Machine Learning in Medical Imaging CW2

Ruoxi Gao
20052569

University College London, London WC1E 6BT, UK
ruoxi.gao.20@ucl.ac.uk

Introduction

UNet has been firmly established as excellent architecture to implement segmentation task on all kinds of images including medical image. Skip connections/Shortcuts within ResNet has been widely used to avoid the problem of vanishing gradients, or to mitigate the degradation (accuracy saturation) problem. By combining standard UNet and residual blocks, a new architecture called ResUNet is proposed. Numerous efforts have been and are being continued to push the boundaries of ResUNet.

In this work, I take advantage of some ideas of ResUNet++ including squeeze and excite (SE) block, attention mechanism, and atrous spatial pyramidal pooling (ASPP) bridge to build ResUNet+[1]. Compared with standard ResUNet, the output of every encoder block is passed through a SE block, and an attention block is applied before decoder block. The goal of SE block is to improve the quality of representations produced by a network by explicitly modelling the interdependencies between the channels of convolutional features. Attention block executes a channel-independent feature weighting step, which aims at helping model find important information on decoded features.

Besdies, an optional adversarial architecture which aims to optimize model is included in appendix but not experimented. The adversarial segmentation utilizes adversarial idea of Generative Adversarial Network (GAN). However, the generator is replaced by segmentation network, and due to this, the discriminator is to classify groud-truth and prediction instead of fake and true[2].

Methods

Segmentation Network Architecture

The plot of architecture is in Fig 1. There are some core blocks' descriptions:

1. Residual Unit

The skip connections is implemented by passing input of standard encoder/decoder through a 1×1 convolutional operation to equalize channel of encoder/decoder output and shortcuts output. Addition operation on these two path outputs is followed before activation.

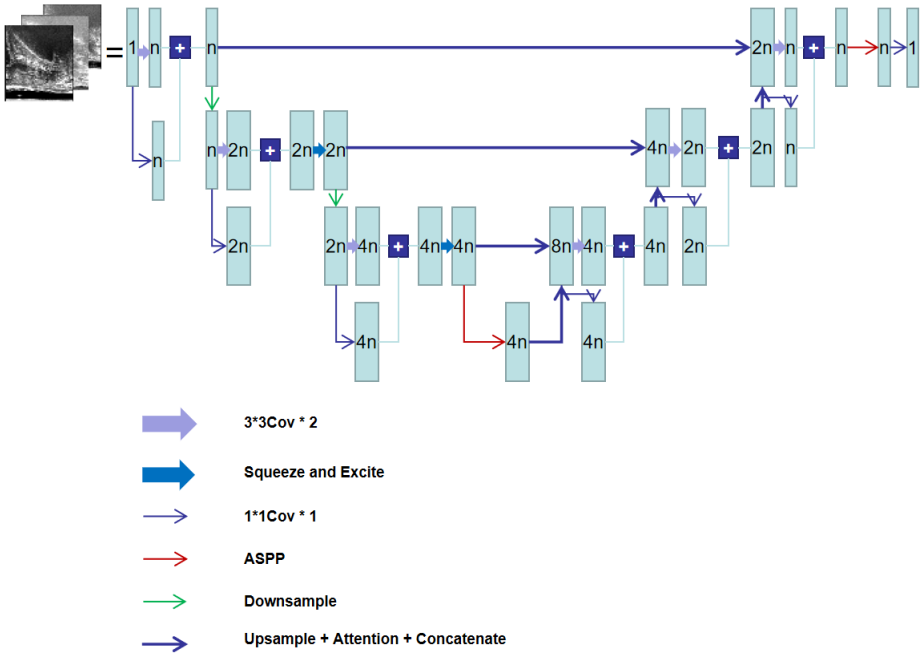

Fig. 1. Segmentation Network Architecture of ResUNet+

2. Attention Mechanism

Attention unit in this work is a pixel-level CNN attention[3]. Instead of directly concatenate encoder and decoder feature maps in standard UNet, decoder feature maps are weighed by a function of themselves and encoder feature maps to capture important information and selectively ignore insignificant information. Detailed architecture is in Appendix.

3. SE Block

SE adaptively recalibrates channel-wise feature responses by explicitly modelling interdependencies between channels. In this way, the learning of convolutional features is expected to be enhanced so that the network is able to increase its sensitivity to informative features which can be exploited by subsequent transformations[4]. Detailed architecture is in Appendix.

4. ASPP Module

ASSP is a semantic segmentation module for resampling a given feature layer at multiple rates, the resampling procedure is implemented through multiple parallel atrous convolutional layers with different sampling rates rather than true resampling. This amounts to probing the original image with multiple filters that have complementary effective fields of view, thus capturing information context at multiple scales[5]. Detailed architecture is in Appendix.

Ensemble Method

Add a dropout layer after every encoder and decoder block to prevent overfitting.

Classification

Choose DenseNet to do classification prior to segmentation. The classification step filters some frame which not contain gland to improve segmentation perfirmance. By assigning 1 to *num_classes* variable of built-in densenet class, and modifying input channel number of the first layer of feature sequential in built-in densenet class to 1, the classification network can adapt to our dataset.

Experiments

Split dataset into train and test set. The first 160 cases are used as train set and other cases as test set. Denote Random Sampling without Classification as Model 1, Consensus Sampling without Classification as Model 2, and Consensus Sampling with Classification as Model 3. Note that hyperparameters of segmentation networks are not carefully tuned. All networks are trained by 100 epochs. Batch size of segmentation is 4 and that of classification is 16. The test results shown in next section are obtained by taking 100 trials average on test set, namely, randomly sample all cases 100 times to get average metrics.

Data Augmentation

Augment images by using linear combination of two frames of same cases as some fake training images. Every epoch trains all train set cases, each of which samples one true frame and one augmented frame.

$$\mathbf{img}_{aug} = \lambda \mathbf{img}_1 + (1-\lambda)\mathbf{img}_2 \tag{1}$$

$$\mathbf{label}_{aug} = \Big(\lambda \mathbf{label}_1 + (1-\lambda)\mathbf{label}_2\Big) > 0.5 \tag{2}$$

Dataloader

By saving the number of frame for each case, each epoch can sample all cases once with random selected frame. In detail, randomly sample one number between 1 and corresponding frame number during each case sampling, and then use this number to find the sampled frame. The dataloader should split into train or test. During train procedure, len class function returns 160 (the number of train case), case number is equal to current index for getitem function. However, while during test procedure, len class function returns $200 - 160 = 40$, case number is $index + 160$. After data augmentation, the number of sampled frame for each case becomes twice, including true and augmented frame. Therefore, some modification should be added on train part of dataser class. First, len class function returns $160 \times 2 = 320$, and when index of getitem function is not less than train case number, case number is $index - 160$, otherwise, it is equal to current index. Besides, data augmentation procedure needs two sampled frame

for each case, so the number of sampled frame when augmenting for each case becomes 2 instead of 1. For two sampling methods in dataset class:

1. Random sampling: randomly choose one number between 0 and 2 to indicate sampled random label.

2. Random sampling: add all labels on pixel level, if the sum result is larger than 1, the pixel level label is 1, otherwise, it is 0.

Results

Numerical Results

	Model 1	Model 2	Model 3
Dice Loss:	0.42707	0.42728	0.47761
F1:	0.76100	0.77451	0.80649
Accuracy:	0.93248	0.94053	0.95362

Table 1. Validation Metrics

Visualized Results

In Appendix.

Conclusion

If two segmentation results is approximately identical, scatter points of Bland-Altman plot will be around horizontal zero axis. From the Bland-Altman plot comparing two sampling methods, contour is diamond. Split whole contour into four slashes:

Left Top: $x_1 + x_2 = x_1 - x_2 \rightarrow x_2 = 0$

Left Bottom: $x_1 + x_2 = x_2 - x_1 \rightarrow x_1 = 0$

Right Top: $x_1 + x_2 - 2 = x_2 - x_1 \rightarrow x_1 = 1$

Right Bottom: $x_1 + x_2 - 2 = x_1 - x_2 \rightarrow x_2 = 1$

Therefore, this plot means there are some pixels whose outputs from two sampling methods are independent, and the closer to contour points are, the more independent results are.

From the Bland-Altman plot comparing with and without classification, the slash which ends on mean = 0.5 indicates $x_1 + x_2 = x_1 - x_2 \rightarrow x_2 = 0$, this means there are some pixels is predicted as 0 no matter what probability segmentation network outputs. Apart from these unique points, other segmentation results are identical which lines along zero horizontal axis. This situation is correspondent to function of classification, where points on slash are who are classified as 0.

The numerical results shows consensus sampling is superior than random sampling, and classification improves performance.

For threshold search, by comparing the accuracy curve with and without classification, we conclude the most obvious improvement occurs when classification threshold is 0.2, so the threshold optimum is 0.2.

References

[1] Jha, D., Smedsrud, P.H., Riegler, M.A., Johansen, D., De Lange, T., Halvorsen, P. and Johansen, H.D., 2019, December. Resunet++: An advanced architecture for medical image segmentation. In 2019 IEEE International Symposium on Multimedia (ISM) (pp. 225-2255). IEEE.

[2] Xue, Y., Xu, T., Zhang, H., Long, L.R. and Huang, X., 2018. Segan: Adversarial network with multi-scale l 1 loss for medical image segmentation. Neuroinformatics, 16(3), pp.383-392.

[3] Oktay, O., Schlemper, J., Folgoc, L.L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N.Y. and Kainz, B., 1804. Attention u-net: Learning where to look for the pancreas. arXiv 2018. arXiv preprint arXiv:1804.03999.

[4] Hu, J., Shen, L. and Sun, G., 2018. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7132-7141).

[5] Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A.L., 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence, 40(4), pp.834-848.

Appendix

Bland-Altman Plot

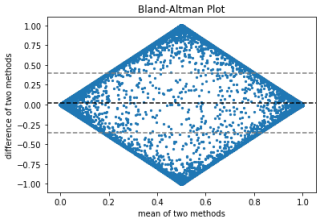

Fig. 2. Two Sampling Methods

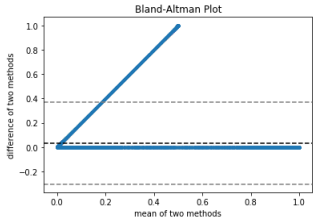

Fig. 3. With and Without Classification

Accuracy V.S. Threshold Curve

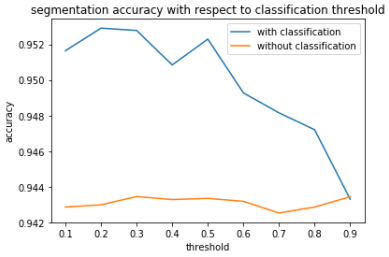

Fig. 4. Optimum Threshold Search

Segmentation Example Images

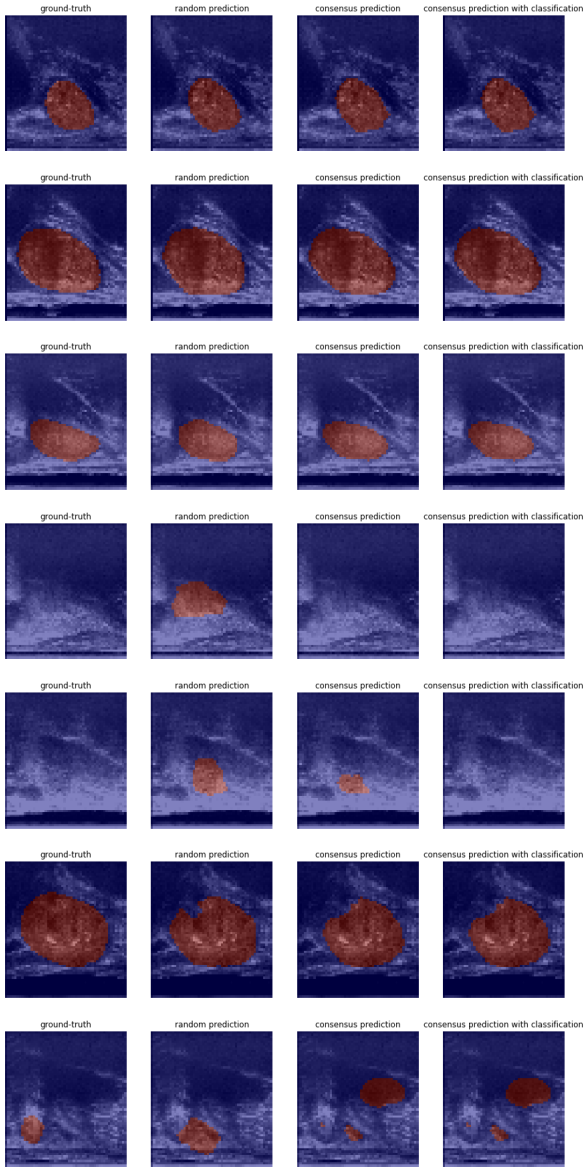

Fig. 5. Segmentation Visualization

Attention Architecture

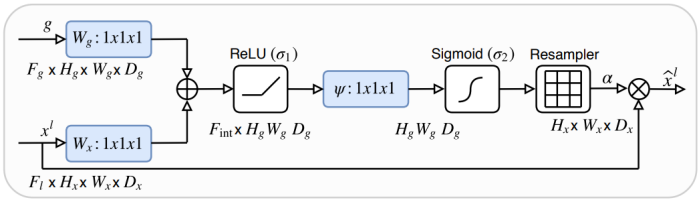

Fig. 6. Details of Attention Mechanism [3]

SE Architecture

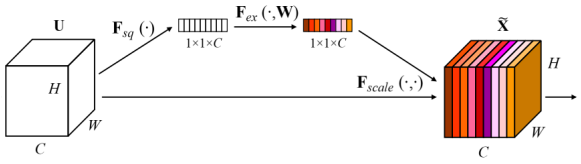

Fig. 7. Details of Squeeze and Excite Block [4]

ASPP Architecture

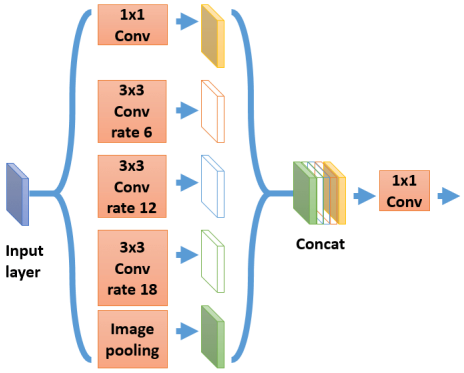

Fig. 8. Details of ASPP Module [5]

Adversarial Network

The adversarial procedure is divided into two steps. Firstly, train discriminator to classify ground-truth and prediction. Secondly, train segmentation network to confuse discriminator. The procedure is an adversarial game between discrimination network and segmentation network, which is expected to help segmentation network learn a segmentation map similar to ground-truth. Discriminator architecture exploration is not included in this work.

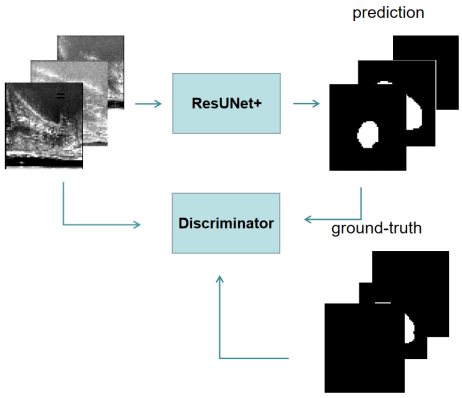

Fig. 9. Optional Adversarial Architecture

Denote segmentation network as S, discrimination network as D. Loss function becomes:

$$Loss_{\mathbf{S}} = L_{Dice}\left(\mathbf{S}(frame), label\right) + \lambda L_{BCE}\left(\mathbf{D}\Big(frame, \mathbf{S}(frame)\Big), 1\right) \quad (3)$$

$$Loss_{\mathbf{D}} = L_{BCE}\left(\mathbf{D}\Big(frame, \mathbf{S}(frame)\Big), 0\right) + L_{BCE}\left(\mathbf{D}(frame, label), 1)label\right) \quad (4)$$