Ultrasound Nerve Segmentation

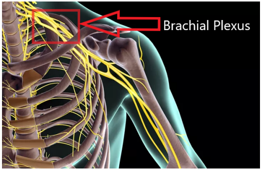
Jia-Wei Liao, Kuok-Tong Ng, Yi-Cheng Hung

Team 25 OPML
Department of Applied Mathematics, NYCU

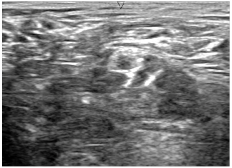
January 6, 2022

Introduction

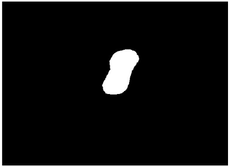
- **Dataset:** Ultrasound Nerve Segmentation contain 5635 training and 5508 testing grayscale image. The size of the image is (580, 420).
- **Goal:** Automatically segment Brachial Plexus (BP) nerve structures in ultrasound images of the neck.



BP



Image

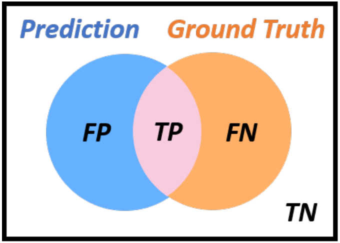

Mask

Metrics

Dice Similarity Coefficient (DSC)

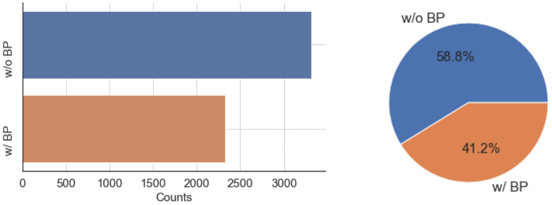
Let TP be the true positive, FP be the false positive and FN be the false negative. The DSC is defined as

$$DSC = \frac{2TP}{2TP + FP + FN}$$

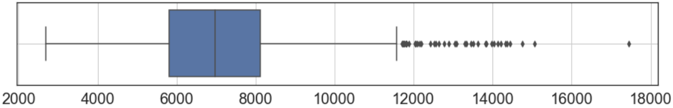


Data Explore

- Number of training masks with and without BP are 2323 and 3185, respectively, which caused data imbalanced.

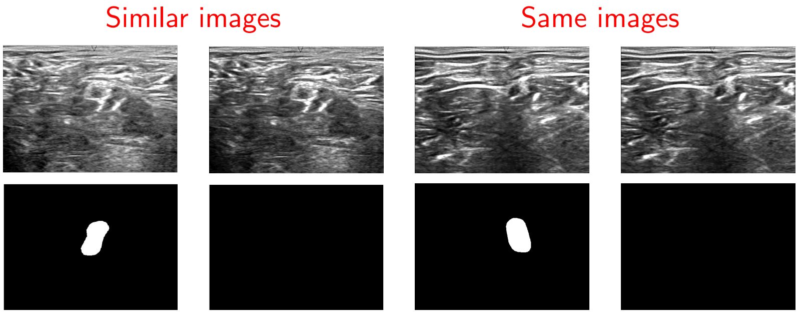


- The box plot shows the pixels count in the mask with BP. The minimum value is 2684.



The Difficulties

- Brachial Plexus does not exist in a most masks (58.8%).
- Annotators were trained by experts.
- Identical images but different masks.



Approach

We will introduce the following approach:

- Data Pre-processing
- Erosion Mask Smoothing
- Model Architecture
- Segmentation Loss
- Adaptive Single Model Ensemble

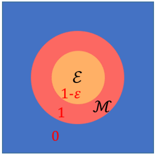
Data Pre-processing

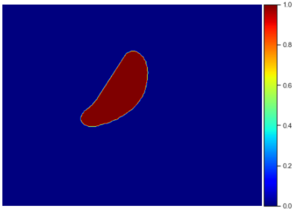
- Data Cleansing
- Splitting training and validation sets in 4:1
- Resize the image from (580,420) to (576,448)
- Randomly flips
- Randomly adjust brightness
- Randomly add noise

Erosion Mask Smoothing (EMS)

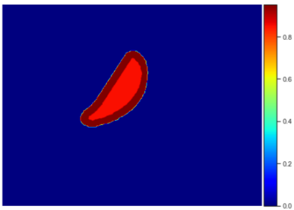
Let $\mathcal{M}$ be the mask, $\mathcal{E}$ be the interior region after eroding mask, and p be the pixel in the mask.

$$\tilde{\mathcal{M}}(p) = \begin{cases} 1-\varepsilon, & \text{if } p \in \mathcal{E} \\ 1, & \text{if } p \in \mathcal{M} \setminus \mathcal{E} \\ 0, & \text{if } p \in \mathcal{M}^c \end{cases}$$





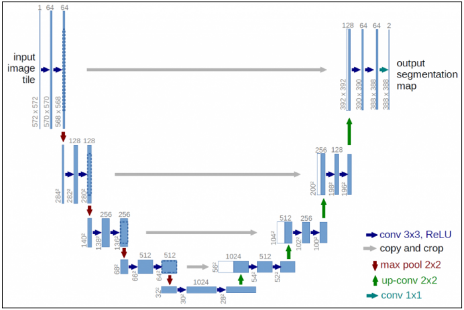
Origin



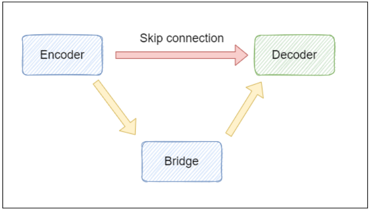
EMS

Model Architecture

- In this task, we choose UNet as our model architecture.



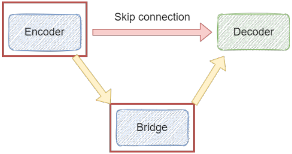
Original architecture



Topological structure

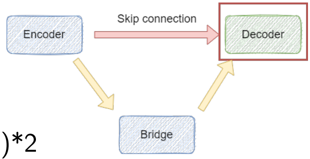
Model Architecture

- Any backbone with downsampling steps can be treated as an encoder.
- Use a pretrained backbone in order to get a better result. (e.g. ResNet, EfficientNet, ec.)
- Treat the last stage of the backbone as the Bridge.



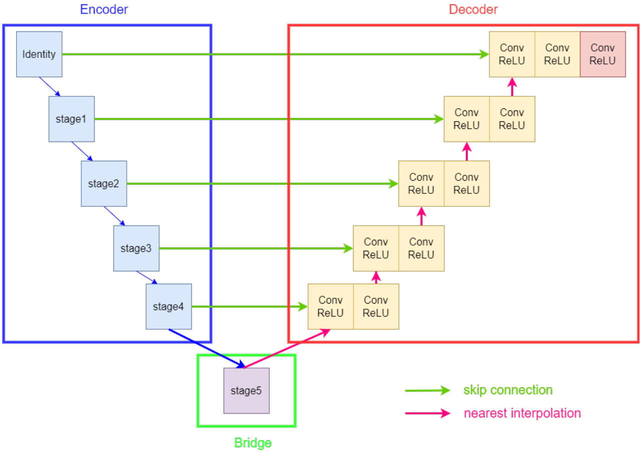
Model Architecture

- Follow the original UNet architecture except replacing "Up-convolution" by "nearest interpolation".
- Each decoder stage: Interpolation+concatenation+(Conv+ReLU)*2
- Apply 1x1 convolution layer in the final to predict the class.



Model Architecture

- Therefore, the model architecture looks like



Segmentation Loss

- **Dice Loss:** Focused on object region

$$\mathcal{L}_{DSC}(\hat{Y}, Y) = 1 - \frac{2|Y \cap \hat{Y}|}{|Y| + |\hat{Y}|},$$

- **Focal Loss:** Adjusted the probability distribution of prediction

$$\mathcal{L}_{FL}(\hat{Y}, Y) = -\sum_{i,j} \alpha \left(1 - \hat{Y}_{i,j,C}\right)^{\gamma} \log \hat{Y}_{i,j,C}$$

- **Segmentation Loss:**

$$\mathcal{L}_{Seg}(\hat{Y}, Y) = \mathcal{L}_{DSC}(\hat{Y}, Y) + \mathcal{L}_{FL}(\hat{Y}, Y)$$

Adaptive Single Model Ensemble

Let T_k be the transforms, I be the image, and f is segmentation model.

Step 1: Predicted $\{I, T_1(I), T_2(I), T_3(I)\}$. Then we have

$$\{f(I), f(T_1(I)), f(T_2(I)), f(T_3(I))\}$$

Step 2: Compute the dice adjacency matrix

DSC	Origin	Fliplr (T_1)	Flipud (T_2)	Fliplr+ud (T_3)
Origin	1	0	0.52	0.79
Fliplr (T_1)	0	1	0	0
Flipud (T_2)	0.52	0	1	0.65
Fliplr+ud (T_3)	0.79	0	0.65	1

Adaptive Single Model Ensemble

DSC	Origin	Fliplr (T_1)	Flipud (T_2)	Fliplr+ud (T_3)
Origin	1	0	0.52	0.79
Fliplr (T_1)	0	1	0	0
Flipud (T_2)	0.52	0	1	0.65
Fliplr+ud (T_3)	0.79	0	0.65	1

Step 3: Solving the eigenvector of A w.r.t the largest eigenvalue

$$v = [0.5810 \quad 0 \quad 0.5341 \quad 0.6141]^\top$$

Step 4: take v to the softmax

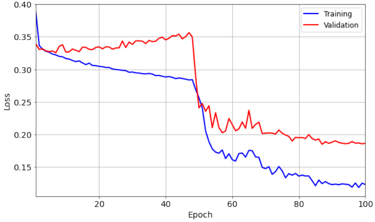
$$[c_0, c_1, c_2, c_3]^\top = [0.3291 \quad 0.0227 \quad 0.2651 \quad 0.3831]^\top$$

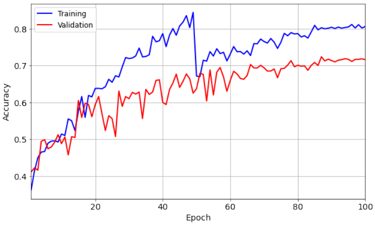
Step 5: Weighted sum

$$\hat{Y} = c_0 f(I) + \sum_{k=1}^{3} c_k (T^{-1} \circ f \circ T)(I)$$

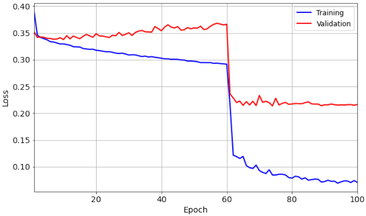
Experiment Result

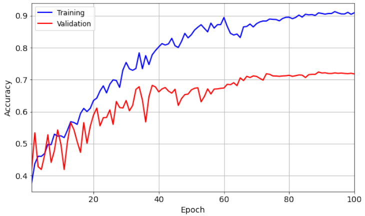
EfficientNet-b1





EfficientNet-b1+EMS





Experiment Result

The following table shows model scores with different backbone.

Model	Backbone	Validation	Test public	Test private
UNet	ResNet34	0.67643	0.68530	0.69100
UNet	ResNet50	0.66040	0.67340	0.67828
UNet	ResNeXt50	0.67162	0.66699	0.66704
UNet	ResNeSt26d	0.71936	0.67015	0.69009
UNet	RegNet32	0.71839	0.67906	0.68959
UNet	EfficientNet-b0	0.70170	0.68135	0.69254
UNet	EfficientNet-b1	**0.72408**	**0.70332**	**0.70111**

- ResNet34 and ResNet50's loss are not stable.
- EfficientNet-b1 perform better than the other encoder.
- The baseline in this task is 0.70753.

ASME Result

The table shows the improvement after applying model ensemble.

Model	Backbone	Origin	ASME
UNet	ResNet34	0.69100	0.71031 (+0.01900)
UNet	ResNet50	0.67828	0.70857 (+0.03029)
UNet	EfficientNet-b0	0.68959	0.70233 (+0.01274)
UNet	EfficientNet-b1	**0.70111**	**0.72341** (+0.02300)

- Implementing this ASME increases private scores by $1 \sim 3\%$.
- The baseline in this task is 0.70753.

Conclusion

1. Proposed Erosion Mask Smoothing to maintain loss stability.
2. Apply UNet based model in this task.
3. Proposed Adaptive Single Model Ensemble which can adaptive the weight of aggregation by itself.
4. Comparing the result with different combinations of encoder and ASME.
5. Combining UNet with EfficientNet-b1 as encoder, EMS and ASME, we obtain a best private dice score 0.72341.

Submission and Description	Private Score	Public Score	Use for Final Score
answer.csv just now by Jia-Wei Liao	0.72341	0.71520	☐

Reference

1. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, Deep Residual Learning for Image Recognition, *CVPR*, 2015.
2. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár, Focal Loss for Dense Object Detection, *CVPR*, 2017.
3. Olaf Ronneberger, Philipp Fischer, and Thomas Brox U-Net: Convolutional Networks for Biomedical Image Segmentation, *CVPR*, 2015.
4. Mingxing Tan, and Quoc V. Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, *ICML*, 2019.

Thank you for your attention.