Deep Segmentation for Mouse Embryo Brain Ventricles

Hui Wei
hw1666@nyu.edu

December 17, 2018

Abstract

Brain ventricles (BV) are important for exploring the brain and the central nerve system for mammals. High frequency ultrasound (HFU) is the only rapid way to get high-resolution images for BV. In this work, we try to segment BV from HFU images of entire mouse embryos using deep learning method. The system includes two main parts: localization and segmentation. A sliding window in the localization part plays the role of attention, and a CNN classifier will determine which part contains BV. Then a deep model will be directly applied to 3D BV sub-volumes to get the segmentation. This deep segmentation model over-performs the traditional segmentation method by a large margin on the 111 test set volumes.

1 Introduction and Related work

Brain ventricle (BV) (please see Figure 1) is a Y-shape part in mammal brains containing cerebral spinal fluid. Since the genes of mice are similar to humans and their BV have same phenotype, doing research on BV part of mice paves the way for doing the similar experiment on human brains. Traditionally, segmenting BV parts asks for the tedious labor work taking 15 minutes on average for a expert to label slice by slice of 3D volumes, which is time-consuming and expensive.

Recently, an image processing method [1,2,3], Nested Graph Cut (NGC), appeared to solve this problem. In those work, they had the same definitions for nodes, edges of images, and defined the source S and sink T nodes, like graph cut, which is one of the traditional computer vision techniques for segmentation. Then, by solving the energy function designed using the prior knowledge, NGC can segment nested structure very well, as Figure 1 shows. However, the prior NGC based work only trained on 36 BV volumes obtained before, the performance of it decreases quickly when applied on our larger dataset which contains 259 training and 111 test images. Also, for getting a good result, NGC relies on the expertise knowledge for BV structure, which limits the application area of this method.

From 2012, computer vision field has been pushed forward fast by the application of deep neural network. CNN [4, 5], inspired by the visual system of animals, has dominated the area and outperforms traditional computer vision

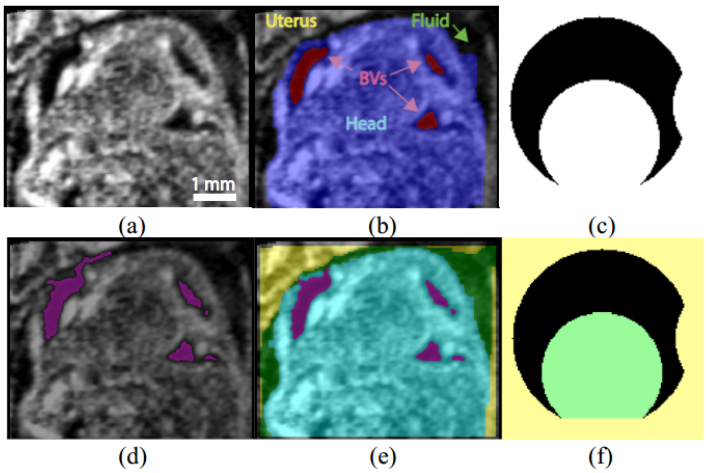

Figure 1: (a) a slide of a 3D BV image. (b) the nested structure of mouse embryo. (d) the BV segmentation result of NGC method [3]. (e) overall nested structure segmentation result of NGC method. (c)(f): NGC can correctly segment the nested structure and automatically add the boundary between different parts.

methods in different problems. Drawing on shared parameters and the hierarchy structure, CNN does not depend on the prior knowledge anymore. Instead, it can extract and learn the features from images automatically. Since then, more and more successor models [6,7,8,9] approach or surpass the human performance on classification tasks.

Different from classification, which assigns an entire image to various categories, segmentation needs to associate each pixel/voxel with different classes or objects. J.Long et.al [10] first applied CNN to semantic segmentation. In their fully convolutional network (FCN), all the fully connected layers are replaced by the convolutional layers, and high-resolution and low-resolution feature maps are combined together to fuse coarse, semantic and local, appearance information. They also designed a deconvolutional layer to map the low-resolution feature map to the one of original image size, which is in order to predict the class pixel by pixel.

For biomedical image, due to the fact that there is always the issue of privacy and needs expertise knowledge to label, the number of it is scarce. Based on fully convolutional network mentioned above, U-Net [11] is a symmetric structure, with several up-sampling layers added. Like FCN, U-Net concatenates local and global information together; while unlike FCN, it divides the whole structure into several stages, in each stage, several convolutional layers are added, and excessive data augmentation are utilized to make up for the little data size.

U-Net was the state-of-the-art model for 2D biomedical image segmentation task. However, in biomedical image analysis, 3D images are more common. To convert U-Net to solving 3D image segmentation, V-Net [12] adapts each kernel into 3D and still has down-sampling and up-sampling phases. In particular, like [8], V-Net needs to learn the inter-layer residual functions, and for the loss function, it uses Dice loss function which intuitively, evaluates on the whole

sub-volume instead of the single voxel.

In this work, we designed a V-Net like network to segment BV part from the whole mouse embryo HFU 3D volumes. Here, since BV part only occupies a extremely small part of the whole embryo (on average, 0.335%, which is smaller than the proportion in the dataset V-Net applied on), we cannot apply V-Net directly to our task. To get a sub-volume whose BV part is much larger, a sliding window cuts the original volume into several sub-cubes and then a classifier detects whether each cube contains BV with certain confidence. After obtaining BV-covered subvolumes, a deep segmentation network segments the BV part from each of them.

2 The Architecture

2.1 Localization

Like [13], for localization, we first apply a 3D sliding window to get "proposals" of the 3D image, and then use the trained classifier to determine whether it the "proposal" contains the whole BV part.

Taking advantage of the observation that the maximum size of BV is 128 pixels, we fix the size of the sliding window as $128 \times 128 \times 128$, in order to contain the largest BV. When any side of the whole embryo image is less than 128, we zero pad that axis to 128. Due to the limited computing resource, we use sliding windows of $64 \times 64 \times 64$, and then change sub-volumes containing entire BV back to the original size.

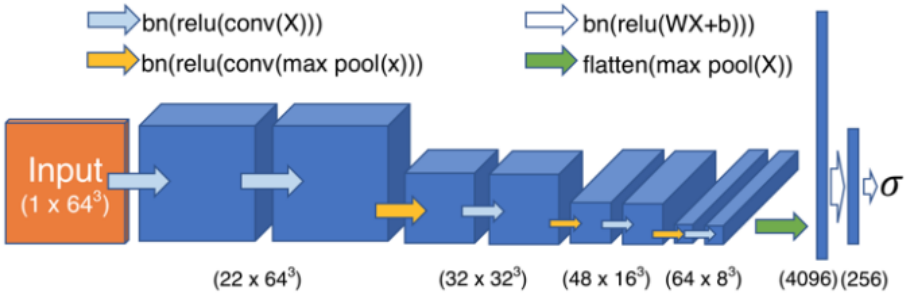

Figure 2: Architecture for localization. Note the input is $64 \times 64 \times 64$. After localization, they need to be upsampled by the factor of 2 to recover the original size. The last layer has softmax function to get the classification probabilities for positive and negative category.

For the classifier, VGG net [7] is adapted to 3D and used for classifying each subvolume extracted from the entire image as contained the whole BV or not. As Figure 2 shows, there are eight conv layers and three fully connected layers in total, and each conv and fully connected layer (except the last layer) is followed by ReLU non-linearity and batch normalization. Between the network, 2×2 max pooling layers is for downsampling feature maps. In order to avoid overfitting, dropout layers are added to each layer except the last one, with the rate of 0.15, and after the first fully connected layer, the rate becomes 0.4. For

getting the probability of each subvolume, the last layer is 2 dimensional, and followed by a softmax layer. The kernel size of each conv layer is 3 and stride is 1 for extracting the feature maps.

2.1.1 Segmentation

After getting the candidate "proposal" box, we feed them into a V-Net like network to get the final segmentation of BV. Please note that after the localization part, we upsample the box from $64 \times 64 \times 64$ back to $128 \times 128 \times 128$.

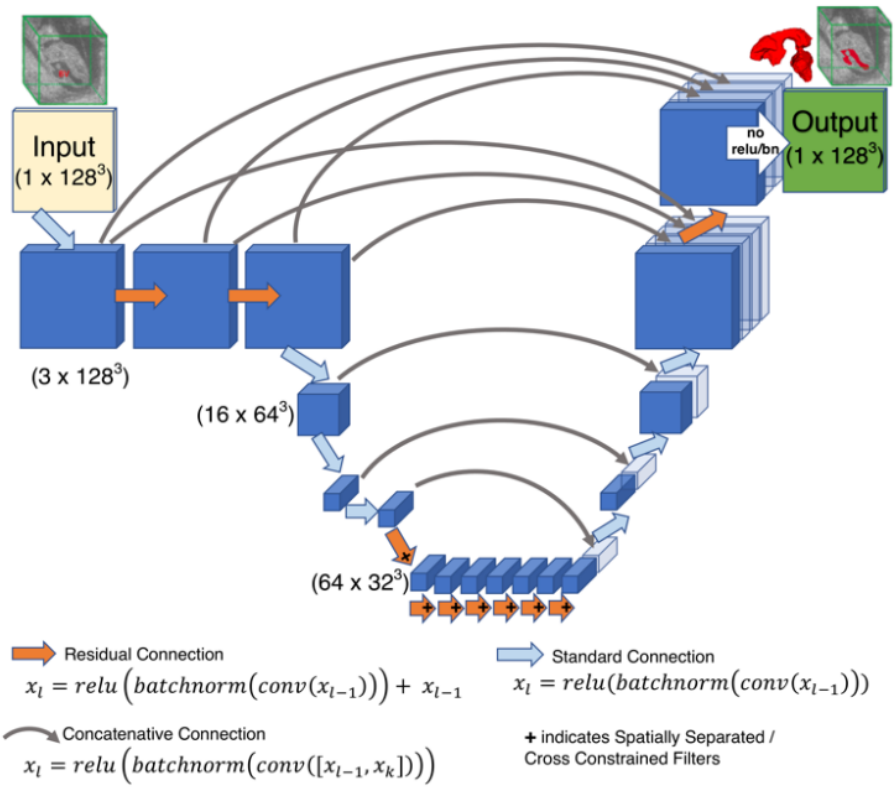

Figure 3: Architecture for Segmentation. There are 7 LRP layers to add the depth of the network. Each output from the compression stage is concatenated to the corresponding decompression stage. The last layer uses sigmoid function to get the probability for each voxel.

In Figure 3, we can see that similar to V-Net, our network is also divided into several stages, and for each stages, the conv layers instead of pooling layers to downsample feature maps. The trick combining the coarse and finer feature map is also used here to get a better performance, since in mouse embryo images, there are always some difficult cases, for example, the border of the BV is vague. Without the global context information, even a human expert cannot make sure whether some part belongs to BV or not. Also, for solving this problem, we

add several low resolution processing (LRP) layers for the lowest stage, whose receptive field is largest compared to the prior layers.

Since our computing resource is limited and LRP layers are compute-dense, we use spatially separated / cross constrained filter structures for LRP layers. Basically, spatially separated constrained filter uses 1D conv kernel for each axis (e.g. $k \times 1 \times 1$ for x-axis) and then sums up the three feature maps obtained by convolving 1D kernels with the whole volume. Due to this trick, we succeed reducing the total parameters from 15M to 2M, avoiding sever overfitting.

In the whole network, the kernel size for every conv layers is 7, and the stride is 2 for downsampling convolution layers and upsampling deconvolution layers. For the last layer, we have one output channel for each voxel, and use sigmoid function to get the probability for each voxel prediction.

3 Experiment

3.1 Dataset and Data Augmentation

For segmentation, we used Amira [14] to label 370 mouse embryo 3D HFU volumes, whose size is varied from $150 \times 161 \times 81$ to $300 \times 281 \times 362$. Each voxel stands for $50 \times 50 \times 50$ μm. According to the empirical 7:3 division, the whole dataset is divided into 259 training data and 111 test data.

As mentioned before, biomedical image analysis always encounters the problem of extreme shortage of training data. As have down in previous work [11, 12], we augment the training data on the fly for each training epoch. Here, flipping and rotation are considered. For each image, the probability of flipping or rotation is 0.7. Each image can be flipped along x, y, z or any combination between them, and rotated along each axis by 90, 180 or 270 degree. The same augmentation process is applied for both localization and segmentation.

3.2 Training

3.2.1 Localization

Since the proportion of BV is extremely small compared to the entire mouse embryo body, the number of the negative samples, which does not contain the entire BV, is much larger than that of the positive samples. Therefore, to extract positive and negative training samples for the classifier from the original 259 training volumes, the sliding window has different strides: 3 for extracting negative and 2 for positive samples. Here, we define the positive samples as the $128 \times 128 \times 128$ subvolume containing at least 99% of BV, while the negative samples only contains less than 80% of BV part. Others are considered as ambiguous. Due to the shortage of computing resource, we downsample the whole image by the factor of 2, and then use $64 \times 64 \times 64$ sliding window.

Cross-entropy loss is used for improving the classification accuracy. As mentioned before, the number of negative samples is far more than that of positive samples, so we add the corresponding weights (1 for negative, 1.2 for positive) to the cross-entropy items. With the loss function, we trained the classifier for 5 epochs with SGD, momentum 0.9, and weight decay rate 0.00001, and set the initial learning rate to 0.01 and degrade by 0.1 after the third epoch. 200 is used for mini-batch size.

3.2.2 Segmentation

Similar to V-Net [12], we use Dice based function (DSC), which is defined as follows, for loss function.

$$DSC = \frac{\epsilon + 2\hat{Y}Y}{\epsilon + \sum_{\hat{y}\in\hat{Y}} \hat{y} + \sum_{y\in Y} y}$$

where $\hat{y} \in [0, 1]$, denoting the predicted probability for each voxel, $y \in \{0, 1\}$, the groundtruth label for each voxel, and $\epsilon = 1 \times 10^{-4}$ is used for smoothness and numerical zero issues. The DSC is measured on the whole volume instead of the single voxel.

After extracting from the original 259 training images with $128 \times 128 \times 128$ sliding window with at least 97% BV part, we get 64K $128 \times 128 \times 128$ training subvolumes for the segmentation network. Then we trained it for 5 epochs with SGD of momentum 0.9, weight decay 0.0001, initial learning rate 0.01, and degrades by 0.1 after the third epoch. In addition, 4 subvolumes are used for each mini-batch.

3.3 Testing

For localization part, stride 3 is utilized for getting the candidate volumes for test image, and if the probability predicted by the classifier is more then 95%, then we define it as containing the whole BV. Because of the random initialization and augmentation, we trained 3 different models with the same aforementioned architecture and methods, tested on them, and used the arithmetic mean of three models to get the ultimate test accuracy.

After getting the candidate subvolumes which are considered as containing the whole BV, we upsampled them by the factor of 2 and feed them into the segmentation network, if the probability for each voxel is more than 92%, it is belonged to BV part.

4 Results and Discussion

Figure 4 shows the segmentation results. For each line, the gray images are slides along three axis respectively, the red part is the segmentation result for each slide, and 3D mesh is the final result. We can see that the performance of the model is good enough to adapt to different BV orientations, sizes and postures.

To get the accuracy of the segmentation, we still use DSC to evaluate the final result. In the experiment, prior NGC-based method [3] was used as the baseline, then for the deep segmentation method, we used 2 models: 1) the model mentioned before (single) 2) three localization model combined together, computer the arithmetic mean, then use the V-Net like network to segment (combined). From Table 1, we can see that compared to the baseline, the deep learning method has a greatly better performance due to the excellent feature extraction ability of CNN.

Although the deep segmentation method has the performance gain by a large margin, there are still several shortages for this system to be improved for the future work. First, like R-CNN [13], the localization needs to compute

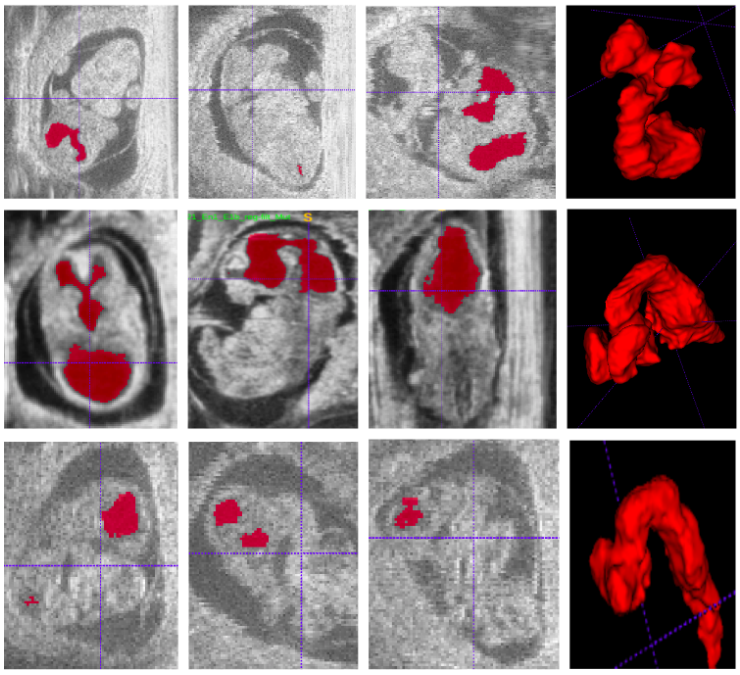

Figure 4: Segmentation results. The first three columns are slides from different axis, respectively. Red parts belongs to BV. The 3D mesh on the last column is the 3D combination from the whole volume.

Table 1: DSC for different models

Models	**Mean DSC**
NGC-based	0.7119
Deep Segmentation (single)	0.8911
Deep Segmentation (combined)	**0.8956**

the convolution over different subvolumes. This does not take advantage of [15, 16, 17], which shares one convolution computing for the whole image, thus greatly reduce the inference time. In addition, transfer learning methods could be explored to see whether we can use state-of-the-art model V-Net architecture to this problem.

5 Conclusion

In this work, we proposed a deep learning based segmentation method for brain ventricles in 3D mouse embryo images. Due to the parameter sharing and hierarchy architecture, this method outperforms the state-of-the-art traditional NGC model by a large margin. DSC loss does not need to re-weight when the foreground and background samples are extremely unbalanced and avoid the network stuck in the local minima. Our future work will focus on speeding up the inference process, which will draw on the idea in the state-of-the-art object

detection algorithms.

6 Acknowledgements

The author of this paper thanks Ziming Qiu from Electrical Engineering Department, NYU Tandon School of Engineering, for giving the idea and suggestions for this work. The author also thanks professor Rajesh Ranganath , who gave an entirely amazing and excellent course in this semester, through which the author got through the problems and methods in machine learning for healthcare, and learnt how to model the real-world problem using mathematics, then use corresponding machine learning methods to tackle them.

References

[1] J. W. Kuo, Y. Wang, O. Aristizabal, D. H. Turnbull, J.Ketterling, J. Mamou, Automatic Mouse Embryo Brain Ventricle Segmentation, Gestation Stage Estimation, and Mutant Detection from 3D 40-MHz Ultrasound Data. *Ultrasonics Symp. (IUS)*, 2015 IEEE Int., pp. 1-4.

[2] J.W. Kuo, J. Mamou, O. Aristizabal, X. Zhao, J. A.Ketterling, and Y. Wang, Nested Graph Cut for Automatic Segmentation of High-Frequency Ultrasound Images of the Mouse Embryo. *IEEE Trans. Med. Imag. (MI), vol. 35, no. 2, pp. 427-441*, 2015.

[3] J.W. Kuo, Z. Qiu, O. Aristizabal, J. Mamou, D. H.Turnbull, J. Ketterling, and Y. Wang, Automatic Body Localization and Brain Ventricle Segmentation in 3D High Frequency Ultrasound Images of Mouse Embryos. *2018 IEEE 15th Int. Symp. Biomedical Imaging (ISBI), pp. 635-639, IEEE.*

[4] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, et al. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, 1990.

[5] A. Krizhevsky, I.Sutskever, and G.E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS, pp. 1106–1114*, 2012.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[7] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[9] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[10] J. Long, E. Shelhamer, and T. Darrell, Fully convolutional networks for semantic segmentation. in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[11] O. Ronneberger, P. Fischer, and T. Brox, U-net: Convolutional Networks for Biomedical Image Segmentation. *Int. Conf. on Medical Image Computing and Computer-assisted Intervention (MICCAI), pp. 234-241*, 2015.

[12] F. Milletari, N. Navab, and S. A. Ahmadi, V-net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. *2016 4th Int. Conf. on 3D Vision(3DV)*, pp. 565-571, IEEE, 2016.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation. in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[14] H Hege (2018). AMIRA. [Online] Thermo Fisher Scientific. Available: https://www.fei.com/software/amira-for-life-sciences/.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*. 2014.

[16] R. Girshick. Fast R-CNN. In *ICCV*, 2015.

[17] S. Ren, K. He, R. Girshick, and J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks. in *Neural Information Processing Systems (NIPS)*, 2015.