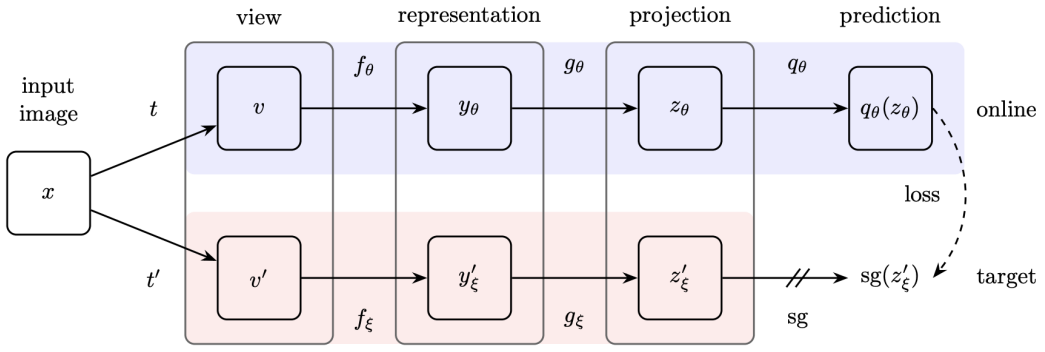
DLCV HW1 Report

學號：r13921068

姓名：吳家萱

Problem 1: Self-Supervised Pre-training for Image Classification

1. Describe the implementation details of your SSL method for pre-training the ResNet50 backbone.



SSL method: 我使用的是助教推薦的方法 Bootstrap Your Own Latent, BYOL (Github source)

Data augmentation:

```
self.transform = transforms.Compose([
    transforms.Resize((128, 128)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),
])
```

我的 pretrain 參數選擇如下：

Learning Rate	Optimizer	Batch Size	Epochs
2e-4	Adam	64	350

2. Please conduct the Image classification on Office-Home dataset as the downstream task. Also, please complete the following Table, which contains different image classification setting, and discuss/analyze the results.
   以下是各種 Setting 在 Office-Home dataset 進行的圖像分類實驗的結果與分析：

Setting	Pre-training (Mini-ImageNet)	Fine-tuning (Office-Home dataset)	Validation accuracy (Office-Home dataset)
A	-	Train full model (backbone + classifier)	0.4606
B	w/ label (TAs have provided this backbone)	Train full model (backbone + classifier)	0.5468
C	w/o label (Your SSL pre-trained backbone)	Train full model (backbone + classifier)	0.5123
D	w/ label (TAs have provided this backbone)	Fix the backbone. Train classifier only	0.2783
E	w/o label (Your SSL pre-trained backbone)	Fix the backbone. Train classifier only	0.2291

- 結果分析：
  1. Setting A：由於未經過 pretrain，模型需要從頭學習特徵，因此 validation accuracy 相對 Setting B & Setting C 較低，僅有 0.4606。
  2. Setting B：使用由助教提供的 supervised pretrain model 進行訓練，validation accuracy 明顯較高，達到 0.5468。
  3. Setting C：我使用了 BYOL 作為 self-supervised pretrain SSL，在最後的 validation accuracy 為 0.4901，低於 Setting B，且 Setting C 在

fine-tune 的過程中收斂的速度也比 Setting B 要慢。

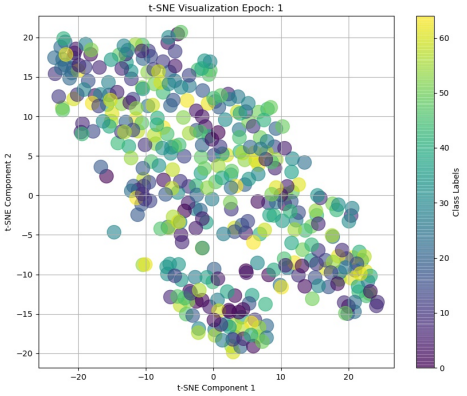
4. Setting D & Setting E：由於這兩者皆只訓練 classifier，相較於在其他條件皆相同下的 Setting B & C， 甚至是 Setting A，Setting D & E 的 validation accuracy 都明顯較差。

- 總結：

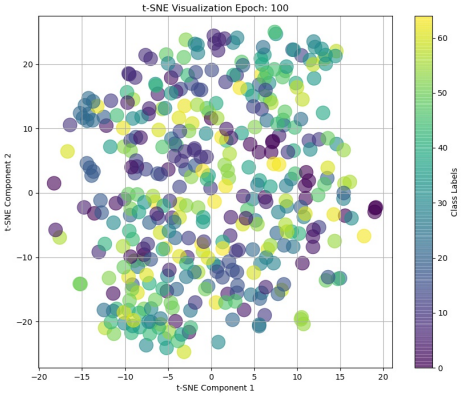
  在以上各條件的比較下，可以發現是否使用 pretrain model，雖然會影響 performance，但還是能有不錯的 validation accuracy；然而，是否只訓練 classifier，則會非常顯著地影響 performance；而 pretrain model 使用 supervised 或 self-supervised 的方式進行預訓練，在 performance 上雖然會有一些差異，但兩者的 validation accuracy 都相當好，我認為兩者都是很有幫助的 pretrain model 方法。

3. Visualize the learned visual representation of setting C on the train set by implementing t-SNE (t-distributed Stochastic Neighbor Embedding) on the output of the second last layer. Depict your visualization from both the first and the last epochs. Briefly explain the results.

- t-SNE Visuallization Epoch 1



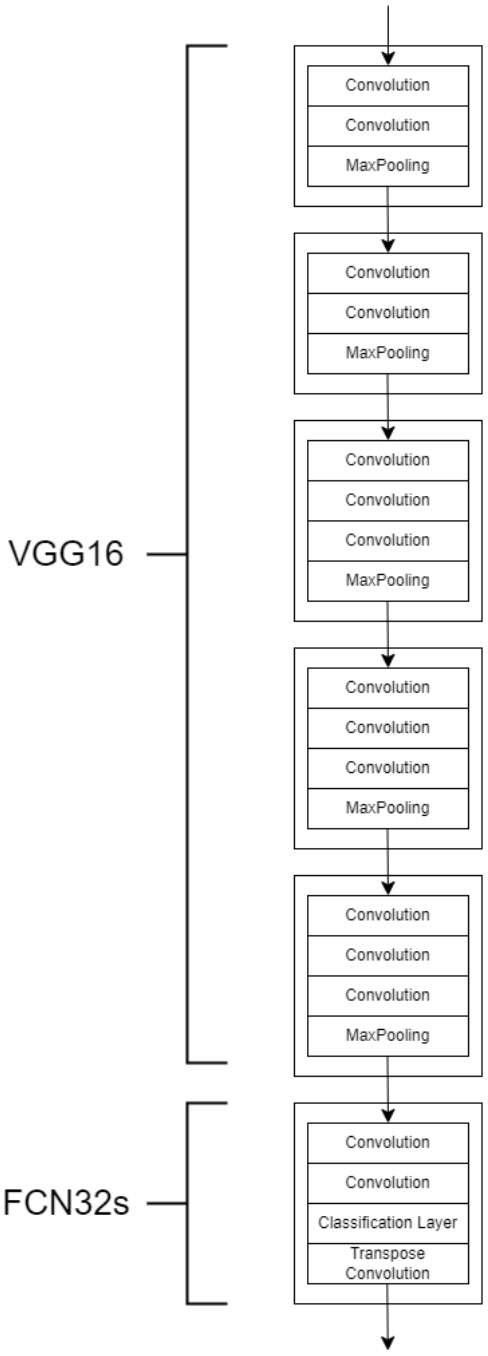
- t-SNE Visuallization Epoch 100



- 在 epoch 1 中，t-SNE 圖中各類別之間的區分較不明顯，許多類別的資料點混在一起，模型尚未能區分不同類別的特徵。
- 在 epoch 100 中，t-SNE 圖中各類別之間的區分變得較為明顯，儘管不同的 class labels 仍有重疊，但已經可以觀察到不同類別的點逐漸分開形成更明顯的一群一群，代表模型學到了如何區分不同類別的特徵，能夠更好地將不同類別的資料分開。

Problem 2: Semantic Segmentation

1. Draw the network architecture of your VGG16-FCN32s model (model A).



2. Draw the network architecture of the improved model (model B) and explain it differs from your VGG16-FCN32s model.

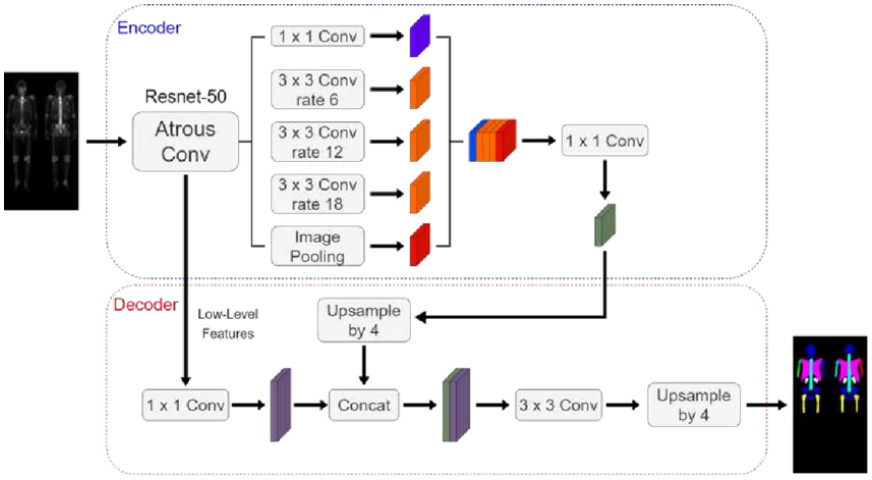
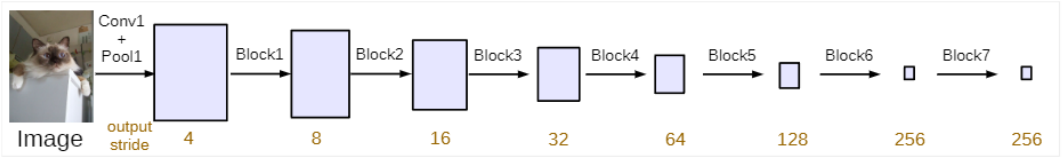


Image Source

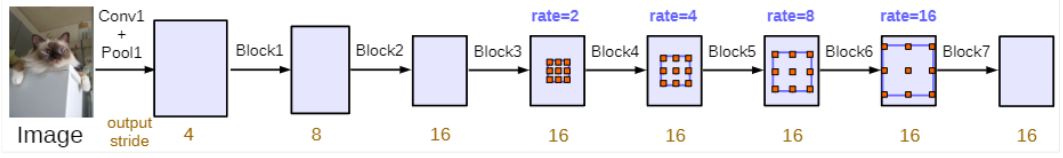
Model from: L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, 'Rethinking Atrous Convolution for Semantic Image Segmentation', CoRR, vol. abs/1706.05587, 2017

Pytorch model source: Semantic Segmentation deeplabv3_resnet50

- 我使用的 improved model 為 DeepLabV3-ResNet50，DeepLabV3 引入了 Atrous Convolution (空洞卷積) 和 Atrous Spatial Pyramid Pooling, ASPP 模組，可以捕獲多尺度的上下文資訊，如下圖所示。
  - Atrous Convolution: 在卷積核的元素之間引入「空洞」來擴大元素之間的間隔，擴大卷積的視野範圍但不增加計算量或減少輸出圖像的分辨率。



(a) Going deeper without atrous convolution.



(b) Going deeper with atrous convolution. Atrous convolution with $rate > 1$ is applied after block3 when $output\_stride = 16$.

Figure 3. Cascaded modules without and with atrous convolution.

  - ASPP: 透過多尺度的空洞卷積來捕捉圖像中的不同尺度資訊。

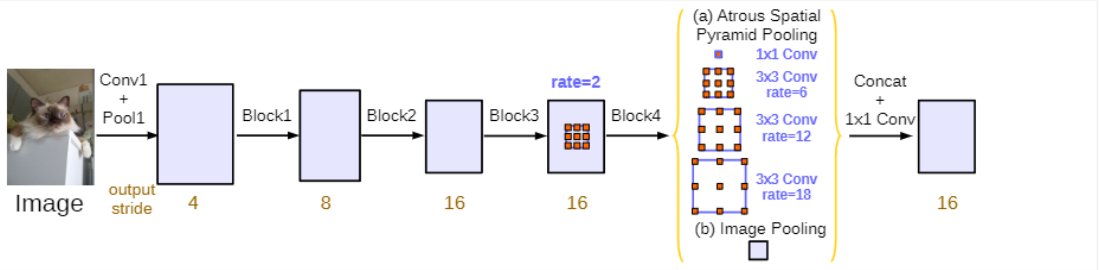


Figure 5. Parallel modules with atrous convolution (ASPP), augmented with image-level features.

Image Source

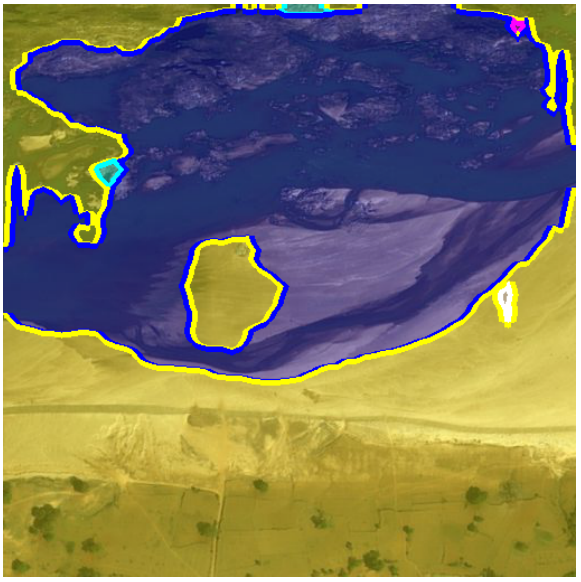
- Difference from VGG16-FCN32s:
  - VGG16-FCN32s 只能處理單尺度特徵圖，而 DeepLabV3 透過 ASPP 可以處理不同尺度的特徵，並且能更有效地進行 semantic segmentation。

3. Report mIoUs of two models on the validation set.

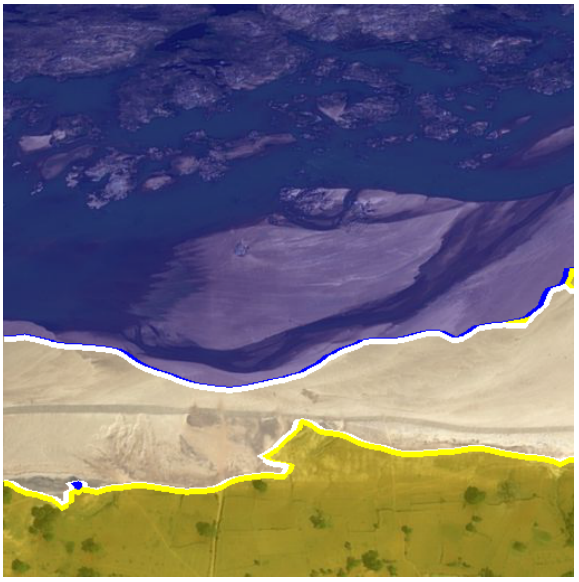
model A (VGG16-FCN32s)	model B (DeepLabV3-ResNet50)
0.65493	0.75882

4. Show the predicted segmentation mask of “validation/0013_sat.jpg”, “validation/0062_sat.jpg”, “validation/0104_sat.jpg” during the early, middle, and the final stage during the training process of the improved model.

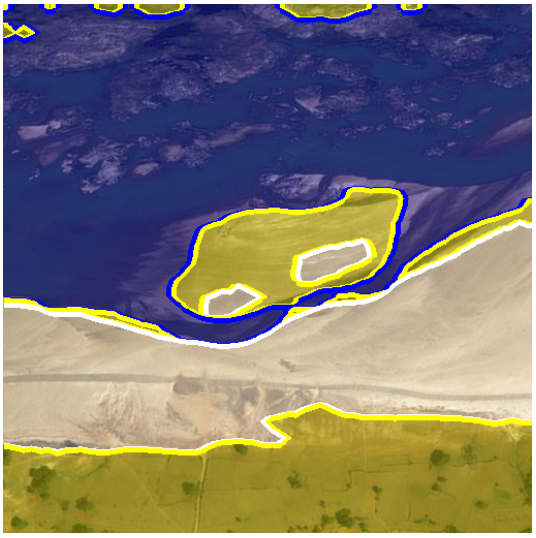
- [epoch 1] validation/0013

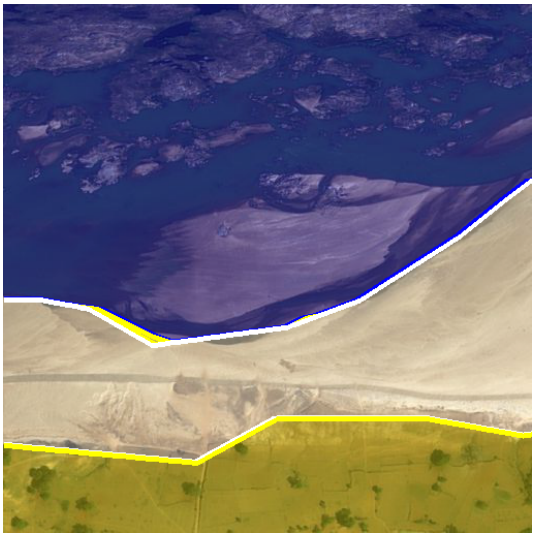


- [epoch 50] validation/0013

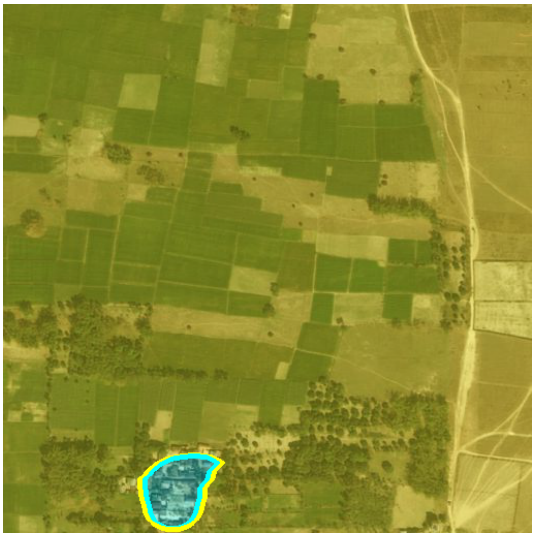


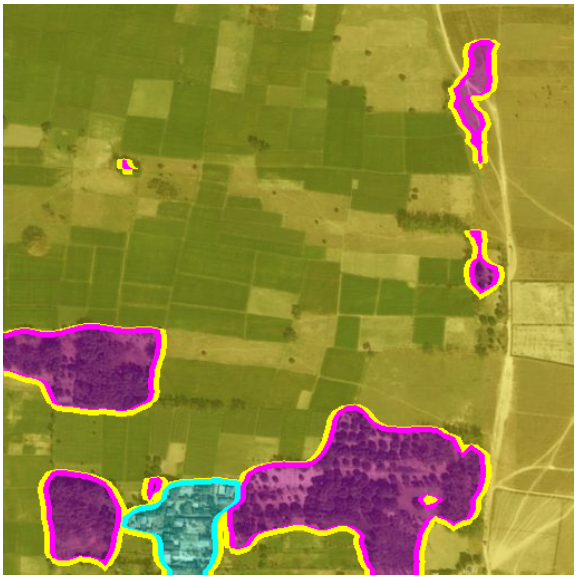
- [epoch 100] validation/0013
- [label] validation/0013



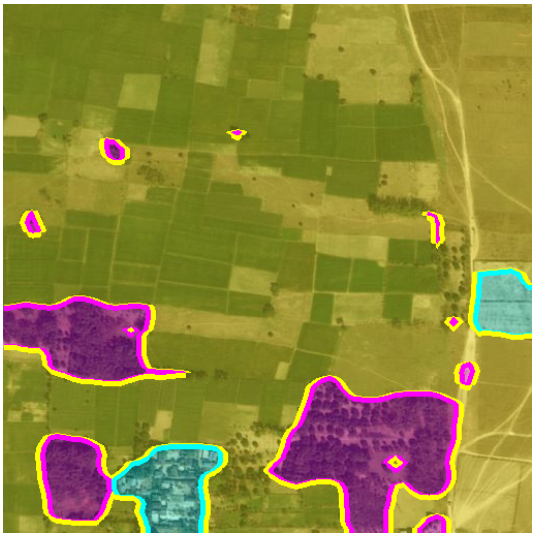


- [epoch 1] validation/0062
- [epoch 50] validation/0062





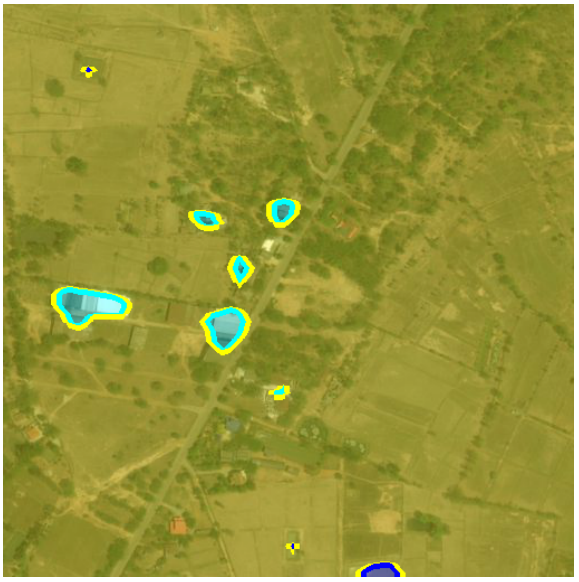
- [epoch 100] validation/0062
- [label] validation/0062



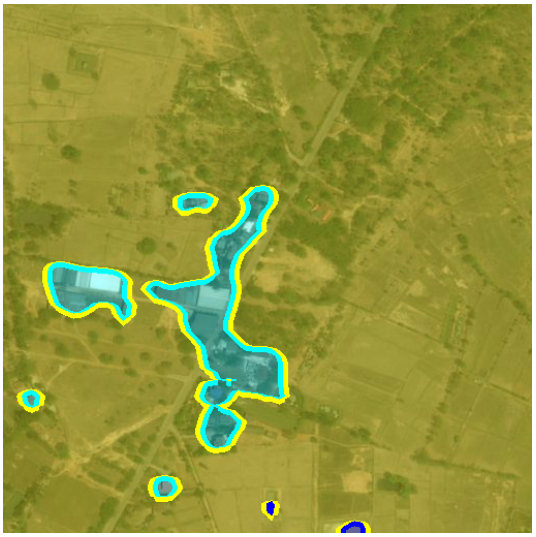
- [epoch 1] validation/0104

- [epoch 50] validation/0104

- [epoch 100] validation/0104



- [label] validation/0104

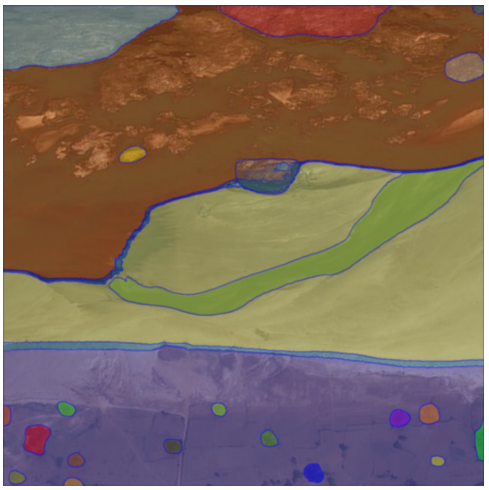


5. Use segment anything model (SAM) to segment three of the images in the validation dataset, report the result images and the method you use.

- 我使用的方式為 Meta 的網站 (https://segment-anything.com/demo) ，以下分別為三張 validation 的 SAM predicted segmentation mask:

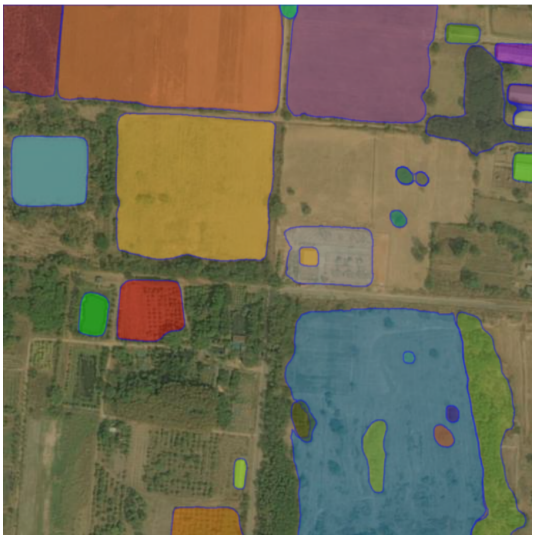
- validation/0013_sat.jpg

- [SAM] validation/0013_sat.jpg

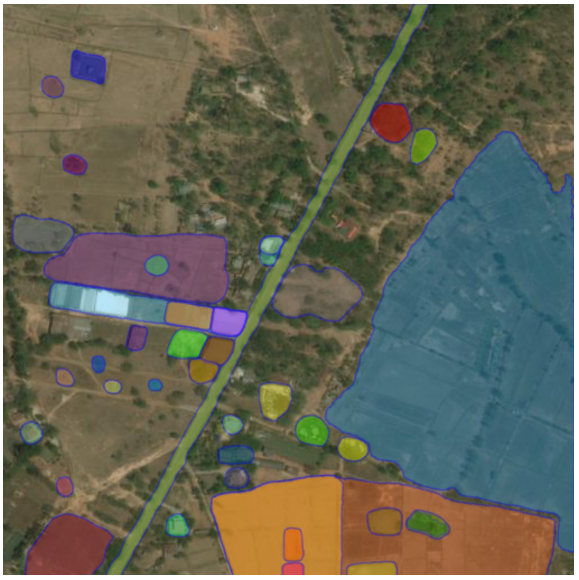


- validation/0062_sat.jpg

- [SAM] validation/0062_sat.jpg



- validation/0104_sat.jpg
- [SAM] validation/0104_sat.jpg



- 分析:
  - 雖然我只是使用線上的 SAM 工具來進行分析，但我覺得光是透過線上 SAM 生成出的 predicted segmentation mask 結果便相當之好，在直觀上比起我於 model A (VGG16-FCN32s) 和 model B (DeepLabV3-ResNet50) 訓練結束後生成的 predicted segmentation mask 還要更精準的預測了每個區塊。

Reference

Bootstrap Your Own Latent (BYOL), in Pytorch

Long et al., “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015

DeepLabv3
Building Blocks for Robust Segmentation Models

L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, ‘Rethinking Atrous Convolution for Semantic Image Segmentation’, CoRR, vol. abs/1706.05587, 2017

Pytorch model source: Semantic Segmentation deeplabv3_resnet50

Meta AI SAM

ChatGPT-4o