DLCV HW3

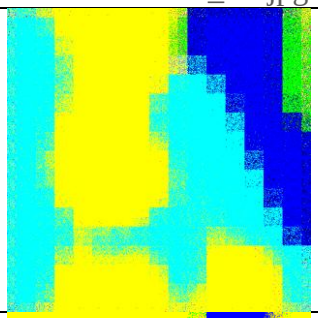
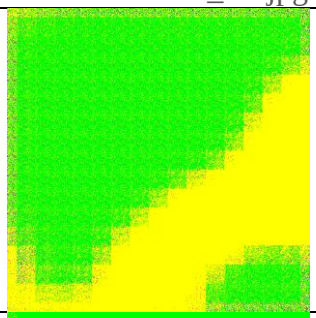
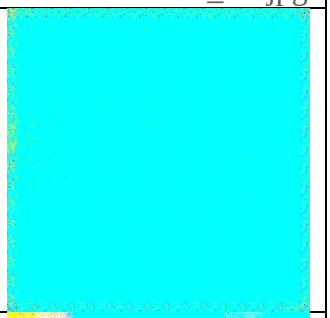
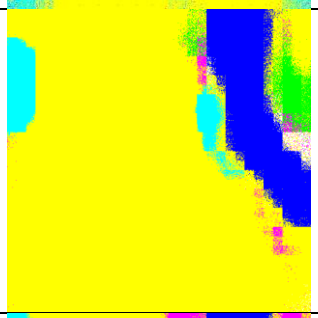
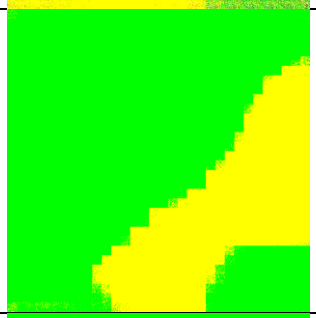
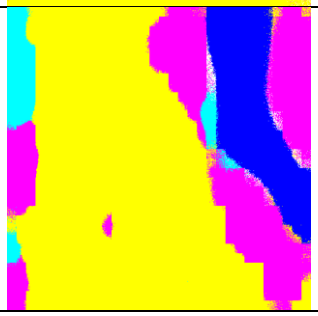
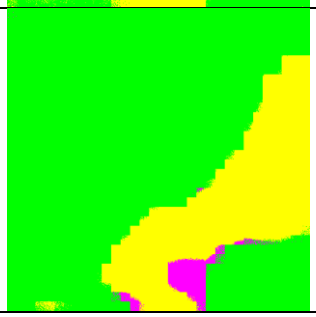
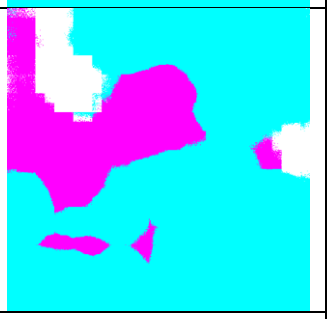
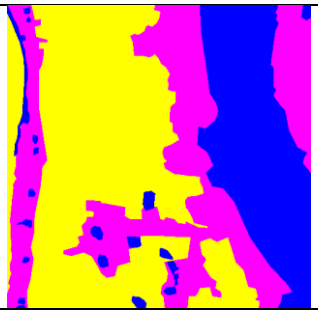
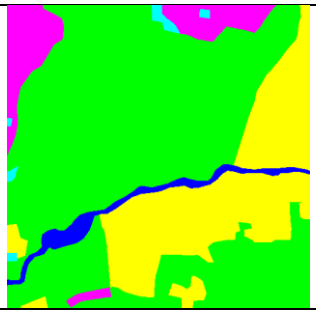
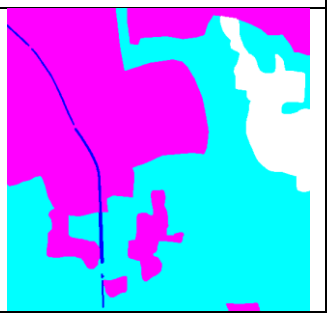
Name: 王冠驊 Dep.:電信碩二 Student ID:R05942102

1. (5%) Print the network architecture of your VGG16-FCN32s model.

Layer (type)	Output Shape	Param #
input_4 (InputLayer)	(None, 512, 512, 3)	0
batch_normalization_4 (Batch	(None, 512, 512, 3)	12
block1_conv1 (Conv2D)	(None, 512, 512, 64)	1792
block1_conv2 (Conv2D)	(None, 512, 512, 64)	36928
block1_pool (MaxPooling2D)	(None, 256, 256, 64)	0
block2_conv1 (Conv2D)	(None, 256, 256, 128)	73856
block2_conv2 (Conv2D)	(None, 256, 256, 128)	147584
block2_pool (MaxPooling2D)	(None, 128, 128, 128)	0
block3_conv1 (Conv2D)	(None, 128, 128, 256)	295168
block3_conv2 (Conv2D)	(None, 128, 128, 256)	590080
block3_conv3 (Conv2D)	(None, 128, 128, 256)	590080
block3_pool (MaxPooling2D)	(None, 64, 64, 256)	0
block4_conv1 (Conv2D)	(None, 64, 64, 512)	1180160
block4_conv2 (Conv2D)	(None, 64, 64, 512)	2359808
block4_conv3 (Conv2D)	(None, 64, 64, 512)	2359808
block4_pool (MaxPooling2D)	(None, 32, 32, 512)	0
block5_conv1 (Conv2D)	(None, 32, 32, 512)	2359808
block5_conv2 (Conv2D)	(None, 32, 32, 512)	2359808
block5_conv3 (Conv2D)	(None, 32, 32, 512)	2359808
block5_pool (MaxPooling2D)	(None, 16, 16, 512)	0
conv2d_10 (Conv2D)	(None, 16, 16, 1024)	4719616
dropout_7 (Dropout)	(None, 16, 16, 1024)	0
conv2d_11 (Conv2D)	(None, 16, 16, 1024)	1049600
dropout_8 (Dropout)	(None, 16, 16, 1024)	0
conv2d_12 (Conv2D)	(None, 16, 16, 7)	7175
conv2d_transpose_4 (Conv2DTr	(None, 512, 512, 7)	200704
activation_4 (Activation)	(None, 512, 512, 7)	0

Total params: 20,691,795
Trainable params: 20,691,789
Non-trainable params: 6

2. (10%) Show the predicted segmentation mask of “validation/0008_sat.jpg”, “validation/0097_sat.jpg”, “validation/0107_sat.jpg” during the early, middle, and the final stage during the training stage. (For example, results of 1st, 10th, 20th epoch)

Epoch	validation/0008_sat.jpg	validation/0097_sat.jpg	validation/0107_sat.jpg
1 early			
6 middle			
12 final			
Ground truth			

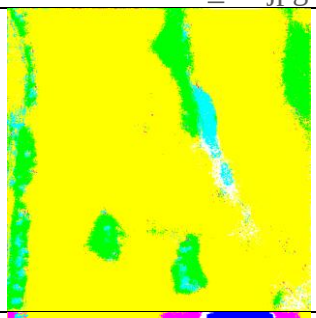
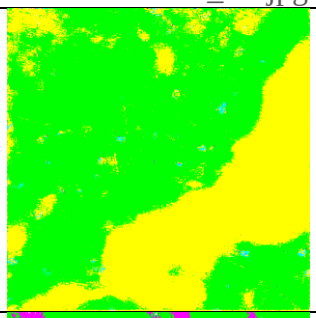
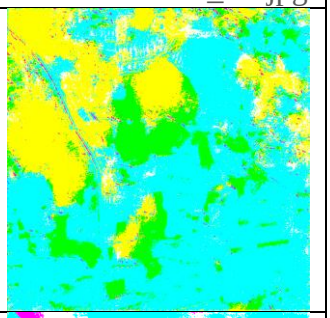
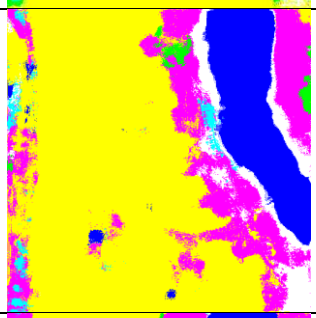
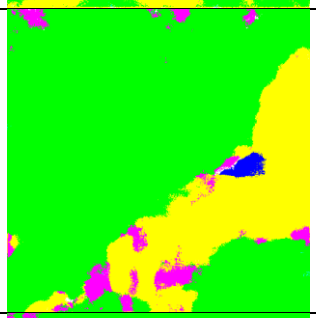
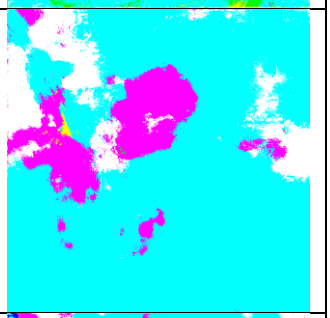
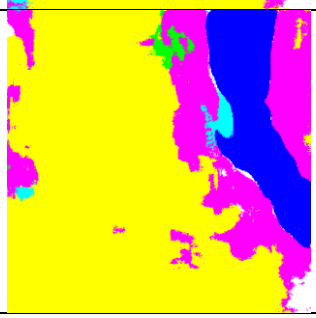
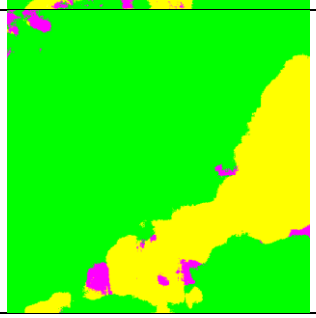
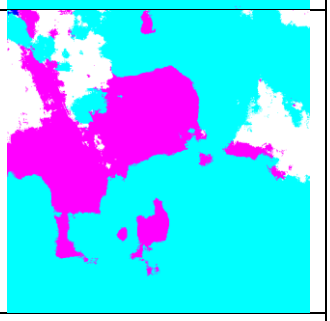
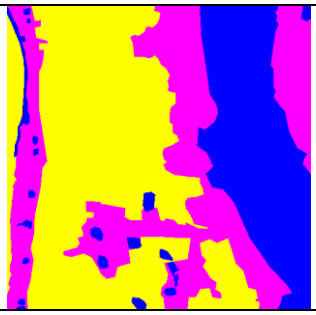
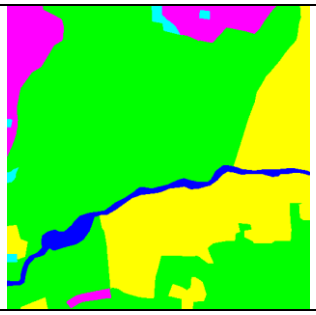
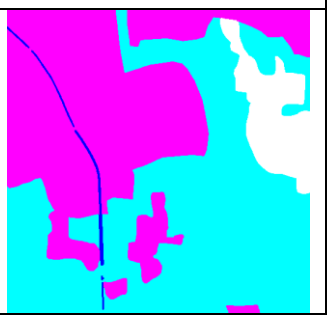
3. (15%) Implement an improved model which performs better than your baseline model. Print the network architecture of this model.

VGGUnet:使用pre-train的VGG16前4個block來初始化Unet。

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 512, 512, 3)	0	
batch_normalization_1 (BatchNor	(None, 512, 512, 3)	12	input_1[0][0]
block1_conv1 (Conv2D)	(None, 512, 512, 64)	1792	batch_normalization_1[0][0]
block1_conv2 (Conv2D)	(None, 512, 512, 64)	36928	block1_conv1[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 256, 256, 64)	0	block1_conv2[0][0]
block2_conv1 (Conv2D)	(None, 256, 256, 128	73856	max_pooling2d_1[0][0]
block2_conv2 (Conv2D)	(None, 256, 256, 128	147584	block2_conv1[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 128, 128, 128	0	block2_conv2[0][0]
block3_conv1 (Conv2D)	(None, 128, 128, 256	295168	max_pooling2d_2[0][0]
block3_conv2 (Conv2D)	(None, 128, 128, 256	590080	block3_conv1[0][0]
block3_conv3 (Conv2D)	(None, 128, 128, 256	590080	block3_conv2[0][0]
max_pooling2d_3 (MaxPooling2D)	(None, 64, 64, 256)	0	block3_conv3[0][0]
block4_conv1 (Conv2D)	(None, 64, 64, 512)	1180160	max_pooling2d_3[0][0]
block4_conv2 (Conv2D)	(None, 64, 64, 512)	2359808	block4_conv1[0][0]
block4_conv3 (Conv2D)	(None, 64, 64, 512)	2359808	block4_conv2[0][0]
conv2d_transpose_1 (Conv2DTrans	(None, 128, 128, 128	262272	block4_conv3[0][0]
concatenate_1 (Concatenate)	(None, 128, 128, 384	0	conv2d_transpose_1[0][0] block3_conv3[0][0]
conv2d_1 (Conv2D)	(None, 128, 128, 256	884992	concatenate_1[0][0]
conv2d_2 (Conv2D)	(None, 128, 128, 256	590080	conv2d_1[0][0]
conv2d_3 (Conv2D)	(None, 128, 128, 256	590080	conv2d_2[0][0]
conv2d_transpose_2 (Conv2DTrans	(None, 256, 256, 64)	65600	conv2d_3[0][0]
concatenate_2 (Concatenate)	(None, 256, 256, 192	0	conv2d_transpose_2[0][0] block2_conv2[0][0]
conv2d_4 (Conv2D)	(None, 256, 256, 128	221312	concatenate_2[0][0]
conv2d_5 (Conv2D)	(None, 256, 256, 128	147584	conv2d_4[0][0]
conv2d_6 (Conv2D)	(None, 256, 256, 128	147584	conv2d_5[0][0]
conv2d_transpose_3 (Conv2DTrans	(None, 512, 512, 32)	16416	conv2d_6[0][0]
concatenate_3 (Concatenate)	(None, 512, 512, 96)	0	conv2d_transpose_3[0][0] block1_conv2[0][0]
conv2d_7 (Conv2D)	(None, 512, 512, 64)	55360	concatenate_3[0][0]
conv2d_8 (Conv2D)	(None, 512, 512, 64)	36928	conv2d_7[0][0]
conv2d_9 (Conv2D)	(None, 512, 512, 7)	455	conv2d_8[0][0]
activation_1 (Activation)	(None, 512, 512, 7)	0	conv2d_9[0][0]

Total params: 10,653,939
Trainable params: 10,653,933
Non-trainable params: 6

4. (10%) Show the predicted segmentation mask of “validation/0008_sat.jpg”, “validation/0097_sat.jpg”, “validation/0107_sat.jpg” during the early, middle, and the final stage during the training process of this improved model.

Epoch	validation/0008_sat.jpg	validation/0097_sat.jpg	validation/0107_sat.jpg
1 early			
16 middle			
34 final			
Ground truth			

5. (15%) Report mIoU score of both models on the validation set. Discuss the reason why the improved model performs better than the baseline one. You may conduct some experiments and show some evidences to support your reasoning.

下圖顯示2種模型的mean IoU; 上面為VGG16-FCN32s 的結果，下則為VGGUnet的結果。

```
gary830317@Gary:~/hw3$ python3 mean_iou_evaluate.py -g ./data/validation/ -p ./data/predict/FCN_Vgg16_32s/
class #0 : 0.74580
class #1 : 0.86807
class #2 : 0.25362
class #3 : 0.76030
class #4 : 0.67762
class #5 : 0.62800

mean_iou: 0.655568
gary830317@Gary:~/hw3$ python3 mean_iou_evaluate.py -g ./data/validation/ -p ./data/predict/VGGUnet/
class #0 : 0.75994
class #1 : 0.86992
class #2 : 0.31305
class #3 : 0.78297
class #4 : 0.75565
class #5 : 0.61201

mean_iou: 0.682257
```

Unet與FCN32相似的地方在於都是使用CNN將影像壓縮至較小的維度再deconvolute 成原來圖片的大小。然而因為Unet的架構中有特別設計在model後半部的layer中會不斷加入由前面的layer所產生的一些較high level的資訊，使得Unet在output端有更完整的資訊可以使用。在我的實驗中，Unet所使用到的參數大約只有FCN32的一半，但是依然可以得到比較好的結果。