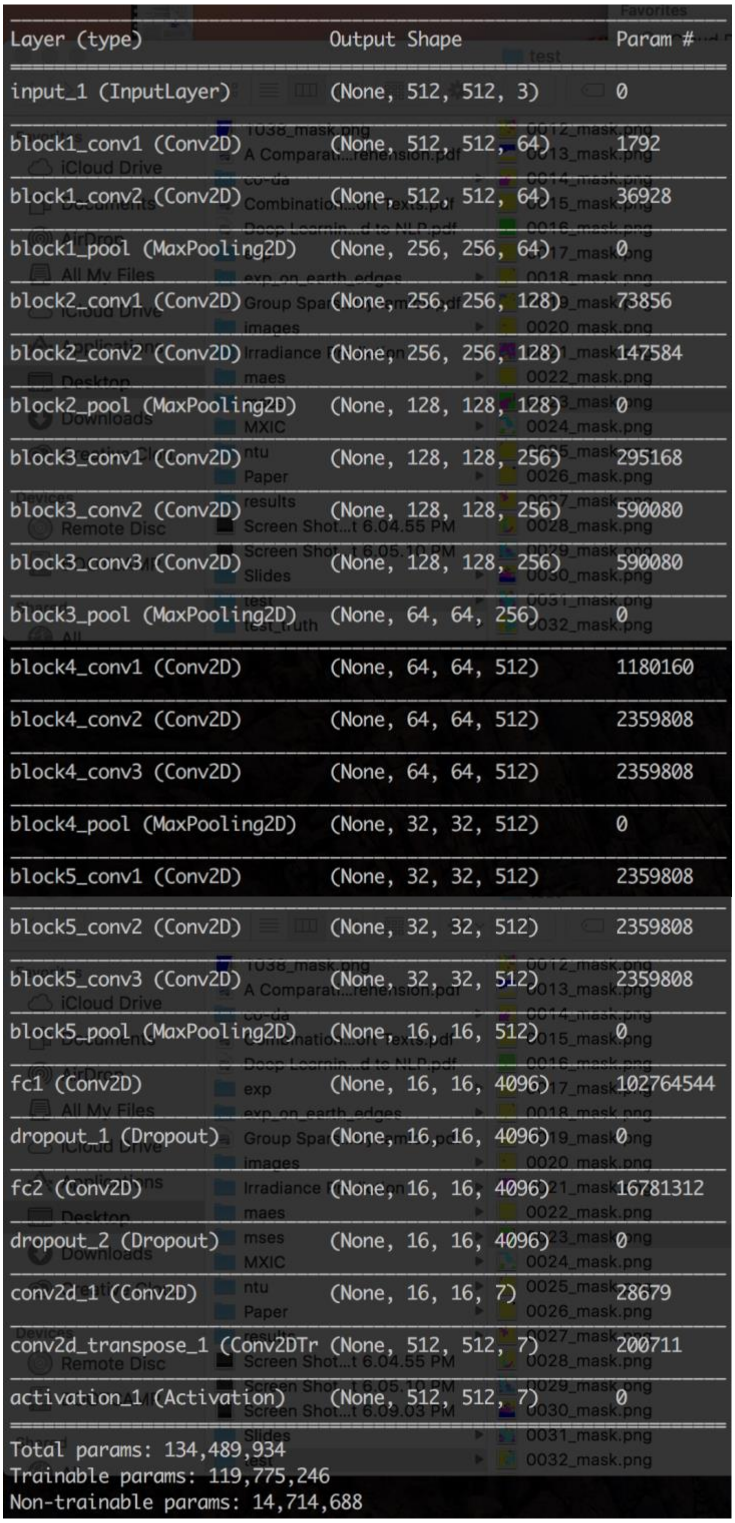
Name: 李冠穎 Dep.:資工碩二 Student ID:R03922165

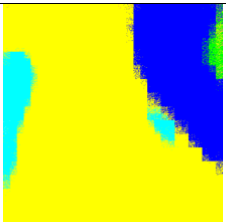
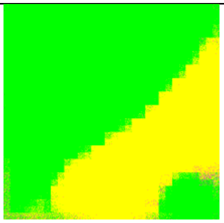
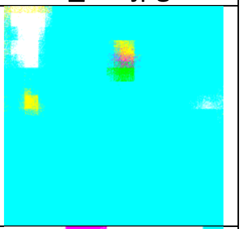
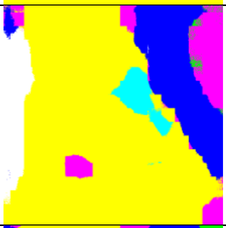
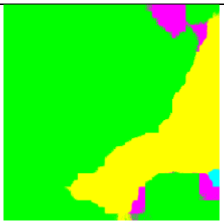
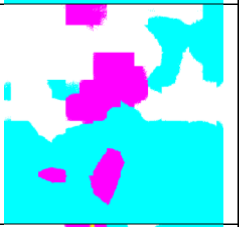
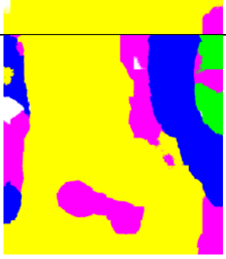
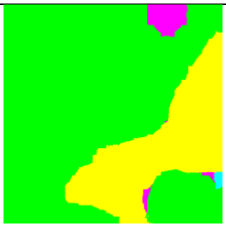
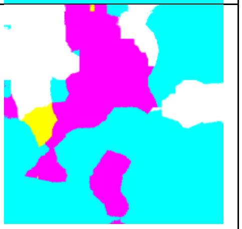
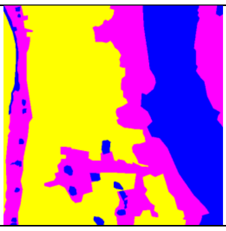
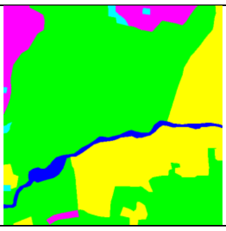
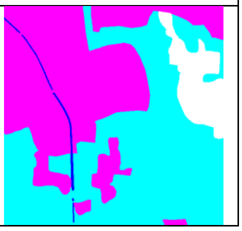
1. (5%) Print the network architecture of your VGG16-FCN32s model.

   將 Pre-trained 好的 VGG model FC layer 換成 Conv layer， 做成 VGG16FCN32s, 並將 VGG 的部分設為 untrainable. Data 部分沒有做任何處理, 而事實上實驗後發現不管是 normalize 到 0-1 之間, 或是減去 127 都沒有比較好, 而 improved model 也是 follow 一樣的 setting.



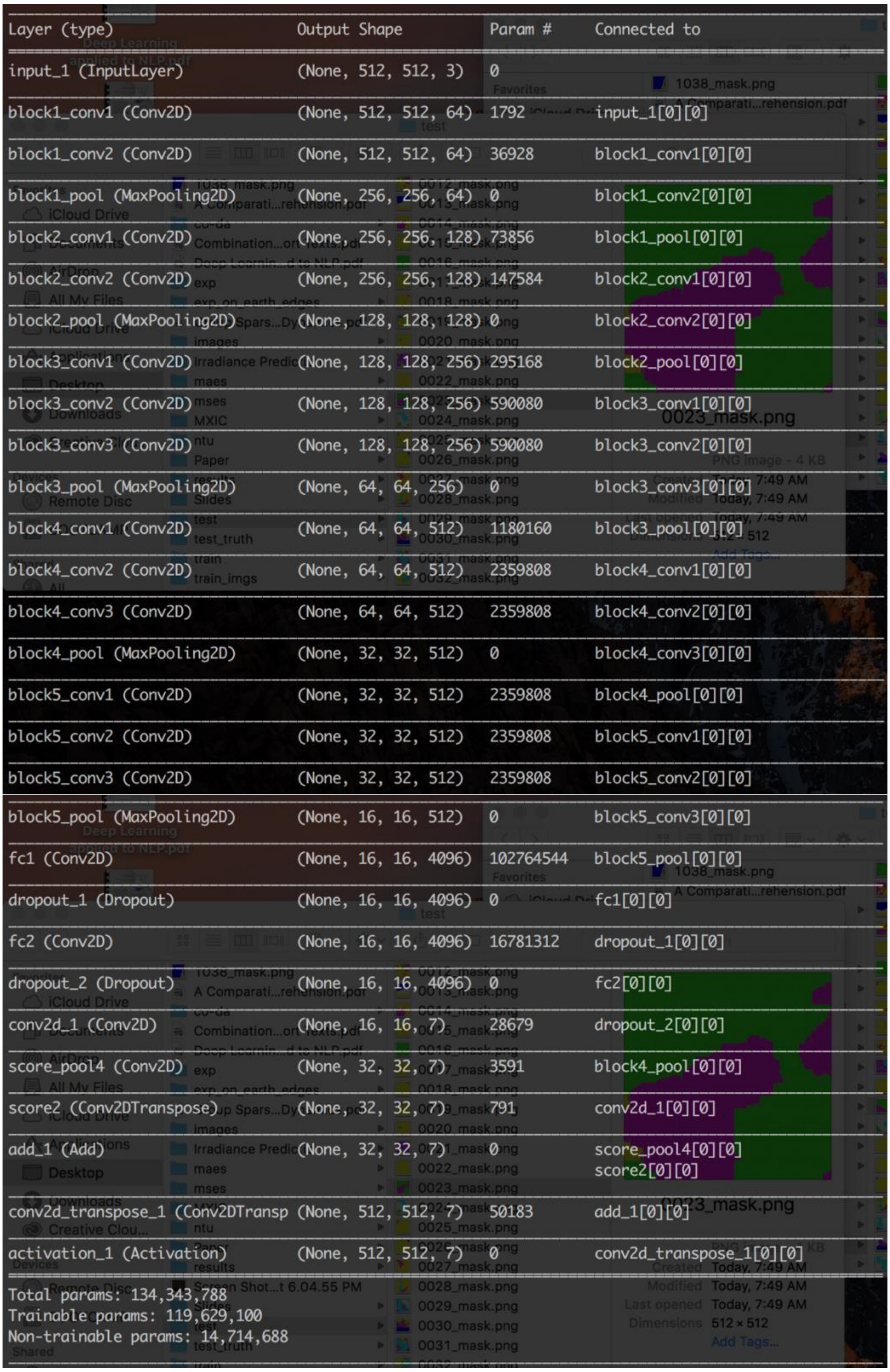
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 512, 512, 3)       0
_________________________________________________________________
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
_________________________________________________________________
dropout_1 (Dropout)          (None, 16, 16, 4096)      0
_________________________________________________________________
fc2 (Conv2D)                 (None, 16, 16, 4096)      16781312
_________________________________________________________________
dropout_2 (Dropout)          (None, 16, 16, 4096)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 16, 16, 7)         28679
_________________________________________________________________
conv2d_transpose_1 (Conv2DTr (None, 512, 512, 7)       200711
_________________________________________________________________
activation_1 (Activation)    (None, 512, 512, 7)       0
=================================================================
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.jpg	0097_sat.jpg	0107_sat.jpg
Early			
Middle (5 epochs)			
Final (10 epochs)			
Ground Truth			

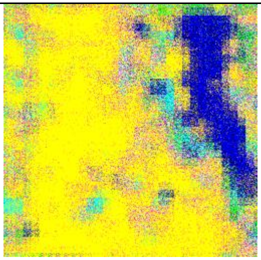
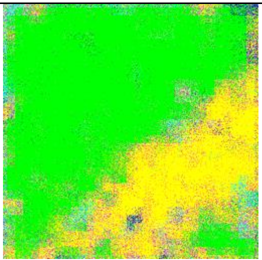
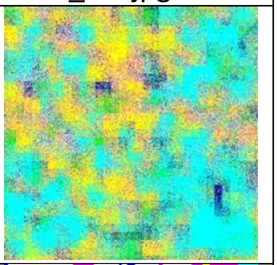
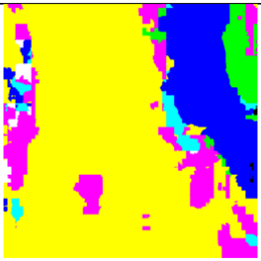
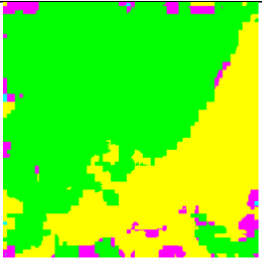
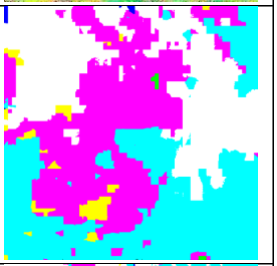
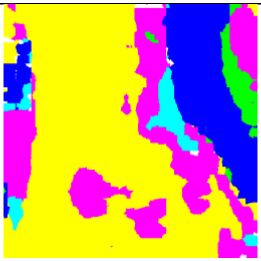
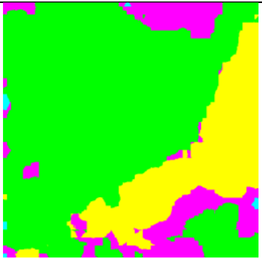
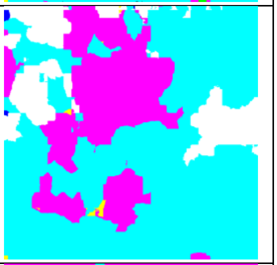
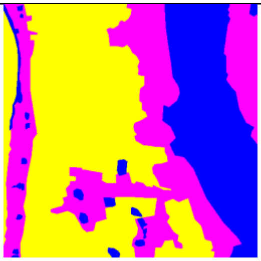
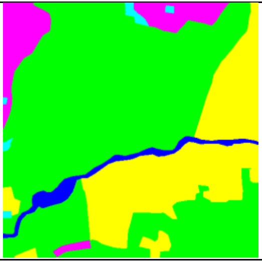
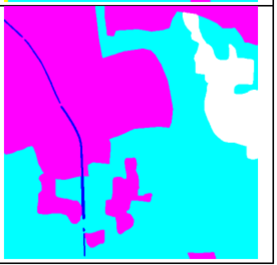
3. (15%) Implement an improved model which performs better than your baseline model. Print the network architecture of this model.

將 network 的 output(16*16)先 deconvolution 到 32*32 後,加入倒數第二個 conv block 的 output(32*32), 完成 VGG16FCN16 的實作, 以期將更 fine-grained 的資訊保留下來,可以將細節處還原得更好.



```
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]
__________________________________________________________________________________________
conv2d_1 (Conv2D)                (None, 16, 16, 7)      28679       dropout_2[0][0]
__________________________________________________________________________________________
score_pool4 (Conv2D)             (None, 32, 32, 7)      3591        block4_pool[0][0]
__________________________________________________________________________________________
score2 (Conv2DTranspose)         (None, 32, 32, 7)      791         conv2d_1[0][0]
__________________________________________________________________________________________
add_1 (Add)                      (None, 32, 32, 7)      0           score_pool4[0][0]
                                                                    score2[0][0]
__________________________________________________________________________________________
conv2d_transpose_1 (Conv2DTransp (None, 512, 512, 7)    50183       add_1[0][0]
__________________________________________________________________________________________
activation_1 (Activation)        (None, 512, 512, 7)    0           conv2d_transpose_1[0][0]
==========================================================================================
Total params: 134,343,788
Trainable params: 119,629,100
Non-trainable params: 14,714,688
```

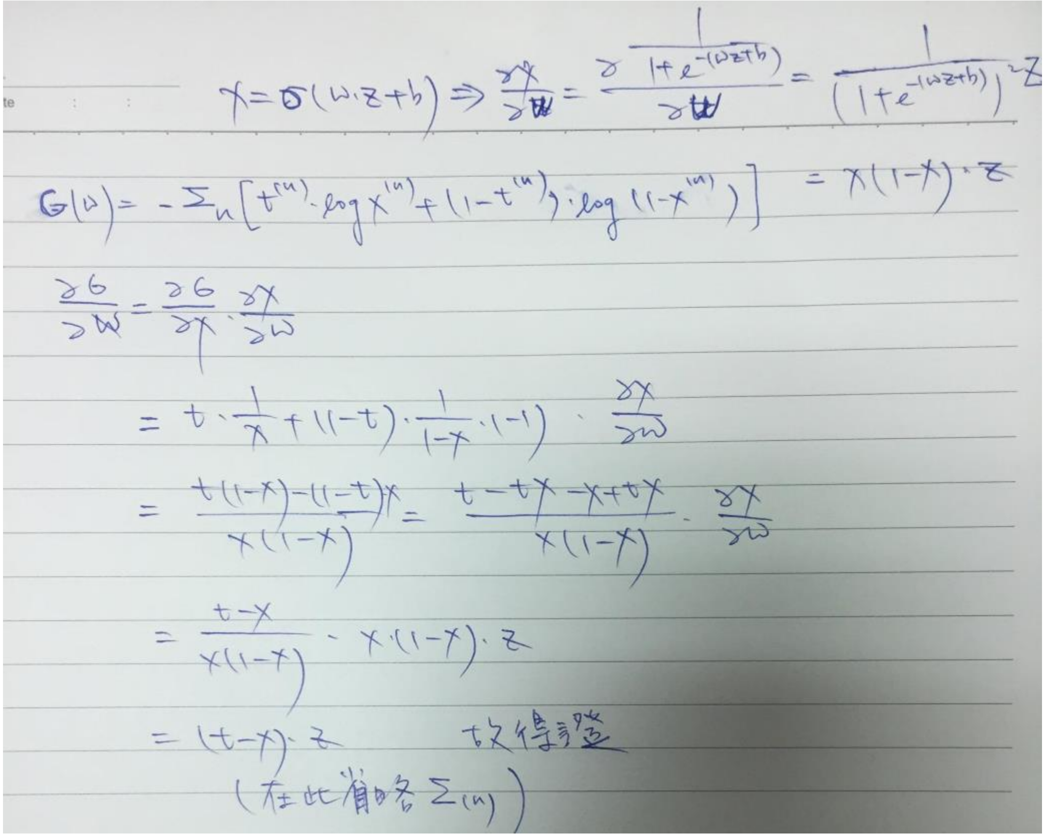
4. (10%) Show the predicted segmentation mask of validation/0008_sat.jpg, validation/0097_sat.jpg, validation/0107_sat.jpg during the early, middle, and the final stage during the training process of this improved model.

	0008_sat.jpg	0097_sat.jpg	0107_sat.jpg
Early			
Middle (6 epochs)			
Final (12 epochs)			
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 discussion.

   Baseline model – VGG16FCN32s: 0.6436
   Improved model – VGG16FCN16s: 0.6485
   可以看到如上的 mask,16s 的結果跟 32s 比起來比較細緻,但有許多不連續的地方,可能肇因於 training 時間不夠久,或是 16s 的還不足以保留太 fine-grained 的資訊, 但可能因為 mean IOU 兩者幾乎一樣, 所以 16s 並沒有看起來比 32s 好太多.

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



$$x=\sigma(w\cdot z+b)\Rightarrow\frac{\partial x}{\partial w}=\frac{\partial\,\frac{1}{1+e^{-(wz+b)}}}{\partial w}=\frac{1}{(1+e^{-(wz+b)})^{2}}z$$

$$=x(1-x)\cdot z$$

$$G(w)=-\Sigma_n\left[t^{(n)}\cdot\log x^{(n)}+(1-t^{(n)})\cdot\log(1-x^{(n)})\right]$$

$$\frac{\partial G}{\partial w}=\frac{\partial G}{\partial x}\cdot\frac{\partial x}{\partial w}$$

$$=t\cdot\frac{1}{x}+(1-t)\cdot\frac{1}{1-x}\cdot(-1))\cdot\frac{\partial x}{\partial w}$$

$$=\frac{t(1-x)-(1-t)x}{x(1-x)}=\frac{t-tx-x+tx}{x(1-x)}\cdot\frac{\partial x}{\partial w}$$

$$=\frac{t-x}{x(1-x)}\cdot x\cdot(1-x)\cdot z$$

$=(t-x)\cdot z$ 故得證

（在此省略 $\Sigma_{(n)}$）