影像處理期末報告　　　　Q56081082 彭羿凡

I. Questions

We have sixty gray level X-ray vertebral digital images with ground truth, and all of the ground truths are drawn by an expert. For every image, the image shape is 500 x 1200 pixels. The target is to segment and do multi-label annotation of all the vertebrae.

II. Methods

Multi-class segmentation of vertebrae is a non-trivial task due to the high correlation in the appearance of adjacent vertebrae. The localization of vertebrae is important because the deep learning network can learn from the image more easily. After the localization is done, I apply histogram equalization to all of the images. Before conducting the multi-class segmentation, I perform a binary segmentation task to pre-train the network. Finally, I use a network for multi-class segmentation to segment all the vertebrae at once.

- Localization

  After reading all the X-ray image, I found that the ROI (region of interest) is around the central of the image (i.e. x = 250). Thus, I crop 50 columns at the left and right of the image, and get a new image of shape (400, 1200) as shown in Figure1.

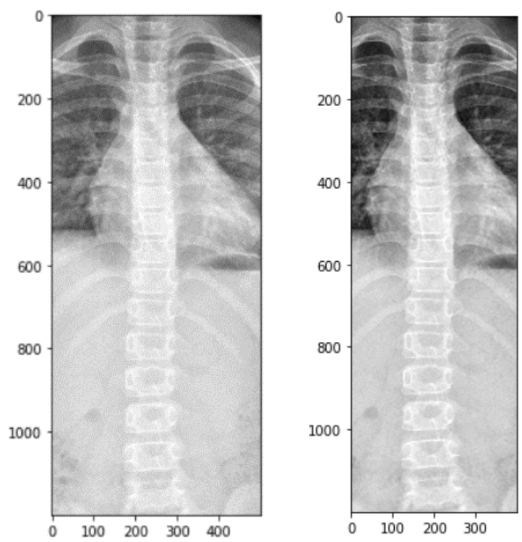


Figure1.
Left : original image
Right : after cropping

- Histogram equalization

  For histogram equalization, I use the openCV library to finish the process. Due to the unbalance of the image contrast, I split the image to four sub-images in the shape of (400, 300). I apply histogram equalization on all sub-images, and then merger the sub-image back to original image size. Finally, I average each pixel value with original image. That is, I get a new image (Figure2) for all pixel (x, y) = 0.5 * [histogram equalization image (x, y)] + 0.5 * [original image (x, y)]. In addition, I resize the image into shape of 224 x 672 pixels.

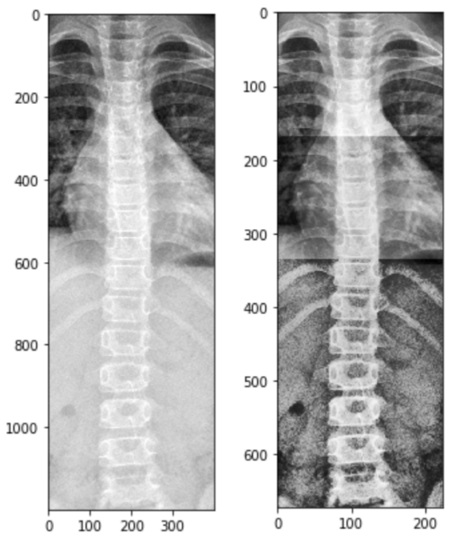


Figure2.
Left : original image
Right : after the process

- Pre-train

  In order to attempt a related more complex network, it is a common practice to pre-train a network and use it as an initialization for the network. The network for pre-train is the same as multi-class segmentation (except the output layer). A network trained for binary segmentation of spine (spine vs. background) is employed as an initialization for the multi-class segmentation. This overcomes the shortcoming of the limited data at my disposal to train a network for the relatively complex task of multi-class segmentation. The pre-train network was run for 1000 epochs and save the best weighting.

- Multi-class segmentation

  The network is implemented in the tensorflow framework. I rely on the Unet as shown in Figure3, and the network predicts a multi-class label. This label indicates whether the pixel belongs to which vertebra or background. The idea of the network is inspired by the semantic segmentation like FCN. The input shape is 224 x 672 pixels, which contain all the vertebrae and all kernel size is 9 x 9 with same padding in all convolution layers. A dice coefficient loss is optimized using an Adam optimizer with learning rate of 1e-4. The network was train by three-folder validation and ran for 2000 epochs for each folder and saved the best weighting during training process. The output layer is activated by softmax which enable to classify a pixel to certain vertebra or background.

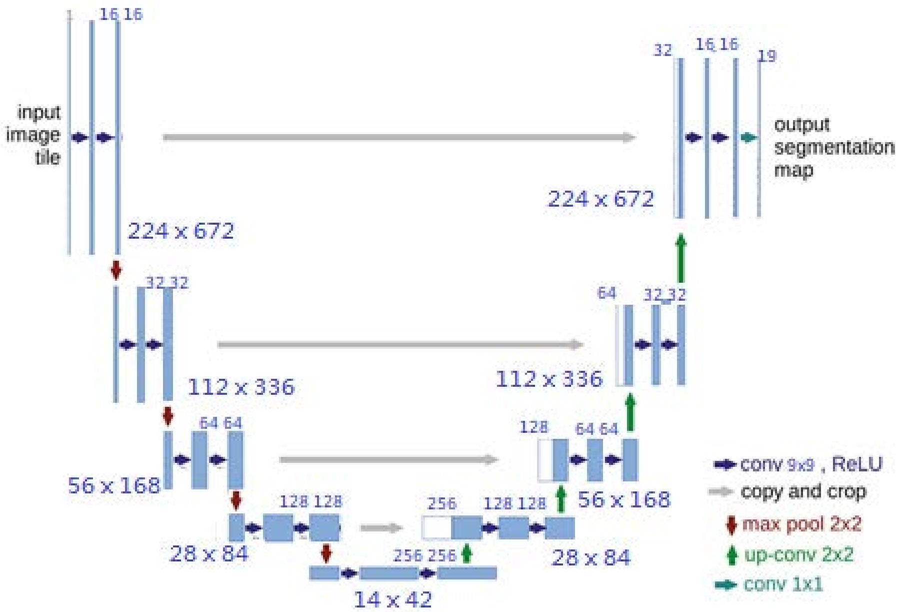
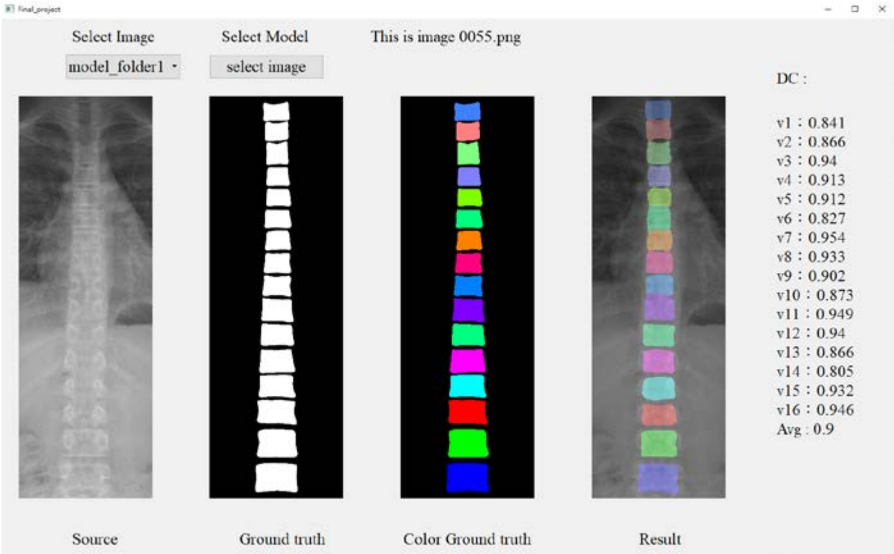


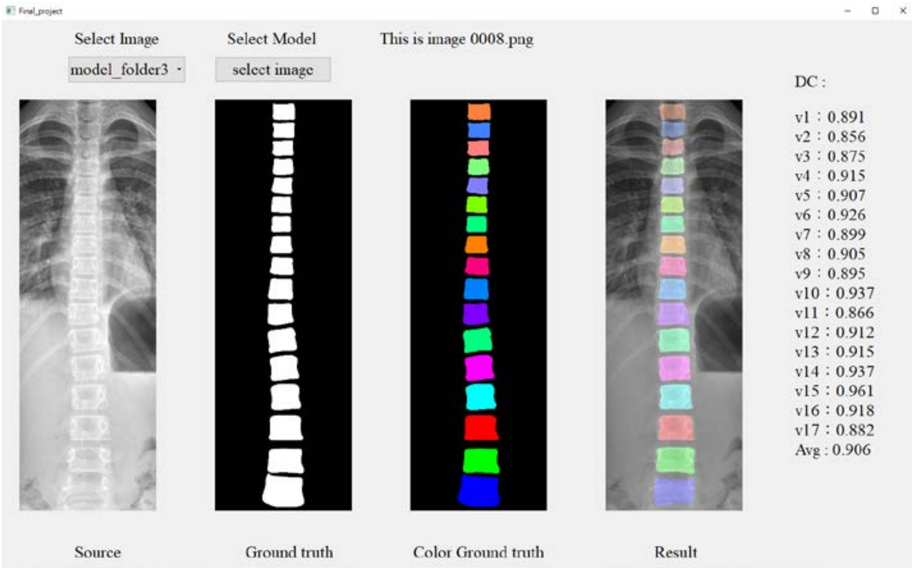
Figure 3.

III. Results

- Output segmentation map

  In order to get the segmentation image, the argmax {y, axis = last channel}, y is the output map, is applied. Thus, I could get an image whose pixel value was [0, 18] for all pixel value is an integer. The value 0 represent background and 1 to 18 represent different vertebrae in the image. The network should learn to identify how many vertebrae are there in the image and segment each vertebra.


Final_project
Select Image
Select Model
This is image 0055.png
model_folder1
select image
DC :
v1 : 0.841
v2 : 0.866
v3 : 0.94
v4 : 0.913
v5 : 0.912
v6 : 0.827
v7 : 0.954
v8 : 0.933
v9 : 0.902
v10 : 0.873
v11 : 0.949
v12 : 0.94
v13 : 0.866
v14 : 0.805
v15 : 0.932
v16 : 0.946
Avg : 0.9
Source
Ground truth
Color Ground truth
Result


Final_project
Select Image
Select Model
This is image 0008.png
model_folder3
select image
DC :
v1 : 0.891
v2 : 0.856
v3 : 0.875
v4 : 0.915
v5 : 0.907
v6 : 0.926
v7 : 0.899
v8 : 0.905
v9 : 0.895
v10 : 0.937
v11 : 0.866
v12 : 0.912
v13 : 0.915
v14 : 0.937
v15 : 0.961
v16 : 0.918
v17 : 0.882
Avg : 0.906
Source
Ground truth
Color Ground truth
Result

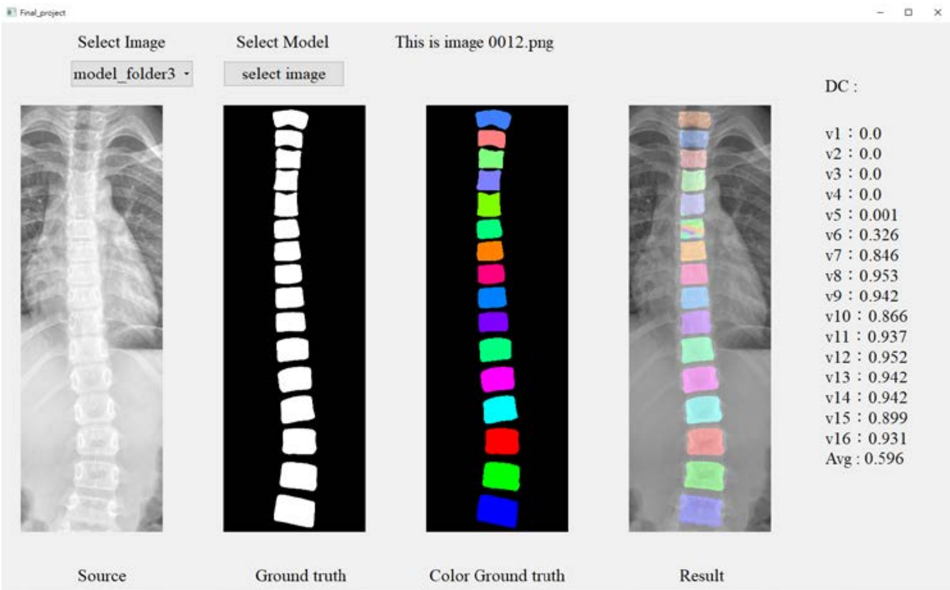

Final_project
Select Image
Select Model
This is image 0012.png
model_folder3
select image
DC :
v1 : 0.0
v2 : 0.0
v3 : 0.0
v4 : 0.0
v5 : 0.001
v6 : 0.326
v7 : 0.846
v8 : 0.953
v9 : 0.942
v10 : 0.866
v11 : 0.937
v12 : 0.952
v13 : 0.942
v14 : 0.942
v15 : 0.899
v16 : 0.931
Avg : 0.596
Source
Ground truth
Color Ground truth
Result

Figure4. (Top) The network have good performance.

(Central) The network have good performance.

(Bottom) v1 to v6 is shift by one result to low average DC.

IV. Discussion

The number of vertebrae in the database is different for four case. Most of the images contain sixteen or seventeen vertebrae. The residual images contain fifteen or eighteen. The non-consistent of the vertebrae number would be a problem when training a network. It would result to the terrible case as shown in Figure4 (Bottom).

V. Conclusion

I learnt a lot from this project, including dealing with the image for getting better contrast, learning how to train a semantic segmentation network. Although the outcome was not quite good, I enjoyed this process and gained more knowledge about digital image processing.