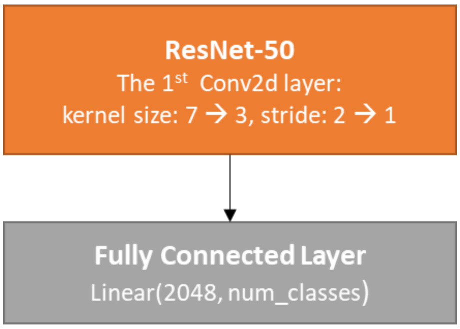
# DLCV HW1 Report

R11921078 李鎮宇

## Problem 1: Image Classification

1. **(2%) Draw the network architecture of method A or B.**
model A:



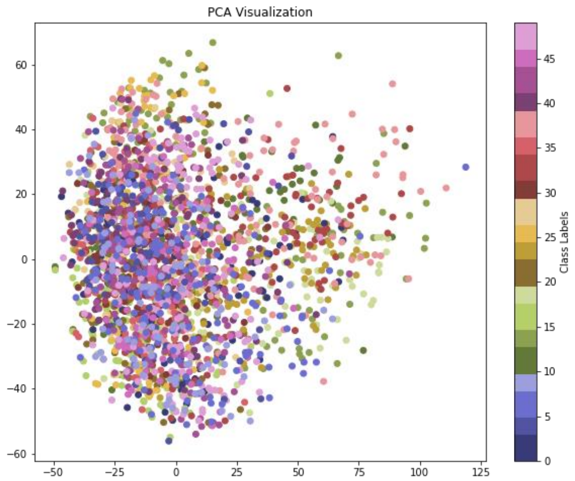
2. **(1%) Report accuracy of your models (both A, B) on the validation set**
model A: 0.2904
model B: 0.9016

3. **(2%) Report your implementation details of model A.**
使用 ResNet-50 作為模型，考量輸入影像大小為 32*32，將第一層 conv2d 調整成 kernel_size=(3, 3), stride=(1, 1)，讓 kernel 看的範圍不要太大、一次不要移動太多；最後一層 fully connected layer 的 output features 改為 50。loss function 使用 cross entropy loss；optimizer 使用 Adam，weight decay 設 1e-4；learning rate 設定 0.0001。

4. **(3%) Report your alternative model or method in B, and describe its difference from model A.**
在 model B 中我使用 EfficientNet_V2_L 作為模型架構，搭配使用 ‘DEFAULT’(=’IMAGENET1K_V1’)作為 pre-trained weights。loss function, optimizer 和 learning rate 都和 model A 設定相同，但我另外加上 MultiStepLR，讓模型訓練的過程中每 5 個 epochs，learning rate 就乘上 0.1，以此遞減學習的速率。影像的部分我 resize 成 224*224 輸入，也使用 RandomHorizontalFlip, RandomRotation, RandomCrop 對 training dataset 做影像增強。

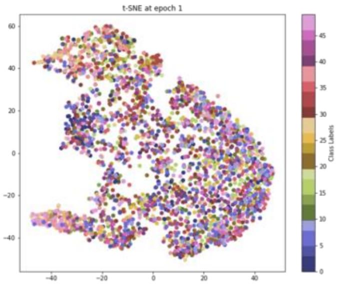
5. **(3%) Visualize the learned visual representations of model A on the validation set by implementing PCA (Principal Component Analysis) on the output of the second last layer. Briefly explain your result of the PCA visualization.**

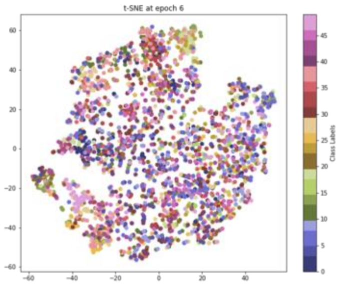
   從結果來看，降維後的資料點幾乎混成一團，看不出cluster分布，因為PCA在降維過程中會損失太多的資訊。

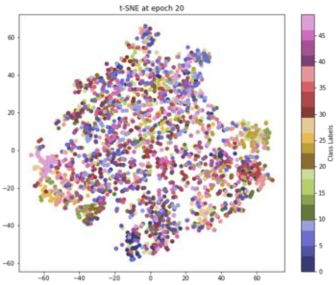


6. **(4%) Visualize the learned visual representation of model A, again on the output of the second last layer, but using t-SNE (t-distributed Stochastic Neighbor Embedding) instead. Depict your visualization from three different epochs including the first one and the last one. Briefly explain the above results.**

   t-SNE的降維分群效果較PCA清楚，以下分別是epoch=1, 6, 20(last)的結果，可以發現在初期資料幾乎還是混成一團，在epoch=6和20時，可以看到已經有一些零散的小群形成，但仍然不明顯。因為沒有加入pre-trained weight，導致模型不容易訓練起來。







Problem 2: Self-Supervised Pre-training for Image Classification

1. **(5%) Describe the implementation details of your SSL method for pre-training the ResNet50 backbone.**
   依循 BYOL 的 github 程式碼[1]與步驟訓練 ResNet50 backbone，在本題中沒有 load pre-trained weights。data augmentation 使用 Resize(128, 128)、CenterCrop(128)；optimizer 使用 Adam()；learning rate 設定為 0.0001；loss function 由正樣本損失和負樣本損失組成，輸入影像會被分成兩種 view (a 和 b)，正樣本損失鼓勵將 view a, view b 的特徵變得更相似，負樣本損失則是把與這兩個 view 特徵不同的樣本分開。

   [1] BYOL github: https://github.com/lucidrains/byol-pytorch

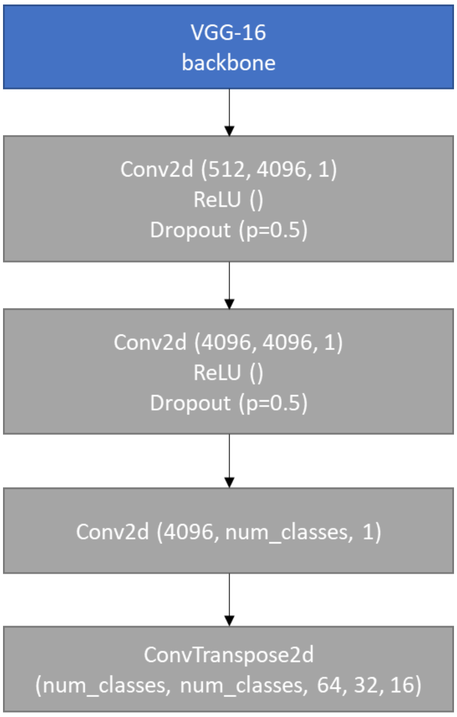
2. **(20%) Please conduct the Image classification on Office-Home dataset as the downstream task. Also, please complete the following Table, which contains different image classification setting, and discuss/analyze the results.**

| Setting | Pre-training (Mini-ImageNet) | Fine-tuning (Office-Home dataset) | Validation accuracy (Office-Home dataset) |
|---|---|---|---|
| A | - | Train full model (backbone + classifier) | 0.0711 |
| B | w/ label (TAs have provided this backbone) | Train full model (backbone + classifier) | 0.2745 |
| C | w/o label (Your SSL pre-trained backbone) | Train full model (backbone + classifier) | 0.3186 |
| D | w/ label (TAs have provided this backbone) | Fix the backbone. Train classifier only | 0.1225 |
| E | w/o label (Your SSL pre-trained backbone) | Fix the backbone. Train classifier only | 0.2157 |

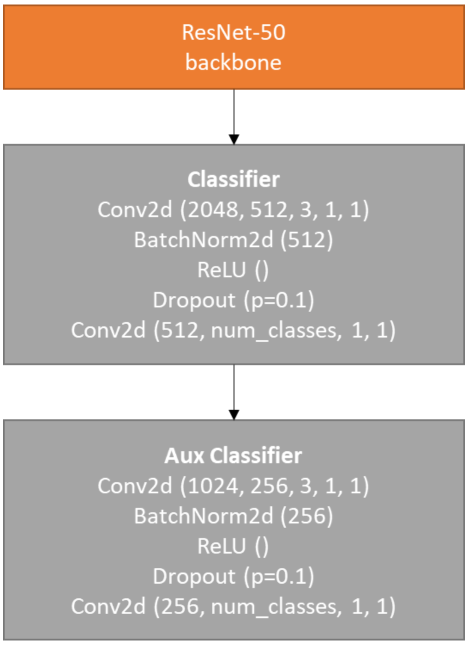
A 沒有任何的 pre-trained weights 或是訓練過的 feature extractor，因此訓練效果最差。從 B vs D 和 C vs E 的結果來看，若一開始 fix backbone，只訓練 classifier 的效果明顯較全部一起訓練來的差，推測是 pre-training 時所使用的資料集與後續 fine-tuning 時不同，因此 backbone 後續也要一起對目標任務的資料特徵進行微調，分類效果才會變好。在 B vs C 當中，發現兩者的準確度相近，但 C 略高一些，說明即便是在資料沒有標記的情況下，SSL 的效果並不會比 SL (supervised learning)來得差。

## Problem 3: Semantic Segmentation

1. **(3%) Draw the network architecture of your VGG16-FCN32s model (model A).**



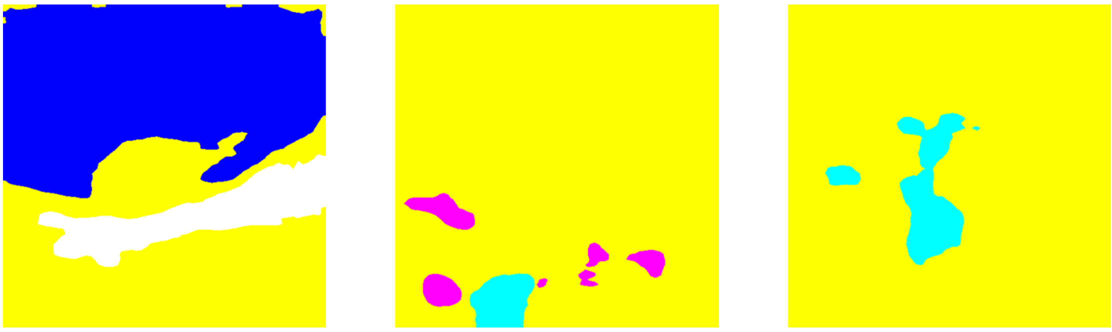
2. **(3%) Draw the network architecture of the improved model (model B) and explain it differs from your VGG16-FCN32s model.**
backbone 改用 ResNet-50，且有 load pre-trained weights，由兩塊 classifier 組成 FCN，與 model A 之 FCN 不同之處在於沒有使用 ConvTranspose2d 作 upsampling。



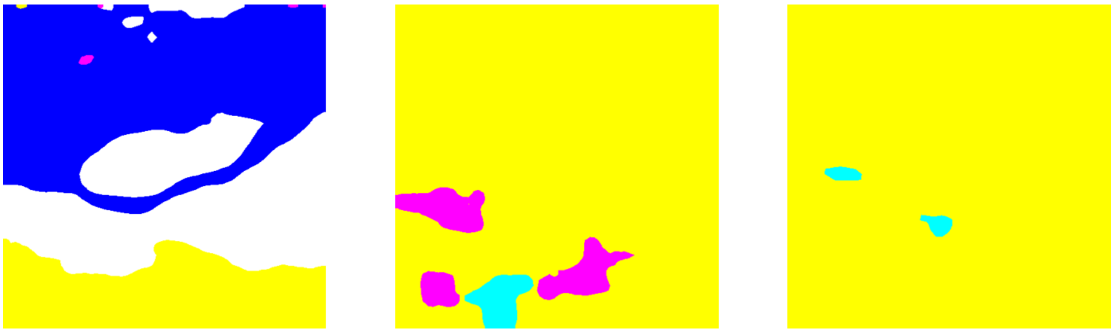
3. **(1%) Report mIoUs of two models on the validation set.**
model A: 0.679024
model B: 0.694208

4. **(3%) 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.**
由左至右分別是 0013_mask.png, 0062_mask.png, 0104_mask.png。

- early stage (epoch = 1):



- middle stage (epoch = 7):



- final stage (epoch = 14):