Name: 楊耀程 Dep.:電機四 Student ID:B03901161

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

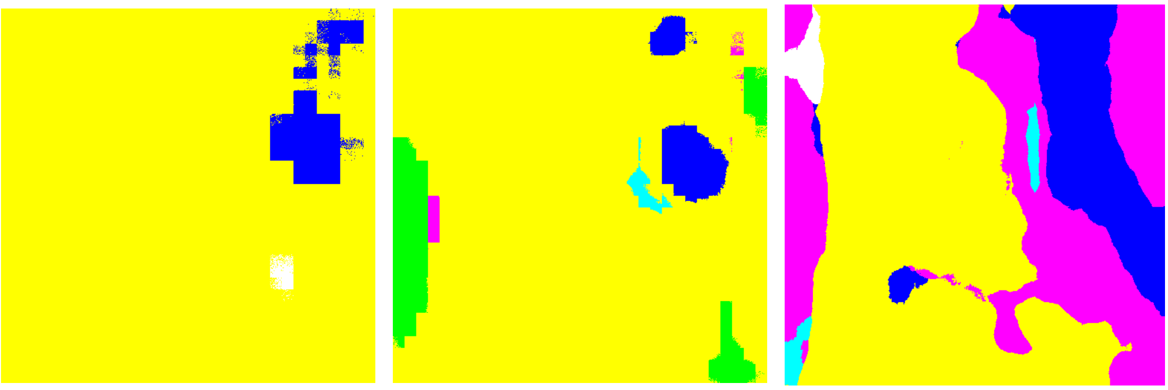
Network structure

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 512, 512, 3)       0
_________________________________________________________________
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
_________________________________________________________________
fc1 (Conv2D)                 (None, 16, 16, 4096)      102764544
_________________________________________________________________
fc2 (Conv2D)                 (None, 16, 16, 4096)      16781312
_________________________________________________________________
fc3 (Conv2D)                 (None, 16, 16, 7)         28679
_________________________________________________________________
up32 (Conv2DTranspose)       (None, 512, 512, 7)       200711
=================================================================
Total params: 134,489,934
Trainable params: 119,775,246
Non-trainable params: 14,714,688
_________________________________________________________________
```

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)

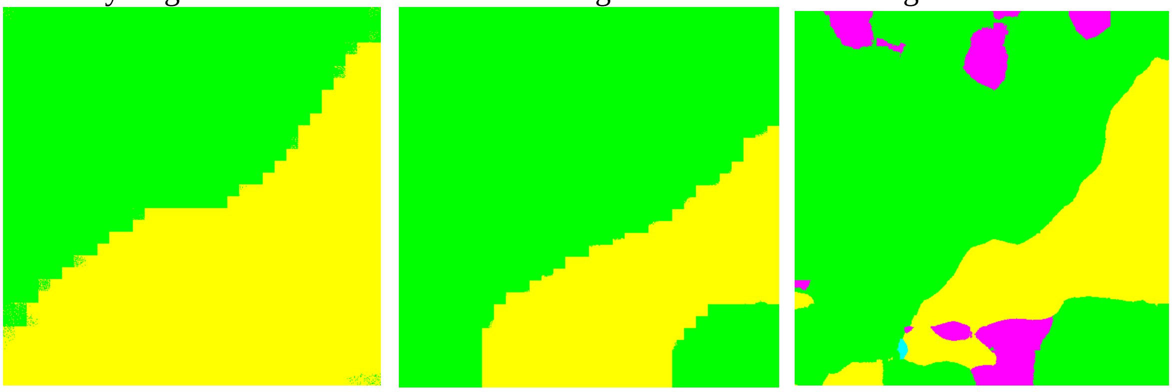
0008_sat

Early stage: Middle stage: Final stage:



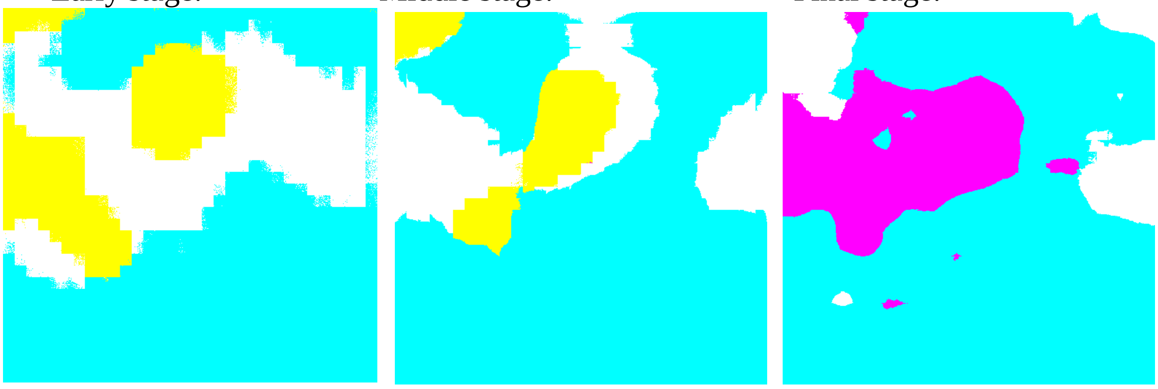
0097_sat

Early stage: Middle stage: Final stage:



0107_sat

Early stage: Middle stage: Final stage:



結論: 可以看到隨著 training 的進行，segmentation 的結果有逐漸 improve。

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

Improved model 為 FCN-16s

derek@derek-System-Product-Name: ~/Documents/dlcv_hw3

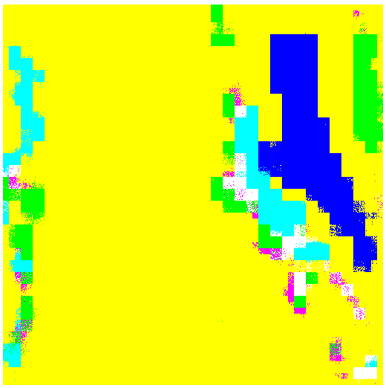
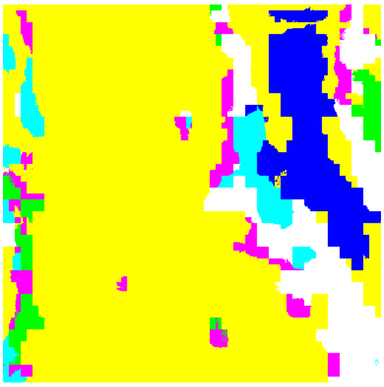
Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 512, 512, 3)	0	
block1_conv1 (Conv2D)	(None, 512, 512, 64)	1792	input_1[0][0]
block1_conv2 (Conv2D)	(None, 512, 512, 64)	36928	block1_conv1[0][0]
block1_pool (MaxPooling2D)	(None, 256, 256, 64)	0	block1_conv2[0][0]
block2_conv1 (Conv2D)	(None, 256, 256, 128	73856	block1_pool[0][0]
block2_conv2 (Conv2D)	(None, 256, 256, 128	147584	block2_conv1[0][0]
block2_pool (MaxPooling2D)	(None, 128, 128, 128	0	block2_conv2[0][0]
block3_conv1 (Conv2D)	(None, 128, 128, 256	295168	block2_pool[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]
block3_pool (MaxPooling2D)	(None, 64, 64, 256)	0	block3_conv3[0][0]
block4_conv1 (Conv2D)	(None, 64, 64, 512)	1180160	block3_pool[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]
block4_pool (MaxPooling2D)	(None, 32, 32, 512)	0	block4_conv3[0][0]
block5_conv1 (Conv2D)	(None, 32, 32, 512)	2359808	block4_pool[0][0]
block5_conv2 (Conv2D)	(None, 32, 32, 512)	2359808	block5_conv1[0][0]
block5_conv3 (Conv2D)	(None, 32, 32, 512)	2359808	block5_conv2[0][0]
block5_pool (MaxPooling2D)	(None, 16, 16, 512)	0	block5_conv3[0][0]
fc1 (Conv2D)	(None, 16, 16, 4096)	102764544	block5_pool[0][0]
dropout_1 (Dropout)	(None, 16, 16, 4096)	0	fc1[0][0]
fc2 (Conv2D)	(None, 16, 16, 4096)	16781312	dropout_1[0][0]
dropout_2 (Dropout)	(None, 16, 16, 4096)	0	fc2[0][0]
fc3 (Conv2D)	(None, 16, 16, 7)	28679	dropout_2[0][0]
up2 (Conv2DTranspose)	(None, 32, 32, 7)	784	fc3[0][0]
conv_pool4 (Conv2D)	(None, 32, 32, 7)	3591	block4_pool[0][0]
add_1 (Add)	(None, 32, 32, 7)	0	up2[0][0] conv_pool4[0][0]
up16 (Conv2DTranspose)	(None, 512, 512, 7)	50176	add_1[0][0]

Total params: 134,343,774
Trainable params: 119,629,086
Non-trainable params: 14,714,688

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.

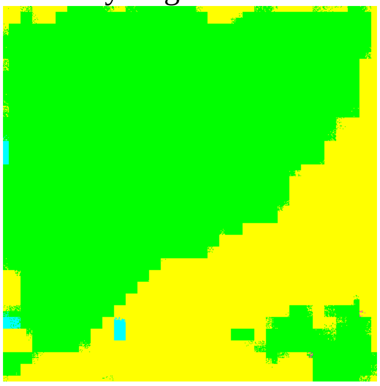
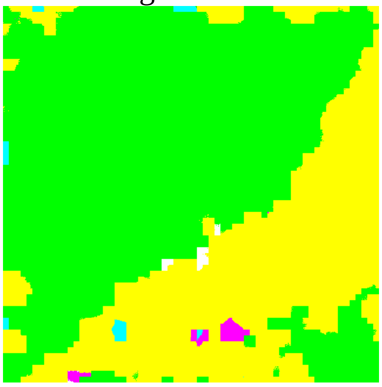
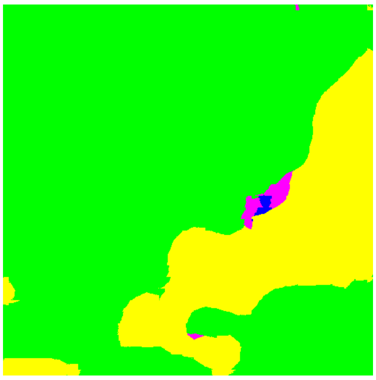
0008_sat

Early stage: Middle stage: Final stage:

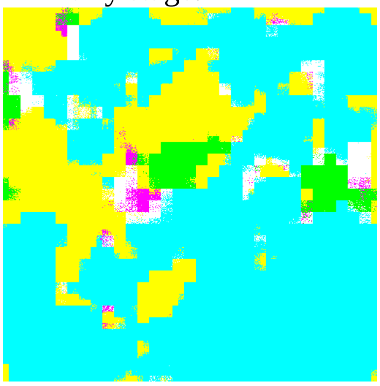
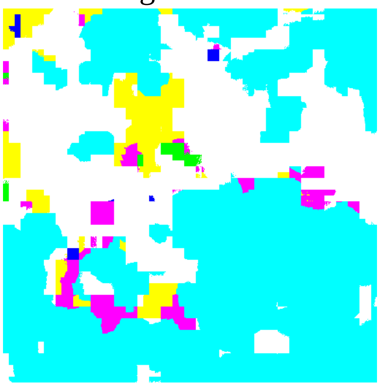

0097_sat

Early stage: Middle stage: Final stage:

0107_sat

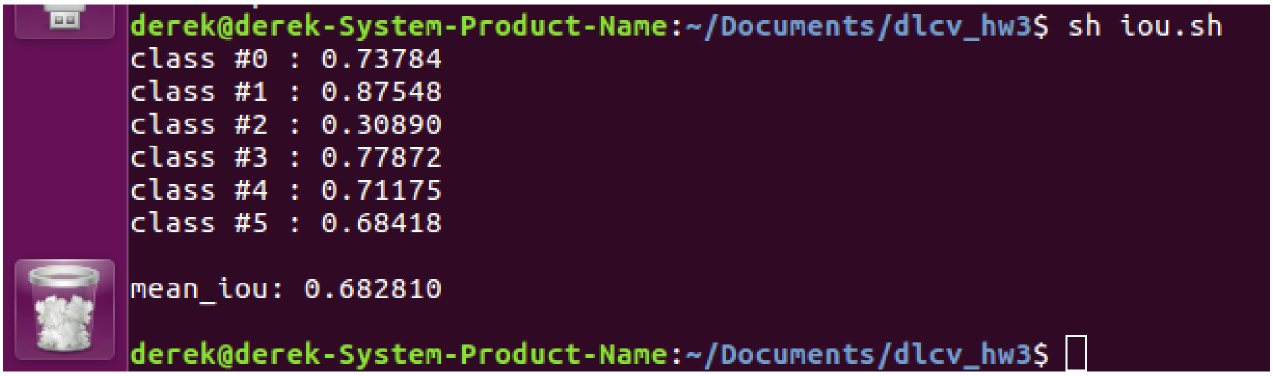
Early stage: Middle stage: Final stage:


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 discussion.

(report mIoU: 所得到的 mean IOU 在 validation set 皆有超過 baseline，而且 improved model 的 mIoU 較 baseline model 更高)

Baseline model:

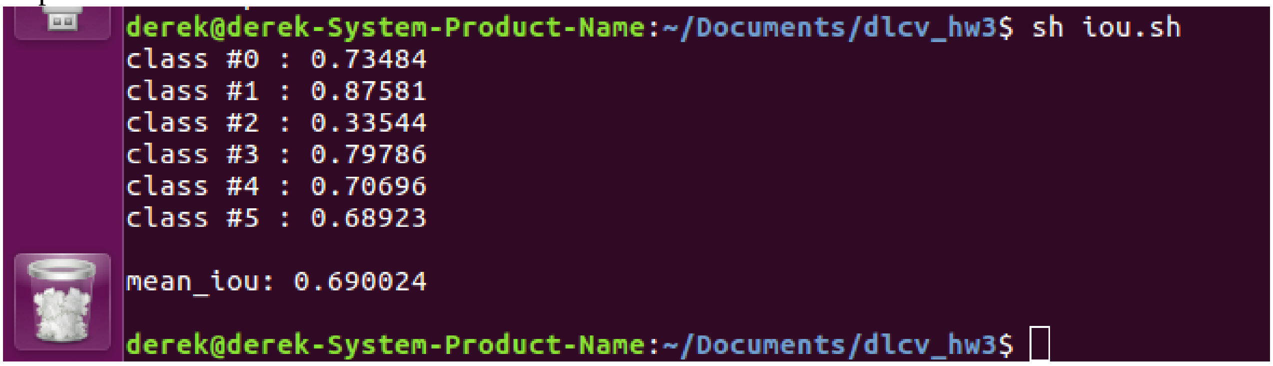


```
derek@derek-System-Product-Name:~/Documents/dlcv_hw3$ sh iou.sh
class #0 : 0.73784
class #1 : 0.87548
class #2 : 0.30890
class #3 : 0.77872
class #4 : 0.71175
class #5 : 0.68418

mean_iou: 0.682810

derek@derek-System-Product-Name:~/Documents/dlcv_hw3$
```

Improved model:



```
derek@derek-System-Product-Name:~/Documents/dlcv_hw3$ sh iou.sh
class #0 : 0.73484
class #1 : 0.87581
class #2 : 0.33544
class #3 : 0.79786
class #4 : 0.70696
class #5 : 0.68923

mean_iou: 0.690024

derek@derek-System-Product-Name:~/Documents/dlcv_hw3$
```

討論: 由 model 架構圖可知，相較於 fcn32s，fcn16s 多了從 VGG block 4 的 pooling 的資訊，再做 transpose convolution，並和原本的資訊做 Sum (Add)。等於是同時用到 VGG block 5 pooling 以及 block 4 pooling，同時參考這兩個不同地方得到的 feature，因此 train 得當的話，fcn16s 確實很可能會比 fcn32s 有更好的 segmentation 結果，有更高的 mean IOU(如本題數據所示)。

6. (5%) [bonus] Calculate the result of d/dw G(w):

objective function:

$$G(\boldsymbol{w}) = -\sum_n \left[t^{(n)} \log \mathrm{x}(\boldsymbol{z}^{(n)};\boldsymbol{w}) + (1-t^n)\log\left(1-\mathrm{x}(\boldsymbol{z}^{(n)};\boldsymbol{w})\right)\right] \quad \geq 0$$

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} G(\boldsymbol{w})$$ choose the weights that minimise the network's surprise about the training data

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{w}}G(\boldsymbol{w}) = \sum_n \frac{\mathrm{d}G(\boldsymbol{w})}{\mathrm{d}x^{(n)}}\frac{\mathrm{d}x^{(n)}}{\mathrm{d}\boldsymbol{w}} = -\sum_n (t^{(n)} - x^{(n)})\boldsymbol{z}^{(n)}$$ = prediction error x feature

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{w}}G(\boldsymbol{w})$$ iteratively step down the objective (gradient points up hill) 39

$$G(\underline{w}) = -\sum_n t^{(n)} \log x(\underline{z}^{(n)}; \underline{w}) + (1-t^n) \log(1 - x(\underline{z}^{(n)}; \underline{w}))] \geq 0$$

$$\frac{dG_n(w)}{dw_j} = \sum_{n=1}^{N} \frac{dG_n(w)}{dx^n} \frac{dx^n}{dw_j} \quad \Bigg(\begin{array}{l} \text{where } x^n = x(w^T z^{(n)}) \\ G_n(w) = -[t^{(n)} \cdot \log(x(w^T z^n)) + \\ (1-t^{(n)}) \cdot \log(1 - x(w^T z^{(n)}))] \end{array}$$

利用 $\frac{dx(\alpha)}{d\alpha} = x(\alpha) \cdot [1 - x(\alpha)]$

$$\frac{dG_n(w)}{dx^{(n)}} = \frac{t^{(n)}}{x^{(n)}} - \frac{1-t^{(n)}}{1-x^{(n)}} \cdots ①$$

$$\frac{dx^{(n)}}{dw_j} = \frac{dx^{(n)}}{dw^T z^{(n)}} \frac{dw^T z^{(n)}}{dw_j} = x^{(n)}(1-x^{(n)}) z^{(n,j)} \cdots ②$$

利用①② $\frac{dG_n(w)}{dw_j} = \left(\frac{t^{(n)}}{x^{(n)}} - \frac{1-t^{(n)}}{1-x^{(n)}}\right) x^{(n)}(1-x^{(n)}) z^{(n,j)}$

$$= [t^{(n)} - t^{(n)}x^{(n)} - x^{(n)} + t^{(n)}x^{(n)}] z^{(n,j)}$$

$$= [t^{(n)} - x^{(n)}] z^{(n,j)}$$

$$\therefore \frac{dG(w)}{dw_j} = \sum_{n=1}^{N} (t^{(n)} - x^{(n)}) z^{(n,j)}$$

$$\therefore \frac{dG(w)}{dw} = \sum_{n=1}^{N} [t^{(n)} - x^{(n)}] z^{(n)} \quad \#$$