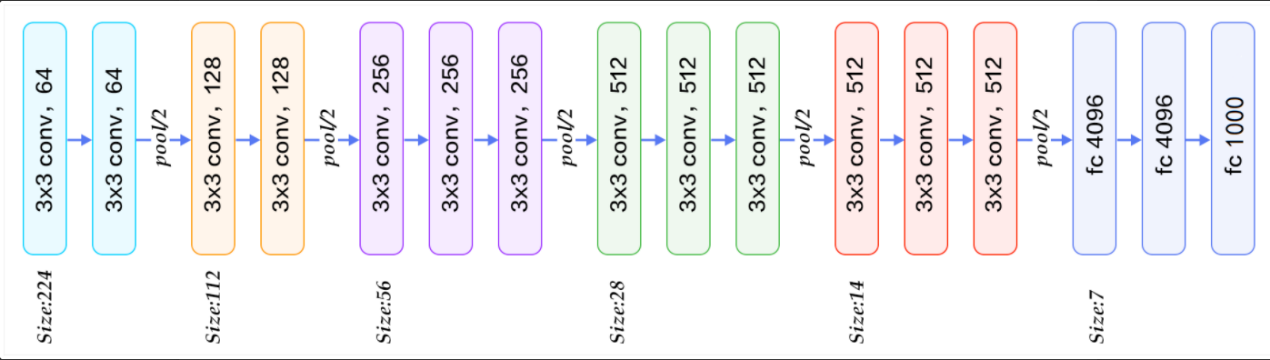
FCN

Implementation

The idea of FCN is based on CNN. It replaced the fully connected layer of CNN by fully convolutional layer and it can classify images at the pixel level. Therefore, it solved the semantic segmentation of image segmentation. Finally, the loss of SoftMax classification is calculated pixel by pixel, which is equivalent to one training label for each pixel. Overall, FCN replaces the fully connected layer at the end of CNN with the convolutional layer, and the output an image which is already labeled.

In our FCN, we first use the VGG16 convolutional part as FCN convolutional part. In original VGG16, the image input size is 224\*224\*3 and the max channels are 512. For each convolutional layer, only the number of channels is changed, and the size is become one half after each maxpool layer, which means after 5 convolutional layers, the size becomes 1/8 of its original size. However, this structure might not work well on this dataset, as the magnetic resonance (MR) dataset is 96\*96\*1. It is necessary to adapt the layers and parameters to fit our dataset. We keep each convolutional layer parameter as same as VGG16 except the input and output channel. The input channel for first layer which is input layer is obviously 1 and the output channel becomes 16. The output channel of rest four convolutional layers is 32, 64, 128, 256 respectively. That’s all of the convolutional part. An advantage of this structure is when doing convolutional part to extract the features, the size never changed and it means information that is not lost at the edge of the image. And the maxpool layer reduce the size of feature map and improve the anti-interference ability of network.

Then the upsample part aims to make the convolved image return to the input size. The easiest way to implement this is to use upsampling at a factor of 32. However, as we can see on figure2 the predicted mask, it is generally similar with origin mask, the details are very poor. In addition, the accuracy of validation in this case is only around 70%.

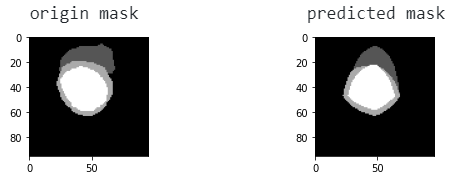
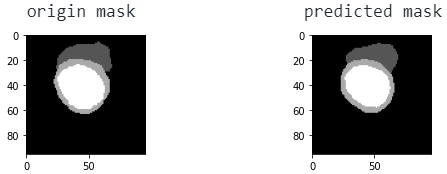


Figure1 Figure2



Then we use upsampling at a factor of 2 first, apply this to the output of convolutional layer, and add with the output of fourth convolutional part, we get “x1”. Using “x1” and add with the output of third convolutional part, we get “x2”. Finally, using “x2” with upsampling at a factor of 8, we get the predicted Figure3 output. Now, the predicted mask as shown on figure 3, most of the details are fine, and we can reach the maximum validation accuracy about 88%.

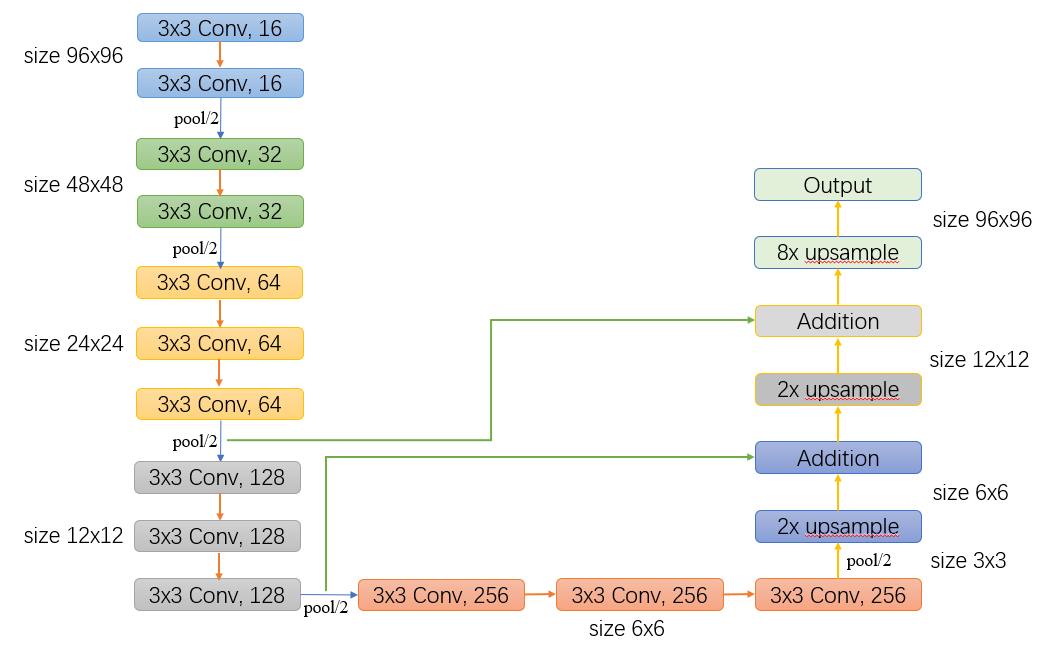
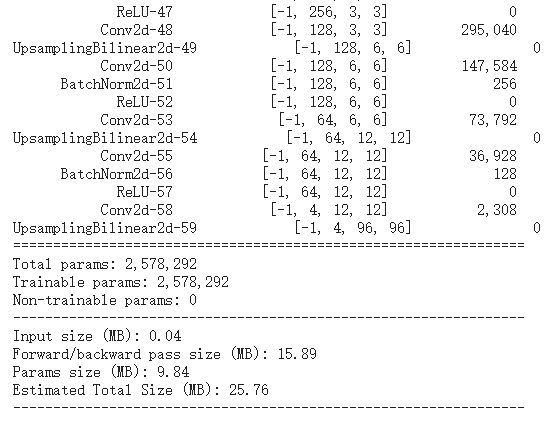
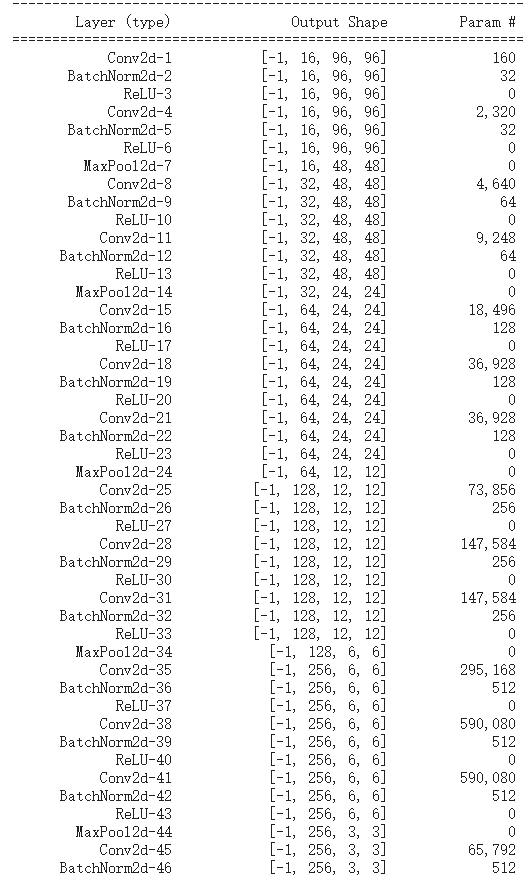


Figure4

In summary, there are 5 convolutional part and 3 upsample part in our network. The output shape of each convolutional part after maxpool are 48\*48\*16, 24\*24\*32, 12\*12\*64, 6\*6\*128, 3\*3\*256. Total parameters in this network are 2,578,292 and total size is about 25.76.



Experiment

We tried how the different channels effect the accuracy