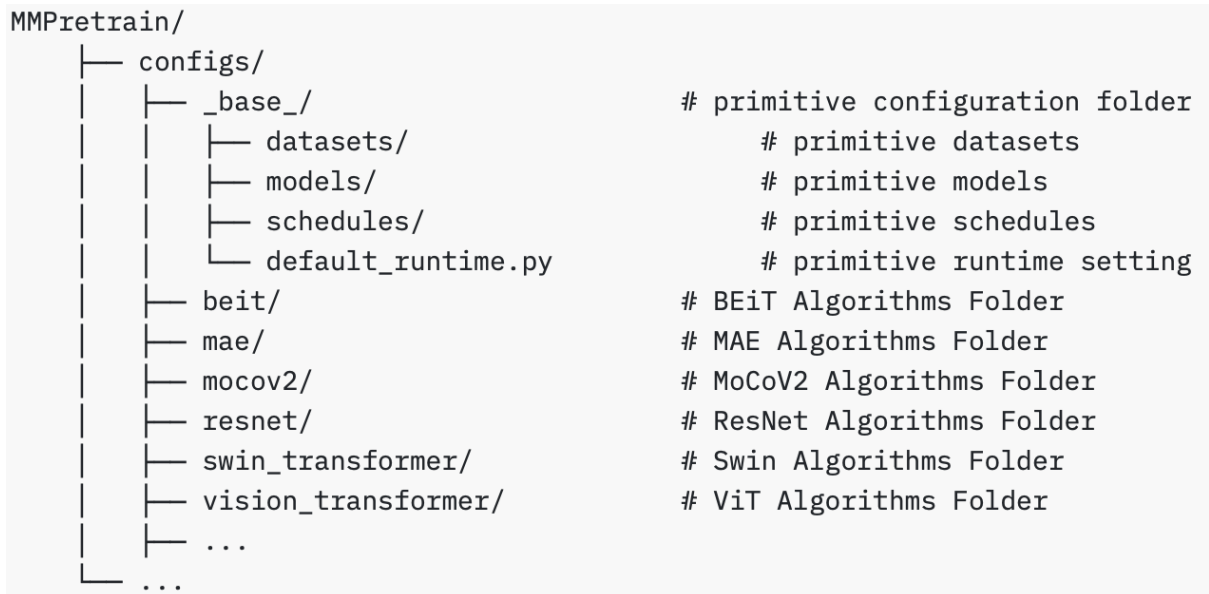
而第二項任務 Colon 則是 binary classification 的任務，因此只需遵照一般的 annotation，大致上長得像以下範例。

```
folder_1/xxx.png 0
folder_1/xyy.png 1
123.png 4
nsdf3.png 3
...
```

（圖十一，圖片來源：https://mmpretrain.readthedocs.io/zh-cn/latest/user_guides/dataset_prepare.html）

- Configuration

依照 mmpretrain 配置文件的教學指示，大致上會長得像以下（圖十二）。



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

因此我直接 clone 整個 mmpretrain 的 Github repo (<https://github.com/open-mmlab/mmpretrain.git>)，再依照我的需求去更改細節。

a. 新增 configs/_base_/models/densenet/densenet121-multilabel.py

Duplicate configs/_base_/models/densenet/densenet121.py，並將最後的 classification head type 從 'LinearClsHead' 改成 'MultiLabelLinearClsHead'，並且命名為 densenet121-multilabel.py。

b. 新增 configs/_base_/datasets/imagenet_bs64-chest.py, imagenet_bs64-colon.py, imagenet_bs64-endo.py

Duplicate configs/_base_/datasets/imagenet_bs64.py，調整他們各自的 dataset_type, data_root, ann_file, data_prefix, and val_evaluator。

c. 新增 configs/densenet/densenet121_4xb256_in1k-chest.py, densenet121_4xb256_in1k-colon.py, densenet121_4xb256_in1k-endo.py

其中 chest 與 endo 所使用的 model 是 densenet121-multilabel.py，而 colon 則是使用原先有的 densenet121.py，使用從 <https://mmpretrain.readthedocs.io/zh-cn/latest/papers/densenet.html> 下載下來放置於 pretrain/的 pretrained model，更改他們各自 classification head 的 num_classes，chest, colon, endo 分別是 19, 2, 4。

- Experiments and Results

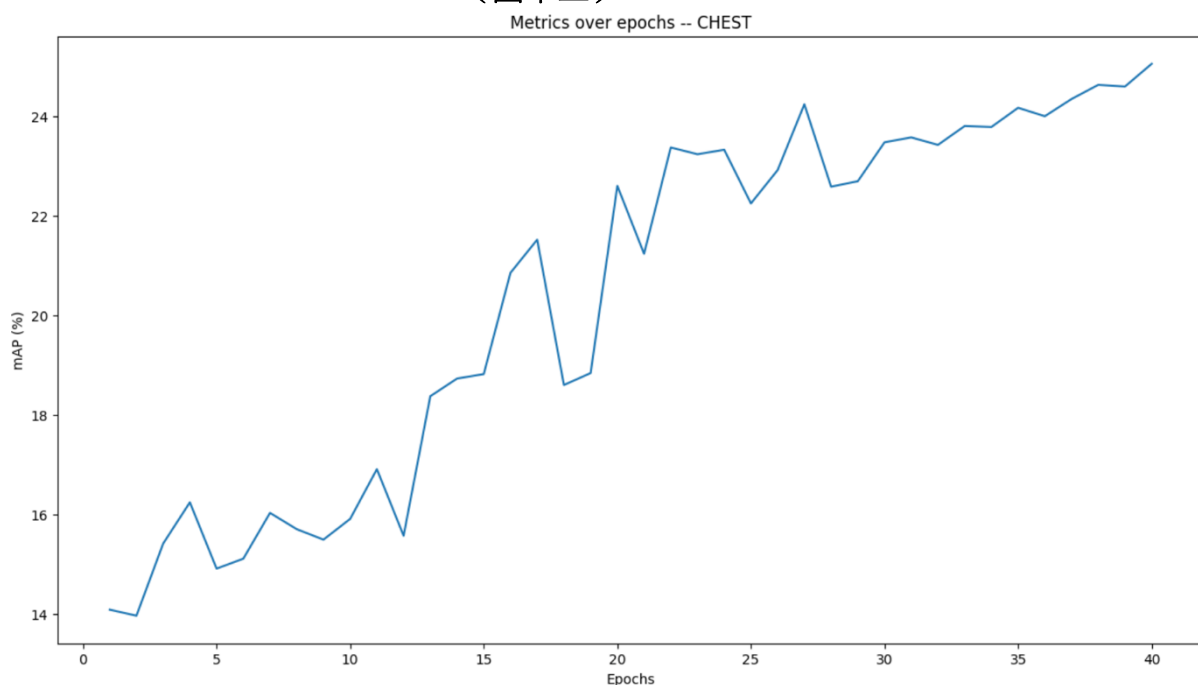
執行 bash train.sh 訓練 densenet 去預測三項任務，並且使用 bash test.sh 去檢測每個 epoch 的 model 表現如何，再使用 evaluate.ipynb 去檢視實驗結果。

a. 第一項任務：**胸部疾病篩檢 (ChestDR)**

可以看到，模型在這 40 個 epochs 當中，表現持續上升，因此預計再訓練多個 epochs 能達到更好的表現。

	0	1	2	3	4	5	6	7	8	9
Epochs	1.0000	2.0000	3.0000	4.0000	5.0000	6.0000	7.0000	8.0000	9.0000	10.0000
mAP (%)	14.0923	13.9705	15.4188	16.2481	14.9165	15.1161	16.0359	15.7086	15.4987	15.9153
	10	11	12	13	14	15	16	17	18	19
Epochs	11.0000	12.0000	13.0000	14.0000	15.0000	16.0000	17.0000	18.0000	19.0000	20.0000
mAP (%)	16.9145	15.5769	18.3793	18.7341	18.8241	20.858	21.5239	18.6044	18.8445	22.604
	20	21	22	23	24	25	26	27	28	29
Epochs	21.0000	22.0000	23.0000	24.0000	25.0000	26.0000	27.0000	28.0000	29.0000	30.0000
mAP (%)	21.2405	23.3758	23.2395	23.3297	22.2515	22.9215	24.2437	22.5878	22.6971	23.4793
	30	31	32	33	34	35	36	37	38	39
Epochs	31.0000	32.0000	33.0000	34.0000	35.0000	36.0000	37.0000	38.0000	39.0000	40.0000
mAP (%)	23.5776	23.4275	23.8062	23.7865	24.1725	24.0034	24.3488	24.6332	24.5994	25.0562

(圖十三)



(圖十四)

b. 第二項任務：病理性腫瘤組織分類 (Colon)

可以看到，模型在跑完第 6 個 epoch 時，表現就已經趨近飽和了。

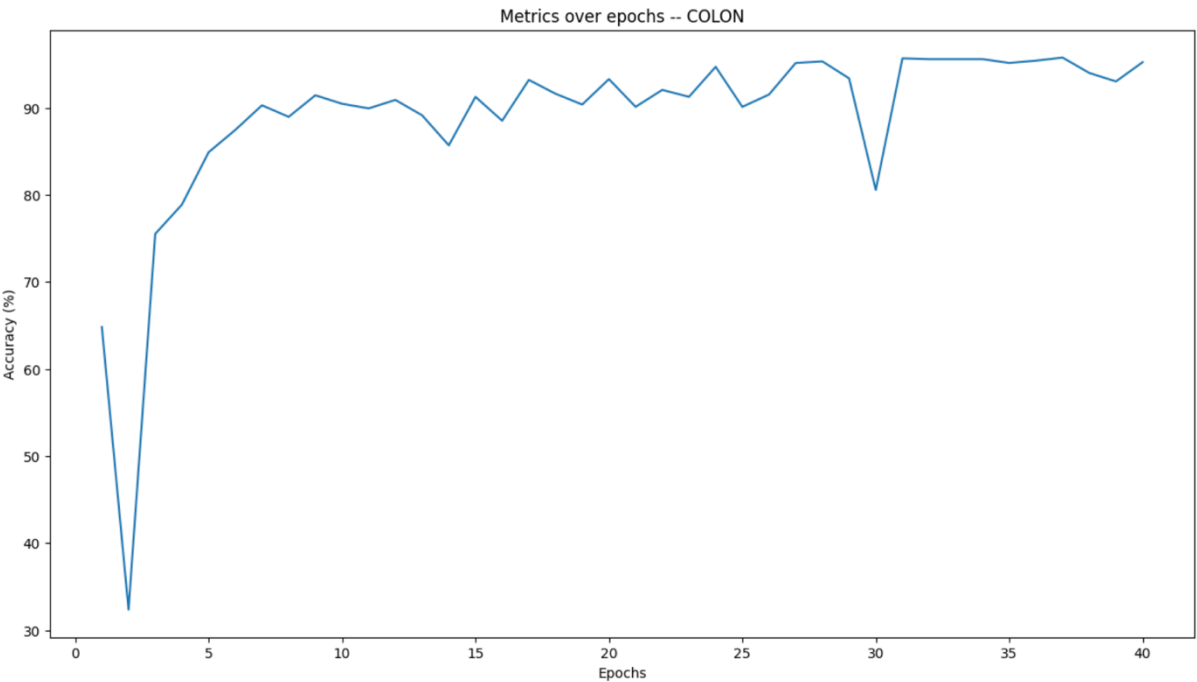
	0	1	2	3	4	5	6	7	8	9
Epochs	1.0000	2.0000	3.0000	4.0000	5.0000	6.0000	7.0000	8.0000	9.0000	10.0000
Accuracy (%)	64.8099	32.3607	75.5084	78.8683	84.8806	87.4447	90.2741	88.9478	91.4235	90.4509

	10	11	12	13	14	15	16	17	18	19
Epochs	11.0000	12.0000	13.0000	14.0000	15.0000	16.0000	17.0000	18.0000	19.0000	20.0000
Accuracy (%)	89.9204	90.893	89.1247	85.6764	91.2467	88.5057	93.1919	91.6003	90.3625	93.2803

	20	21	22	23	24	25	26	27	28	29
Epochs	21.0000	22.0000	23.0000	24.0000	25.0000	26.0000	27.0000	28.0000	29.0000	30.0000
Accuracy (%)	90.0973	92.0424	91.2467	94.695	90.0973	91.5119	95.137	95.3139	93.3687	80.5482

	30	31	32	33	34	35	36	37	38	39
Epochs	31.0000	32.0000	33.0000	34.0000	35.0000	36.0000	37.0000	38.0000	39.0000	40.0000
Accuracy (%)	95.6675	95.5791	95.5791	95.5791	95.137	95.4023	95.756	93.9876	93.015	95.2255

(圖十五)

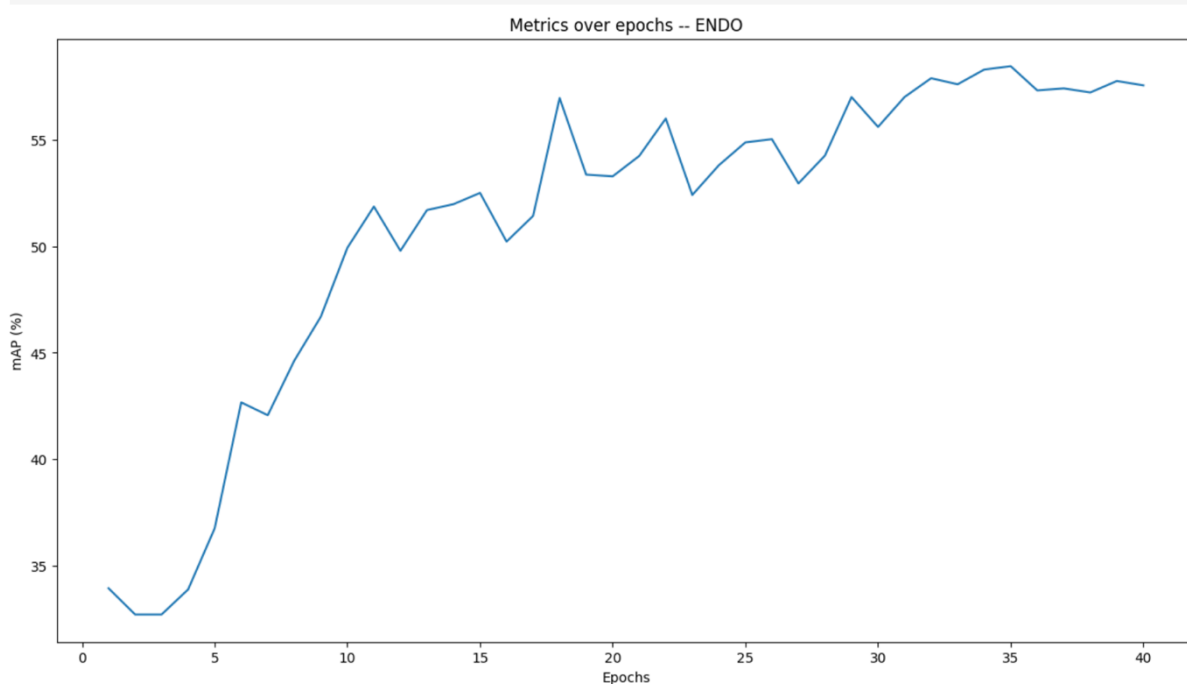


(圖十六)

c. 第三項任務：病變偵測於結腸鏡影像 (Endo)

可以看到，模型在約第 31 個 epoch 時，表現就已經趨近飽和了。

	0	1	2	3	4	5	6	7	8	9
Epochs	1.0000	2.0000	3.0000	4.0000	5.0000	6.0000	7.0000	8.0000	9.0000	10.0000
mAP (%)	33.9335	32.7033	32.7033	33.8798	36.7521	42.6696	42.0675	44.6326	46.6983	49.9349
	10	11	12	13	14	15	16	17	18	19
Epochs	11.0000	12.0000	13.0000	14.0000	15.0000	16.0000	17.0000	18.0000	19.0000	20.0000
mAP (%)	51.8748	49.7963	51.7088	51.9866	52.5154	50.2276	51.4358	56.9755	53.3753	53.2951
	20	21	22	23	24	25	26	27	28	29
Epochs	21.0000	22.0000	23.0000	24.0000	25.0000	26.0000	27.0000	28.0000	29.0000	30.0000
mAP (%)	54.253	56.0103	52.4192	53.8154	54.8904	55.042	52.9618	54.2713	57.0209	55.6179
	30	31	32	33	34	35	36	37	38	39
Epochs	31.0000	32.0000	33.0000	34.0000	35.0000	36.0000	37.0000	38.0000	39.0000	40.0000
mAP (%)	57.0235	57.906	57.6213	58.3124	58.4719	57.3342	57.4276	57.2375	57.7757	57.5705



Future Work – Final Project

- 在分割訓練與驗證資料集時，依照類別分佈切割得更細緻。
- 研究其他 **few-shot learning** 與 **visual prompting** 的方法，並且實作出來。
- 使用 **experiment tracking tools** 去追蹤我的實驗過程與結果，像是 **Neptune AI** 與 **Comet ML** 等等。
 - <https://neptune.ai/blog/ml-experiment-tracking>

- <https://neptune.ai/blog/best-ml-experiment-tracking-tools>
- 使用 Gradient-weighted Class Activation Mapping 等方法視覺化那些圖像在任務中造成重要影響的部分。
- 將所有實作與 reproduce 的細節推上 Github。