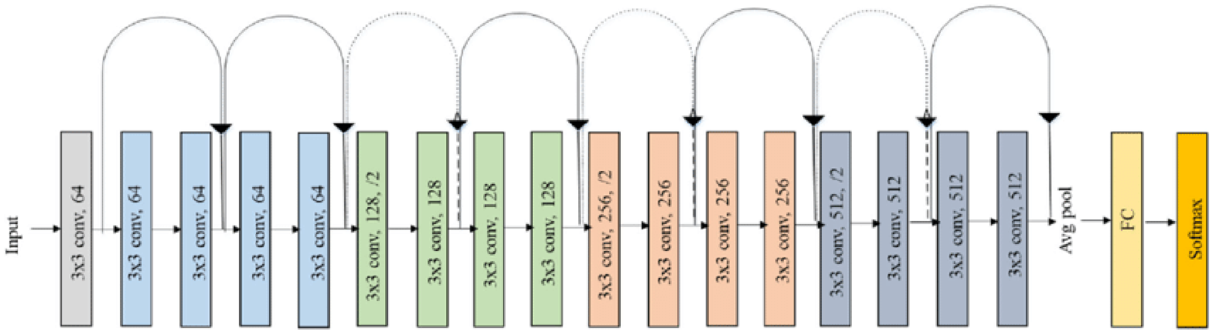
DLCV hw1

ID: r12922075

Name: 吳韋論

Image Classification

1. Network architecture of method A



I use ResNet-18 as my model A. The core architecture is composed of four groups of residual blocks, each containing multiple residual blocks with two 3x3 convolutional layers. After these blocks, a fully connected layer is used for class-specific(50 classes) output.

2. Accuracy of models (both A, B) on the validation set

	validation accuracy
model A	74.96%
model B	90.20%

3. Implementation details of model A

Data Augmentation:

- Resized images to 224x224
- Random resized crop and horizontal flip for data augmentation

- Normalized with mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225]

Model Architecture & Learning rate:

- Manually crafted ResNet-18 architecture
- 100 epochs with a learning rate (lr) of 1e-4

Loss Function & Optimizer:

- Cross-Entropy Loss for classification
- AdamW for gradient descent optimization

Training & Validation:

- Iterate the training data, compute loss, and optimize the model in the training loop.
- Evaluate performance at each epoch on the validation set.
- I record training loss/accuracy, validation loss/accuracy for each epoch.

4. Alternative model or method in B, and describe its difference from model A

Alternative Model in B:

Model B is based on the EfficientNetV2-S architecture

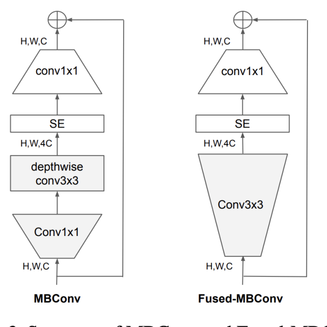
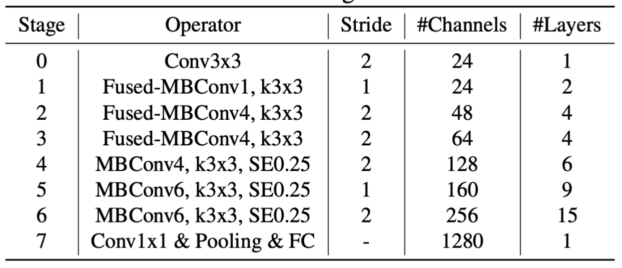


Figure 2. Structure of MBConv and Fused-MBConv.

Table 4. EfficientNetV2-S architecture – MBConv and Fused-MBConv blocks are described in Figure 2.



Stage	Operator	Stride	#Channels	#Layers
0	Conv3x3	2	24	1
1	Fused-MBConv1, k3x3	1	24	2
2	Fused-MBConv4, k3x3	2	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv6, k3x3, SE0.25	1	160	9
6	MBConv6, k3x3, SE0.25	2	256	15
7	Conv1x1 & Pooling & FC	-	1280	1

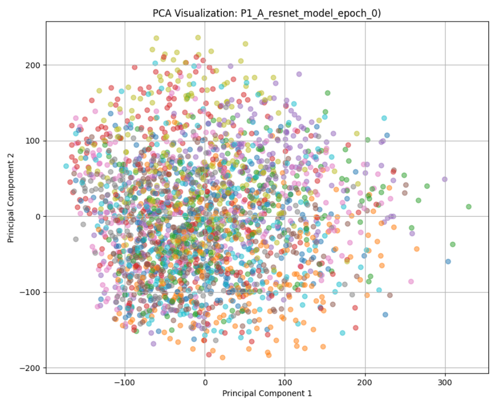
> The architecture of EfficientNetV2-S (Source: EfficientNetV2 paper)

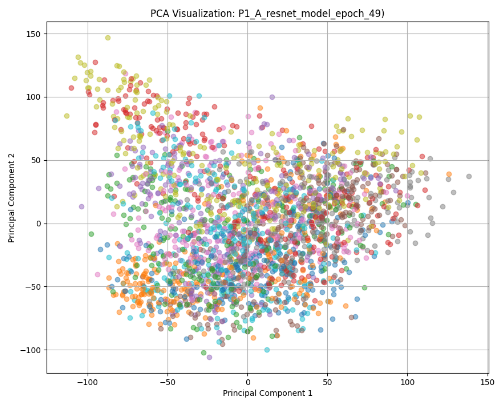
Difference from model A(resnet18):

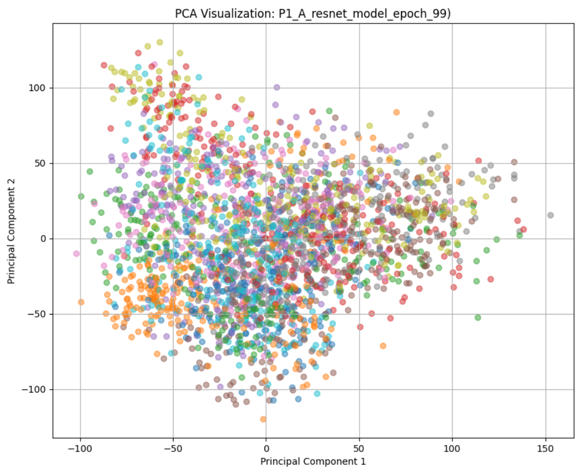
- Architecture: MBConv and Fused-MBConv offers flexibility in balancing model complexity and training speed
- Pretrained weights: EfficientNet_V2_S_Weights.IMAGENET1K_V1
- Training: 30 epochs with an initial lr of 5e-5

To sum up, Model B has pretrained weights and an effective model architecture, it outperforms Model A with just 30 epochs of training.

5. Visualize the learned visual representations of model A by PCA on the output of the second last layer



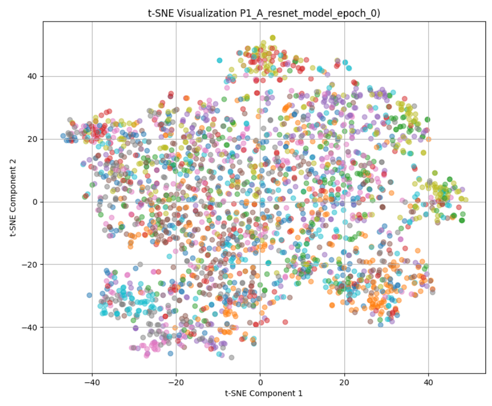


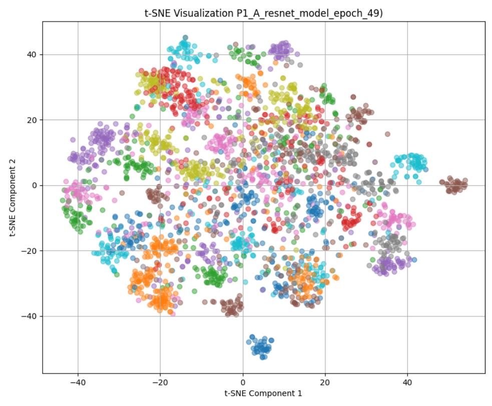


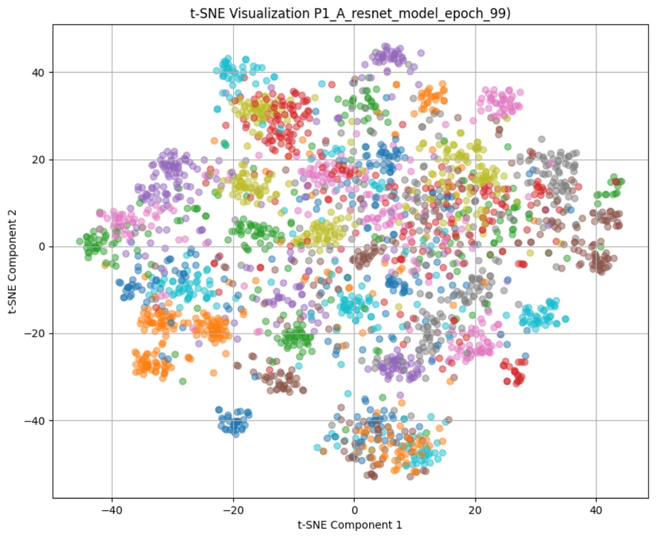
The figures are PCA output of the second last layer for Model A at epochs 1, 50, and 100. The visual representations are not highly distinguishable, with only the orange and red points are clearly separable.

It may be the limitations of PCA, which is a linear technique optimized for data with linear structures. Hence, its performance may be less effective for classifying data with non-linear structures.

6. Visualize the learned visual representations of model A by t-SNE on the output of the second last layer







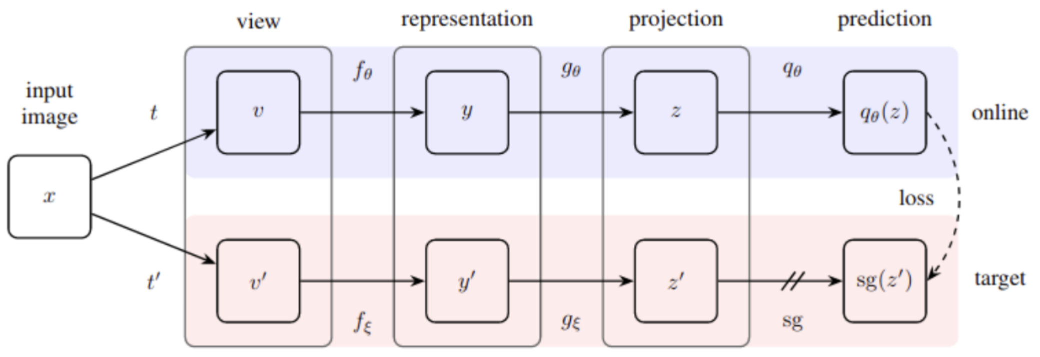
The figures are t-SNE output of the second last layer for Model A at epochs 1, 50, and 100. t-SNE has distinct differences, with most data points becoming separable.

This improvement may be t-SNE's nonlinear approach, emphasizing the preservation of pairwise similarities in a lower-dimensional space, making it more effective in visual representations.

Self-Supervised Pre-training for Image Classification

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

SSL method: BYOL + change fully connected layer in resnet50



- Online and target networks: Both networks have the same architecture, and their primary purpose is to learn representations from the input data.
- Contrastive Learning: Maximize the similarity between the representations the online network and the target network.
- Because we need to train from scratch, I change fully connected layer in resnet50 with num_class=64 to reduce the numbers of parameter.

Data Augmentation: resize + default setting in BYOL

- Resized images to 128x128
- Random colorJitter with prob 0.3
- Random gray scale with prob 0.2
- Random HorizontalFlip
- Random GaussianBlur with prob 0.2
- Random Resized Crop
- Normalized with mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225]

Learning rate scheduler: warmup + CosineAnnealingLR

- First 50 epochs: warmup scheduler
- 100~800 epochs: CosineAnnealingLR

Optimizer: Adam with a learning rate of 3e-4

Batch size: 128

2. Different image classification setting, and discuss/analyze the results

Setting	Pre-training	Fine-tuning	Validation accuracy
A	-	Train full model	44.58%
B	w/ label (TA backbone)	Train full model	49.01%
C	w/o label (SSL backbone)	Train full model	50.74%
D	w/ label (TA backbone)	Fix backbone	28.33%
E	w/o label (SSL backbone)	Fix backbone	30.54%

Train full model (A, B, and C):

Setting A is lacks of pre-training, leads to lower accuracy in fine-tuning because it needs to train with random initial weights.

In settings B and C with pre-training, setting C has the more improvement in accuracy. There are two possible reasons behind this observation.

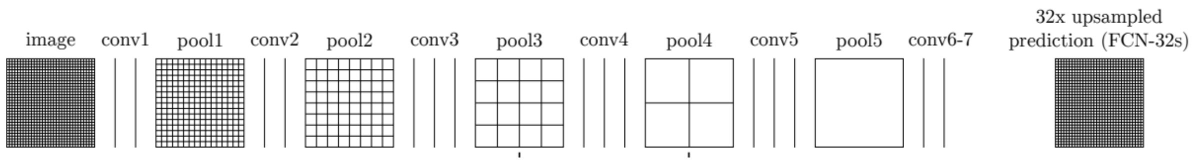
- **Dataset dissimilarity:** The mini-image dataset and the office-home dataset exhibit significant dissimilarities. In this case, SSL pre-training focuses on capturing more generalized features that can bridge the gap between the datasets.
- **Image Resolution:** The images in the mini-image dataset have smaller pixel dimensions. This may result in supervised pre-training (TA) overfitting to the mini-image dataset. In contrast, SSL, which learns representations without labels, can offer advantages in downstream tasks by focusing on more robust and generalizable features.

Fix backbone (D, E):

In settings D and E, where the backbone is fixed, the model can only change weights at the fully-connected layers. Because the pre-training dataset is quite different from the dataset used for fine-tuning, both D, E are hard to improve validation accuracy.

Semantic Segmentation

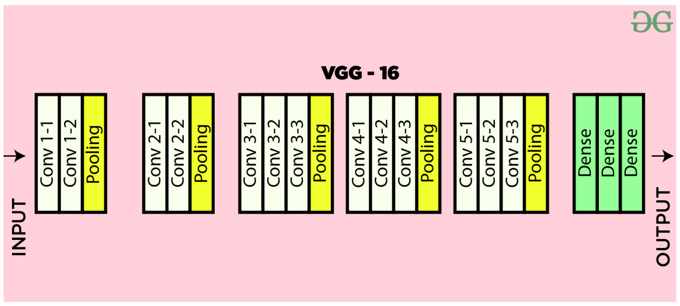
1. Network architecture of VGG16-FCN32s model



> The architecture of VGG16-FCN32s (Source: VGG16-FCN32s paper)

Input size: 512x512

Conv1 to pool5: Feature extractor from pretrained VGG16, as shown in the VGG16 architecture diagram below(excluding fc layer).



Conv6-7: Convolutional layer followed by ReLU activation and dropout.

32x upsample prediction: Transpose convolution to restore the spatial dimensions and provide the segmentation prediction.

2. Network architecture of the improved model and explain it differs from VGG16-FCN32s model

Model architecture

I use DeepLabV3-ResNet50 as my improved model.

- **Block1~4**: Resnet50 as backbone
- **ASPP**: Convolutions with different dilation rates (or atrous rates) to analyze image content at various scales
- **Concat+1x1 Conv**: Concat feature maps and get predictions from 1x1 Conv

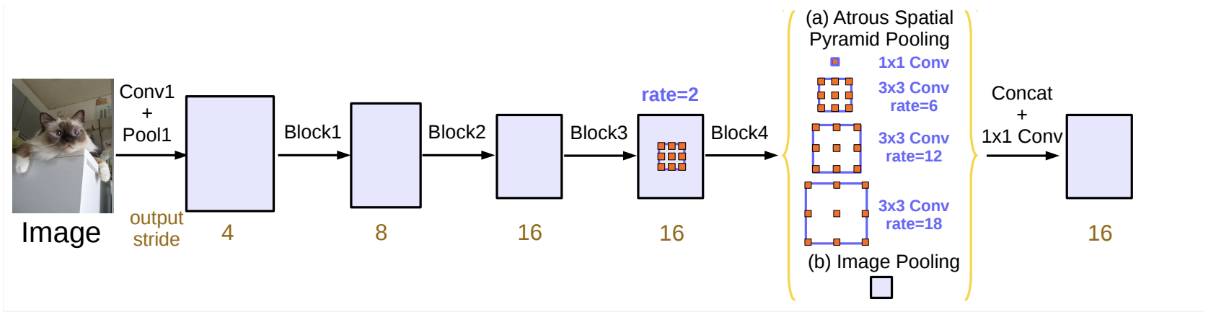
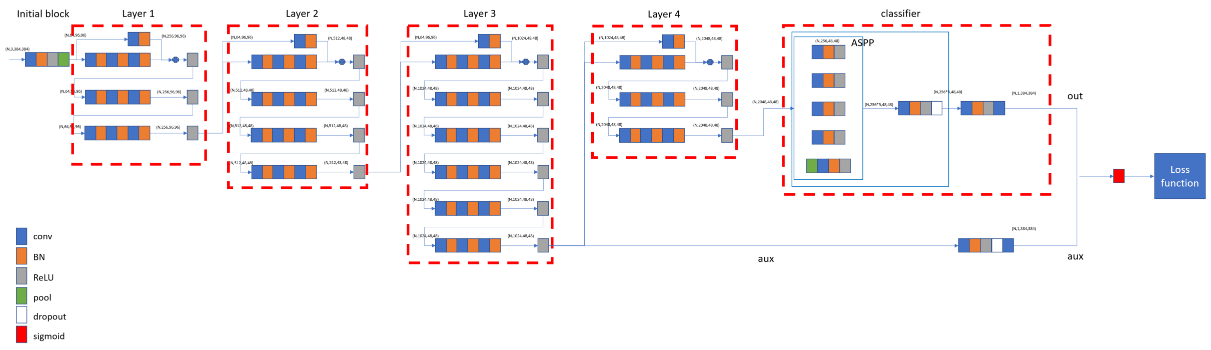


Figure 5. Parallel modules with atrous convolution (ASPP), augmented with image-level features.

Picture below with more details



> The architecture of DeepLabV3-ResNet50 (Source: paper and blog)

Difference from VGG16-FCN32s

- **Architecture**: VGG16-FCN32 has only a single-scale feature map. However, DeepLabv3-ResNet50 includes the ASPP module, which is designed to capture multi-scale contextual information.
- **Loss function**: Because I noticed that class 2 wasn't being recognized effectively during training, I switched to using focal loss, which assigns higher weights to misclassified class.

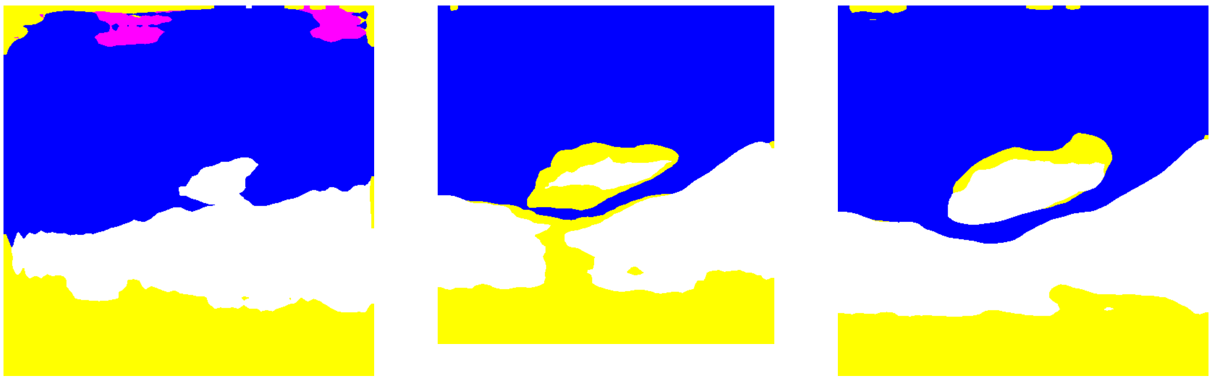
3. mIoUs of two models on the validation set

	validation mIOUs
VGG16-FCN32	0.7253
DeepLabV3-ResNet50	0.7597

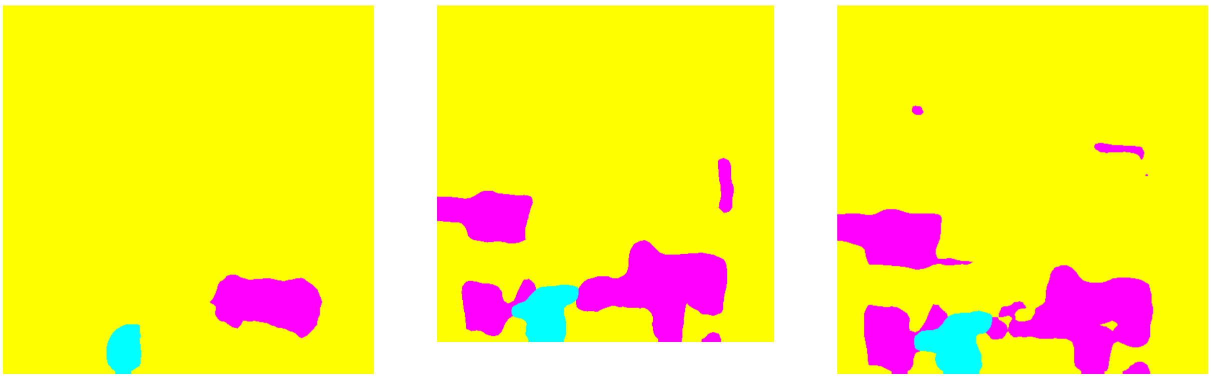
4. Predicted segmentation masks during the early, middle, and the final stage during the training process of the improved

model.

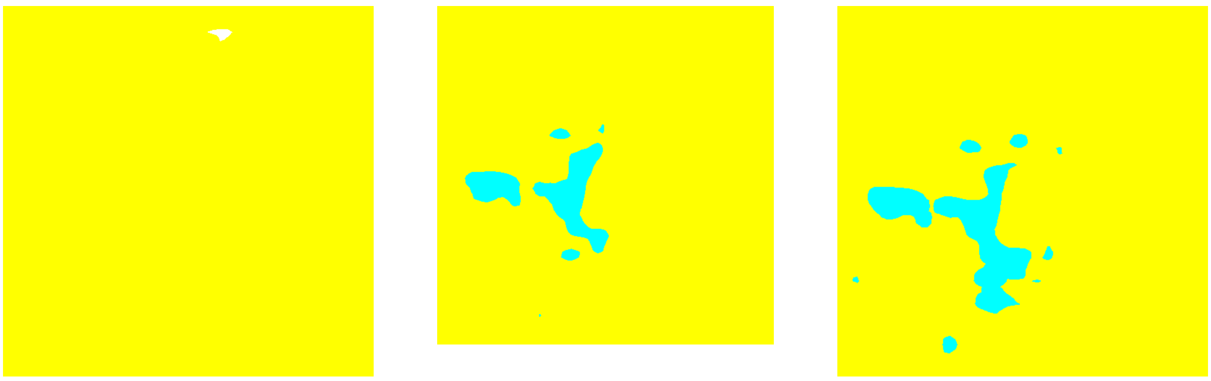
0013_mask.png (left to right: early, middle, final stage)



0062_mask.png (left to right: early, middle, final stage)



0104_mask.png (left to right: early, middle, final stage)



Starting from the middle stage, the model begins to more accurately capture the segmentation class, especially for the more challenging Class 2.