Machine Learning in Medical Imaging Coursework 2

Medical Ultrasound Classification and Segmentation*

Wengxi Li

Medical Physics and Biomedical Engineering, University College London

1 Introduction

Transrectal B-mode ultrasound images are clinically used to guide different urologic procedures, such as ablation therapy and needle biopsies. Real-time segmenting prostate gland from the 2D ultrasound images can help surgeons to localise the relevant anatomical structures and subsequently help targeting regions of interest. However, identifying the boundaries of the prostate gland capsules is a challenging task even for experienced urologists. The objective of this report is to develop and compare different model development strategies with available data set and labels from multiple observers based on deep neural network, which could help develop an automatic 2D image segmentation tool for practical medical usage.

2 Methods

2.1 Segmentation

In the past few years, deep convolutional neural networks (CNN) have outperformed the state of the art in many visual recognition tasks. The typical use of CNN is on classification tasks, where the output to an image is a single class label. However, in many visual tasks, especially in biomedical image processing, the desired output should include localization, i.e., a class label is supposed to be assigned to each pixel. Moreover, thousands of training images are usually beyond reach in biomedical tasks. To produce a precise and fast segmentation result based on very few prostate gland images, a fully CNN called UNet[1] is implemented, which consists of a contracting path to capture context and a symmetric expanding path.

Considering the features of the dataset we have, a simplified and optimized UNet is implemented. There are two max pooling in the down-sampling process, correspondingly two up-convolution in the up-sampling process. In each residual block, there are convolutions, ReLU, batch normalization and residual shortcuts, which could help gradients to propagate further and allow for efficient training of very deep nets. Between down-sampling layers and up-sampling layers, there are skip layers, which are

* Wengxi.li.20@ucl.ac.uk
https://github.com/imerlwx/mphy0041-cw-20091404/tree/main/cw2

the crop of down-sampling layers. The skip layers will be concatenated with input layers of each up-sampling residual block and then participate into the convolutions.

To train the network after building the architecture, a network training procedure need to be developed. The optimizer is based on Adam algorithm. The validation metric value is Intersection over Union (IoU)[2]. The loss is computed in a 2D Dice-based[3] loss function, which is essentially a measure of overlap between two samples. This measure ranges from 0 to 1 where a Dice coefficient of 1 denotes perfect and complete overlap. The Dice coefficient can be calculated as

$$Dice = \frac{2\,|A \cap B|}{|A| + |B|}$$

where |A ∩ B| represents the common elements between sets A and B, and |A| represents the number of elements in set A (and likewise for set B). For the case of evaluating it on segmentation masks, we can approximate |A ∩ B| as the element-wise multiplication between the prediction and target mask, and then sum the resulting matrix.

Because of the complexity of deep neural network, overfitting could become a serious problem. Regularisation is a technique which makes slight modifications to the learning algorithm[4], such as penalizing the weight matrices of the nodes, so that the model could generalize better and the model's performance is also improved. Two regularisation methods are implemented. One is data augmentation, which we use a geometric transformation on the original image, such as flipping that randomly flip the images in horizontal and vertical directions. Another is ensemble, which we choose is dropout that randomly zero out entire channels with probability.

In order to pass the image into the neural network, a data loader is needed. In this class, there are three basic functions. The first one is the class method, which records the basic features of data. To sample all the cases once without replacement in each epoch, the _getitem_ function would take case index as input parameter and loop the index after instantiating the data object. To make all frames in each case have equal chance to be sampled, the frame index is randomly set for each case to grab an image out. Two sampling ways of labels are implemented. One is randomly sample one label of the three for each frame as the label. Another is sample the consensus label for each image frame, which means add the three labels of each frame together and divided by 3, if the value of a pixel is above 0.5 then assign it as 1, otherwise assign it as 0.

2.2 Classification

To predict whether a given frame image frame contains prostate ('true') or not ('false'), we use one of the built-in 2D image classification networks, DenseNet, which is a multi-class classification network. To adapt it to a binary classifier, the "number of classes" parameter of the network is set to 2, and the input channel is set to 1. In the training process, densenet121 is implemented to speed up the process.

Similar to the segmentation, classification also needs a data loader to load the images and labels. For each frame, a consensus label is constructed by majority voting between the three classification labels at image level, i.e., each "true" vote should have two or more none-zero segmentation labels, whilst a "false" classification label is a result of two or more all-zero segmentation labels. To realize it, the method is to sum up all the pixel value of each label. If the total pixel value of the label is zero, we will believe there is no prostate in this label. If two or more of three labels have zero total pixel values, then we believe there is no prostate in this frame, which is 'false'. Otherwise, there is prostate in this frame and it is 'true'.

3 Experiments

To compare the two models with different kinds of labels, an experiment is implemented. In this experiment, holdout test set is loaded by the data loader defined. Then each model makes predictions. Their differences with the second kind of label in test set are calculated, summed up and recorded into a table. A Bland-Altman plot[5] is drawn based on the results, of which x axis represents the average value of the two model results and y axis represents the difference of them. Some example images with overlaid segmentation prediction and ground-truth are also shown.

The second experiment is to compare the segmentation results with and without the trained classification network applied first. Same as the first experiment, a holdout test set is loaded and the second kind of label is thought as ground-truth. The test results with and without the classification network are made a difference with the ground-truth separately and recorded into a table. A Bland-Altman plot is made in the same way. Because the DenseNet uses a linear function rather than a sigmoid function as output so the classification is based on the larger number of the two-class weight. Thus, a sigmoid function is applied to the output classification results.

4 Results

Some of numerical results are shown in Table.1. Top Fig.1 shows an example image with overlaid predications and ground-truth. There are clearly visible differences between the two models' predications. The Bland-Altman plot based on the results are shown in bottom Fig.1. In the plot, 92.5% points are in the acceptable limits, which also proves a not good enough agreement between the two kinds of label methods.

Part of numerical results of experiment 2 is shown in Table.2. The Bland-Altman plot based on the results are shown in bottom Fig.2, from which it is clear to see that about all data points are in the acceptable limits, which means that the segmentation results do not depend much on whether a classification model is applied prior to the segmentation, the improvement on IoU is about 5%. Result also shows that when the classification threshold is set to 0.5 will the highest IoU be achieved, which is 19.6%.

	difference1	difference2	mean	difference
0	-24.148478	7.933079	-8.107699	-32.081558
1	154.936462	82.334137	118.635300	72.602325
2	-6.617885	-38.412399	-22.515142	31.794514
3	-56.971893	-122.697342	-89.834618	65.725449
4	-88.453369	-212.131149	-150.292267	123.677780

Table 1.

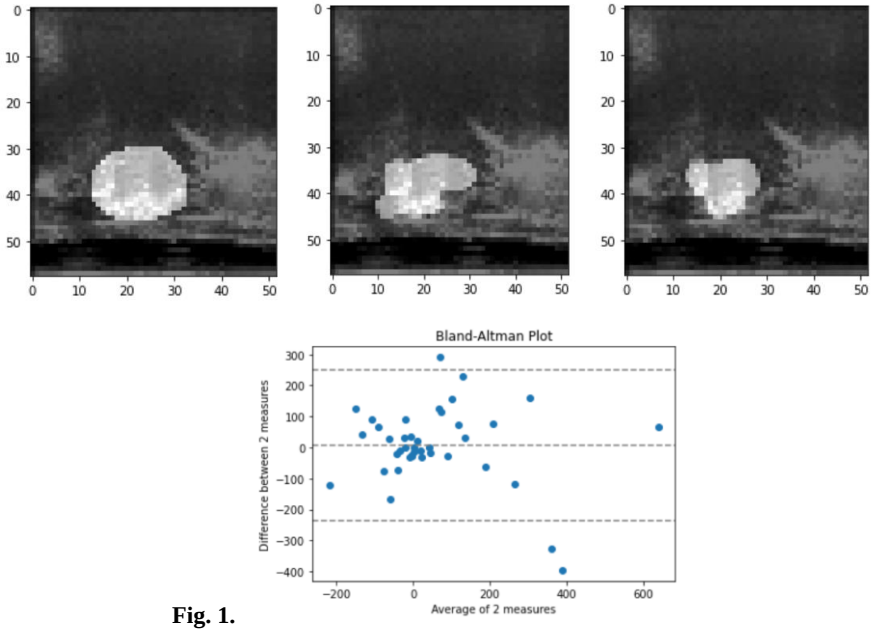

Fig. 1.

	without classificaiton	with classificaiton	mean	difference
0	174.724350	15.060257	94.892303	-159.664093
1	29.396124	17.554838	23.475481	-11.841286
2	485.646179	0.117819	242.882004	-485.528351
3	0.368016	0.512923	0.440470	0.144907
4	125.157372	0.135531	62.646450	-125.021843

Table 2.

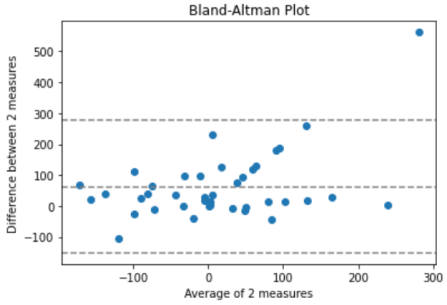

Fig. 2.

5 Conclusion

In this report a segmentation and a classification method based on deep convolutional neural networks are implemented. In the former part, a UNet architecture with proper data loader and regularisation methods are used. An experiment about the comparison of two models with different kinds of labels are implemented. Results show that the two models have some differences on segmentation performance but there is not a better one between them. In the classification part, a DenseNet that is adapted to binary classifier and a data loader with consensus labels are implemented. Another experiment compares the performance of with or without the classification model applied prior to the segmentation. Results show that there is no big difference between whether or not using the classification model first, which may because that almost all frames in test set have nonzero labels. An optimum classification threshold is used, which is about 0.5, to maximum the segmentation IoU to about 20%. The IoU does not depend much on threshold because the classification accuracy is quite high (97%).

6 Reference

1. arXiv:1505.04597 [cs.CV]
2. https://www.jeremyjordan.me/evaluating-image-segmentation-models/
3. https://www.jeremyjordan.me/semantic-segmentation/#loss
4. https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/
5. Giavarina D. (2015). Understanding Bland Altman analysis. Biochemia medica, 25(2), 141–151. https://doi.org/10.11613/BM.2015.015