可研究方向：

* 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。
  + Visual Prompt Tuning: https://paperswithcode.com/paper/visual-prompt-tuning
  + Visual Instruction Tuning: https://paperswithcode.com/paper/visual-instruction-tuning-1
  + Conditional Prompt Learning for Vision-Language Models: https://paperswithcode.com/paper/conditional-prompt-learning-for-vision
  + Image Classification只有出現在Visual Prompt Tuning裡而已
* 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。

To-do list

* 研究為什麼那些事Visual Prompt Tuning
* 研究mmpretrain的模型到底有哪些，怎麼做Visual Prompt Tuning
* 嘗試用其他方式做
* 嘗試融合上課內容

https://paperswithcode.com/paper/visual-prompt-tuning

用mmpretrain一樣試其他visual prompt tuning

用別人的github做其他visual tuning

Main與master有很大的差異

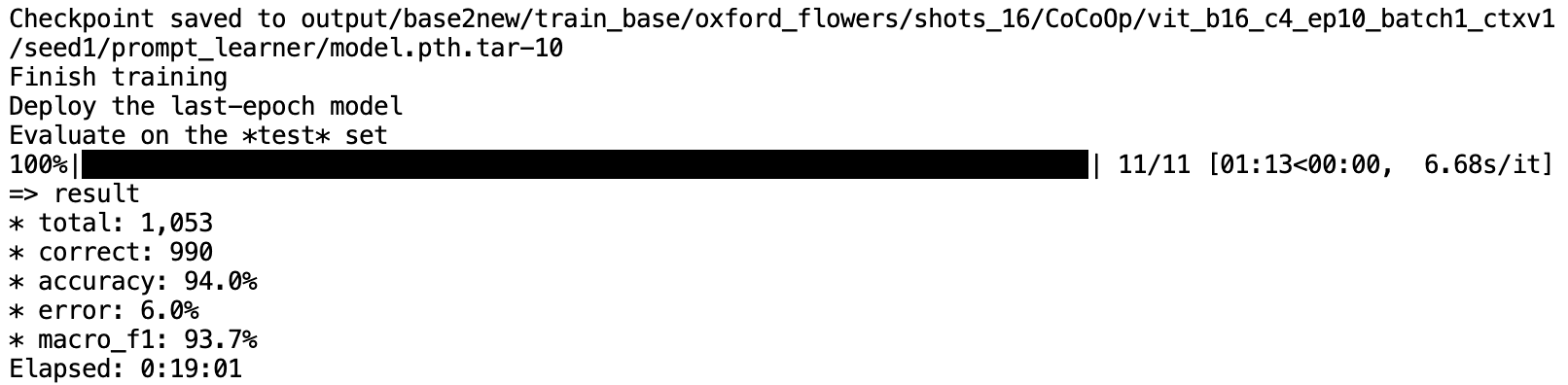
Comment cd ../..

export data=/home/arthur/hw/dlmi/final/CoOp/data

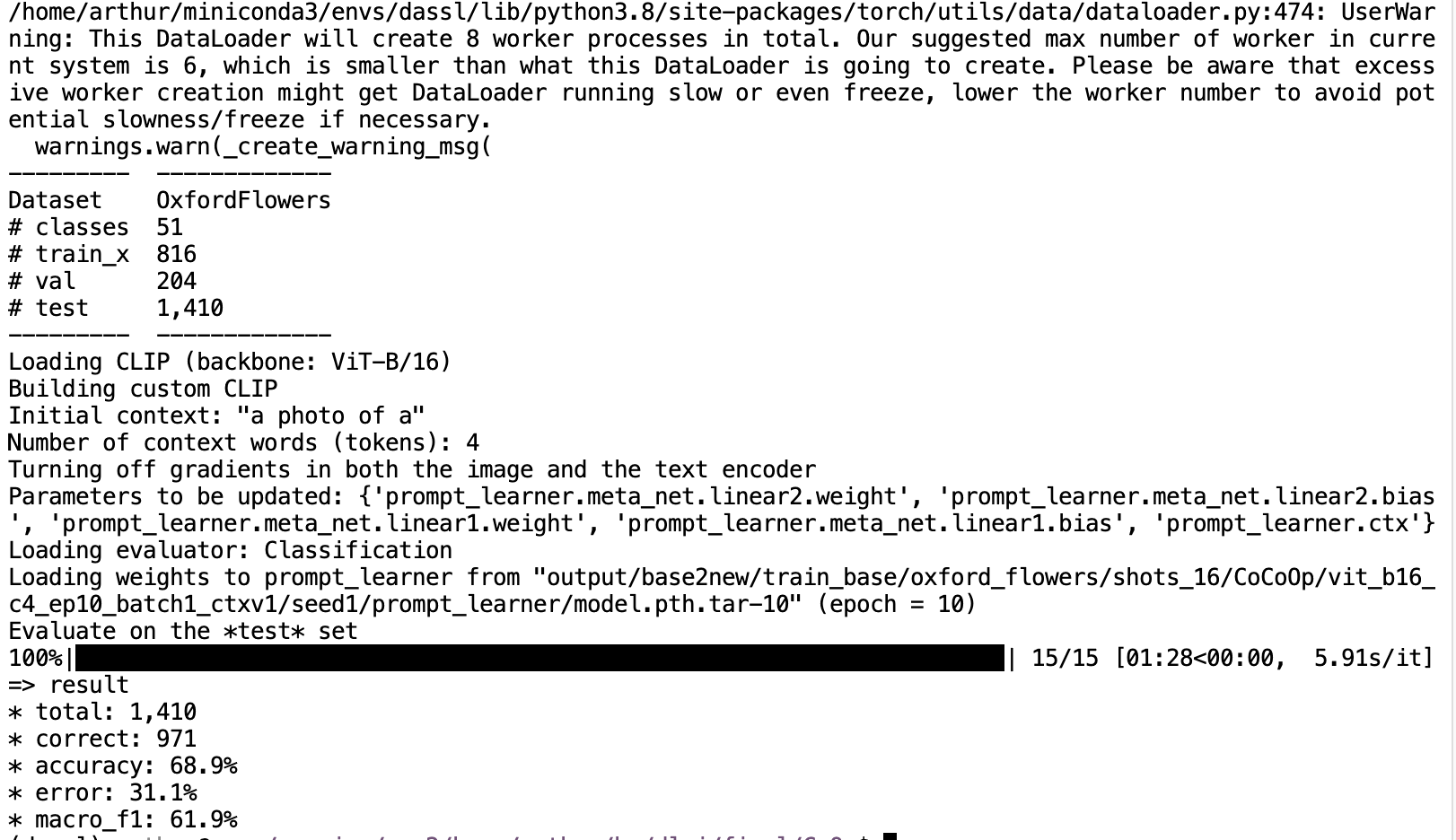
DATA=/home/arthur/hw/dlmi/final/CoOp/data

ln -s /home/arthur/large\_folder/dassl/CoOp/data /home/arthur/hw/dlmi/final/CoOp/data

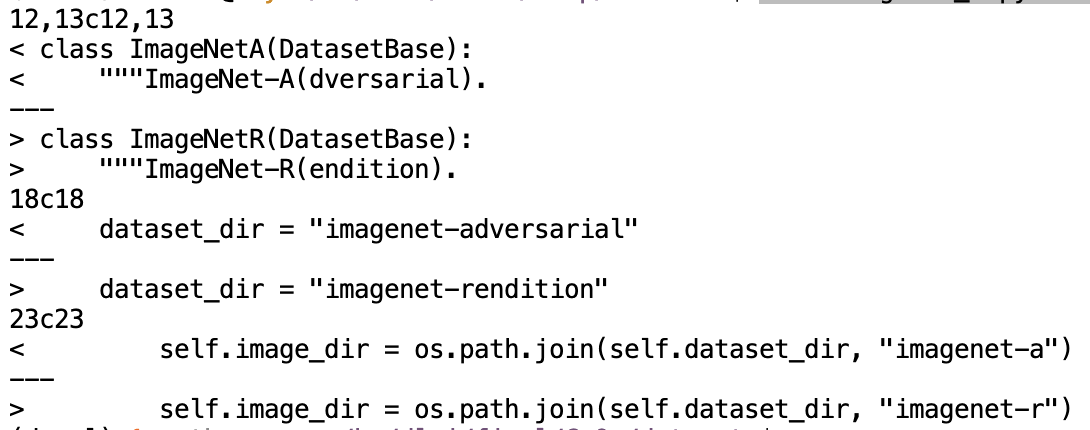
bash scripts/cocoop/base2new\_train.sh oxford\_flowers 1



bash scripts/cocoop/base2new\_test.sh oxford\_flowers 1



diff imagenet\_a.py imagenet\_r.py



Create files:

configs/datasets/medfm\_chest.yaml:



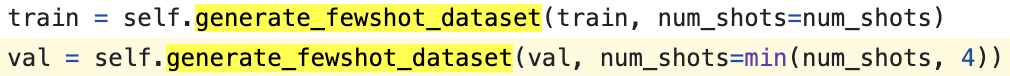
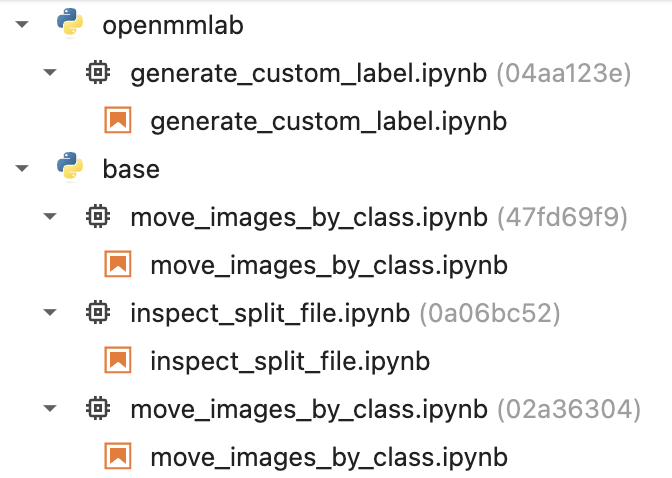
datasets/medfm\_chest.yaml:

https://drive.google.com/file/d/1a4P2Twh7ZCSwS4aPsOZtYX3Y-ClQHMBM/view?usp=drive\_link

https://drive.google.com/file/d/1LsEYcJZj\_5tkyvUsgjnI-81e1plmuh9D/view?usp=drive\_link

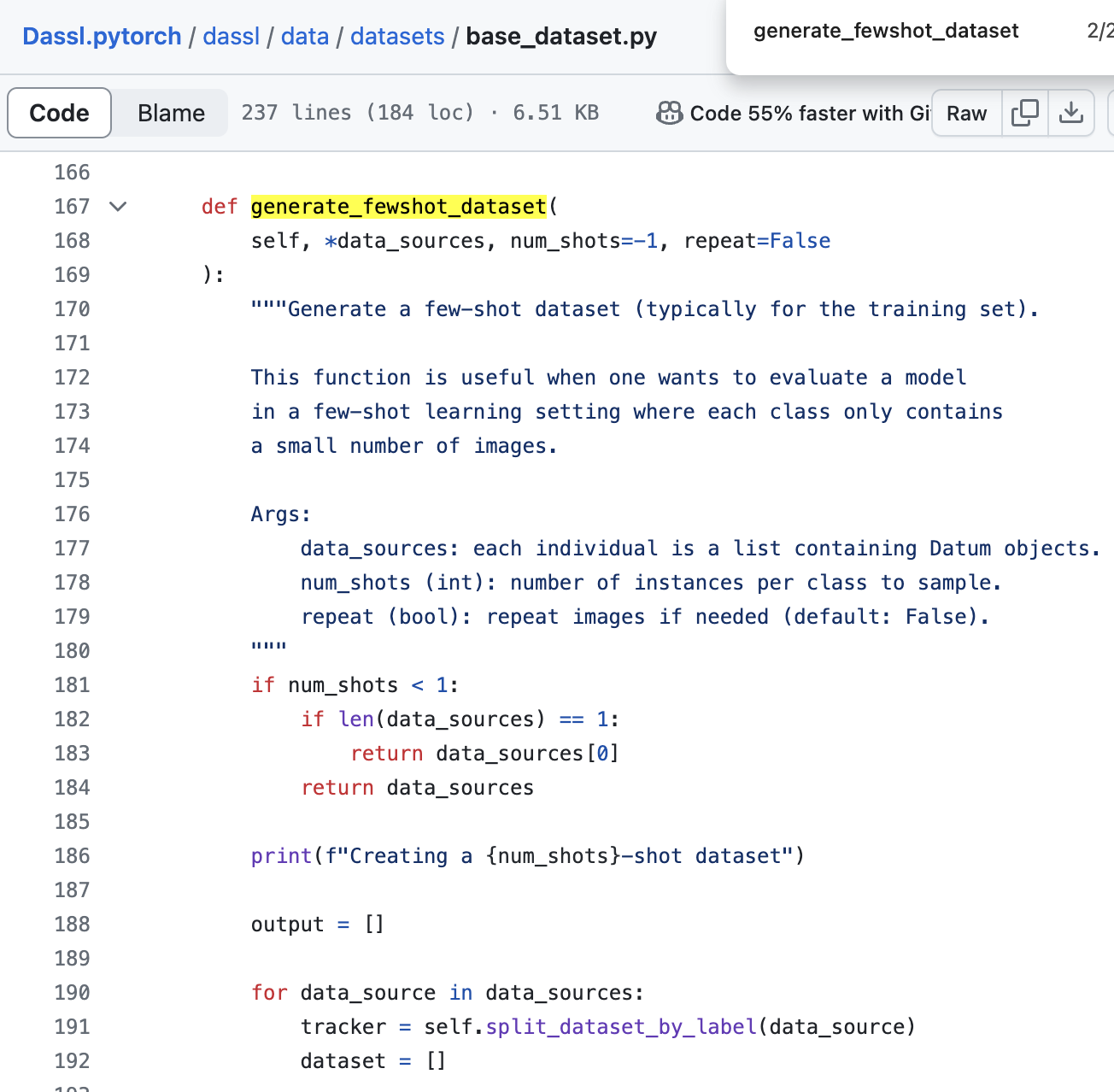
gdown https://drive.google.com/uc?id=1a4P2Twh7ZCSwS4aPsOZtYX3Y-ClQHMBM

gdown https://drive.google.com/uc?id=1LsEYcJZj\_5tkyvUsgjnI-81e1plmuh9D



16 -> 32 because of two classes

4 -> 8 because of two classes



mmpretrain

* ~~80% supervised learning 40 epochs (V) mAP, ACC~~
* 20% transfer learning 40 epochs (V) ~~mAP~~, ACC
* Few shot learning (?) mAP

CoCoOP

* Few shot learning
  + With prompt engineering
    - 1-shot ACC
    - 5-shot ACC
    - 10-shot ACC
  + Without prompt engineering
    - 1-shot ACC
    - 5-shot ACC
    - 10-shot ACC

改進：

* 80% supervised learning多加ACC
* CoCoOp多加mAP
* 使用DualCoOp以及其他multilabel CoOp
* 測試prompt的有效度
* Reproduce baseline
* Introduction
  + Data Source
  + Dataset
  + EDA
* Methodology
  + Data Preprocessing -> Preparation
    - Multilabel classification -> Multiple binary classification
  + Configuration
  + Experiments and Results -> [Heading 1]
* Conclusion
  + Dicussion
  + Future Work