# A Soma Segmentation Benchmark in Full Adult Fly Brain Supplementary Material

Xiaoyu Liu[1] Bo Hu[1] Mingxing Li[1] Wei Huang[1] Yueyi Zhang[1,2] Zhiwei Xiong[1,2,*]
[1]University of Science and Technology of China
[2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
{liuxyu, hubosist, mxli, weih527}@mail.ustc.edu.cn, {zhyuey, zwxiong}@ustc.edu.cn

## 1. Visualization

Here we provide additional visual results and implementation details that have not been presented in the main paper.

### 1.1. Video Visualization

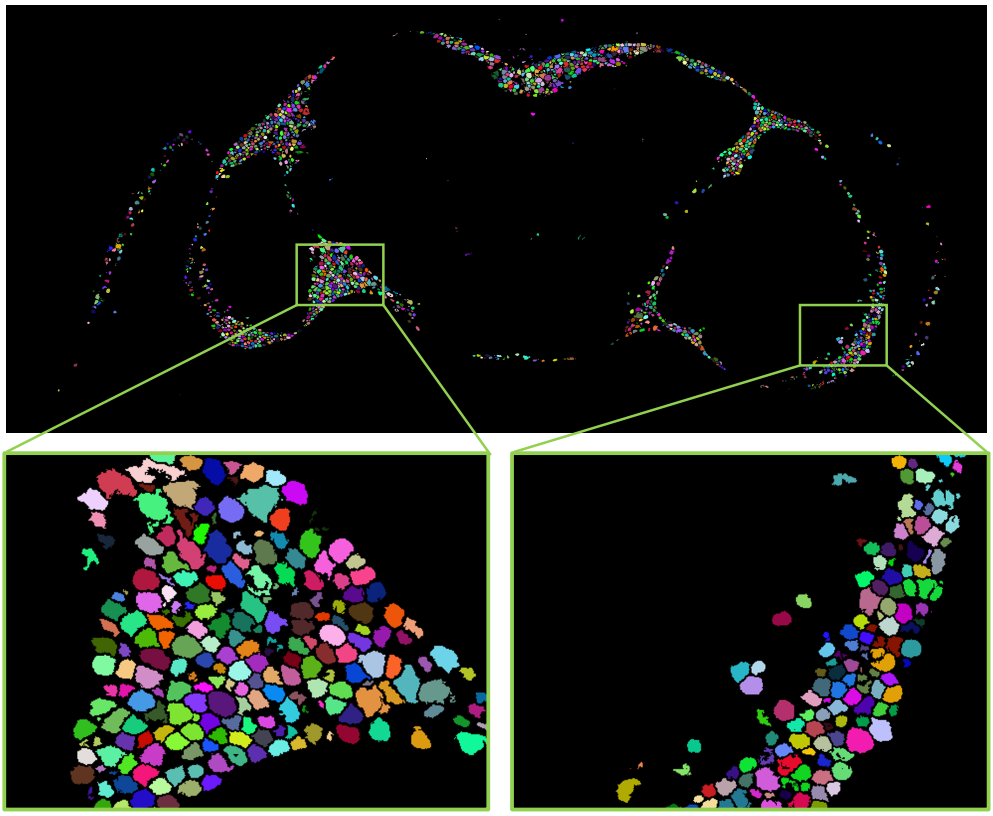

Figure 1. An image example of our provided image sequence with two areas zoomed in.

We provide a 2D image sequence to overall visualize the soma reconstruction results in a full adult fruit fly brain. Specifically, we downsample the full brain reconstruction result by the factors of 10, 10, 4 at the $x$-axis and $y$-axis, and visualize

*Corresponding author.

the result along the $z$-axis. Since our reconstruction is based on the physical resolution 16nm $\times$ 16nm $\times$ 40nm ($x \times y \times z$), the physical resolution of this overall full brain visualization is 160nm $\times$ 160nm $\times$ 160nm ($x \times y \times z$). An image example with two areas zoomed in is shown as Figure 1.

With this image sequence, we can obtain a panoramic view for the full brain reconstruction result. In addition, we also make a 4K video based on this image sequence. The 4K video link is: https://drive.google.com/drive/folders/1KT3f2gVVcGtXjklA-E7G31kkfS4rZKM3. We set the time duration of each image as 0.01 second, and stack the image sequence into a video chronological. Each color denotes a soma instance.

### 1.2. 3D Visualization

We visualize the full brain soma reconstruction result of three 3D sections in Figure 2. Also, we further zoom in a section with a higher factor to see the reconstructed somas in detail in Figure 3.

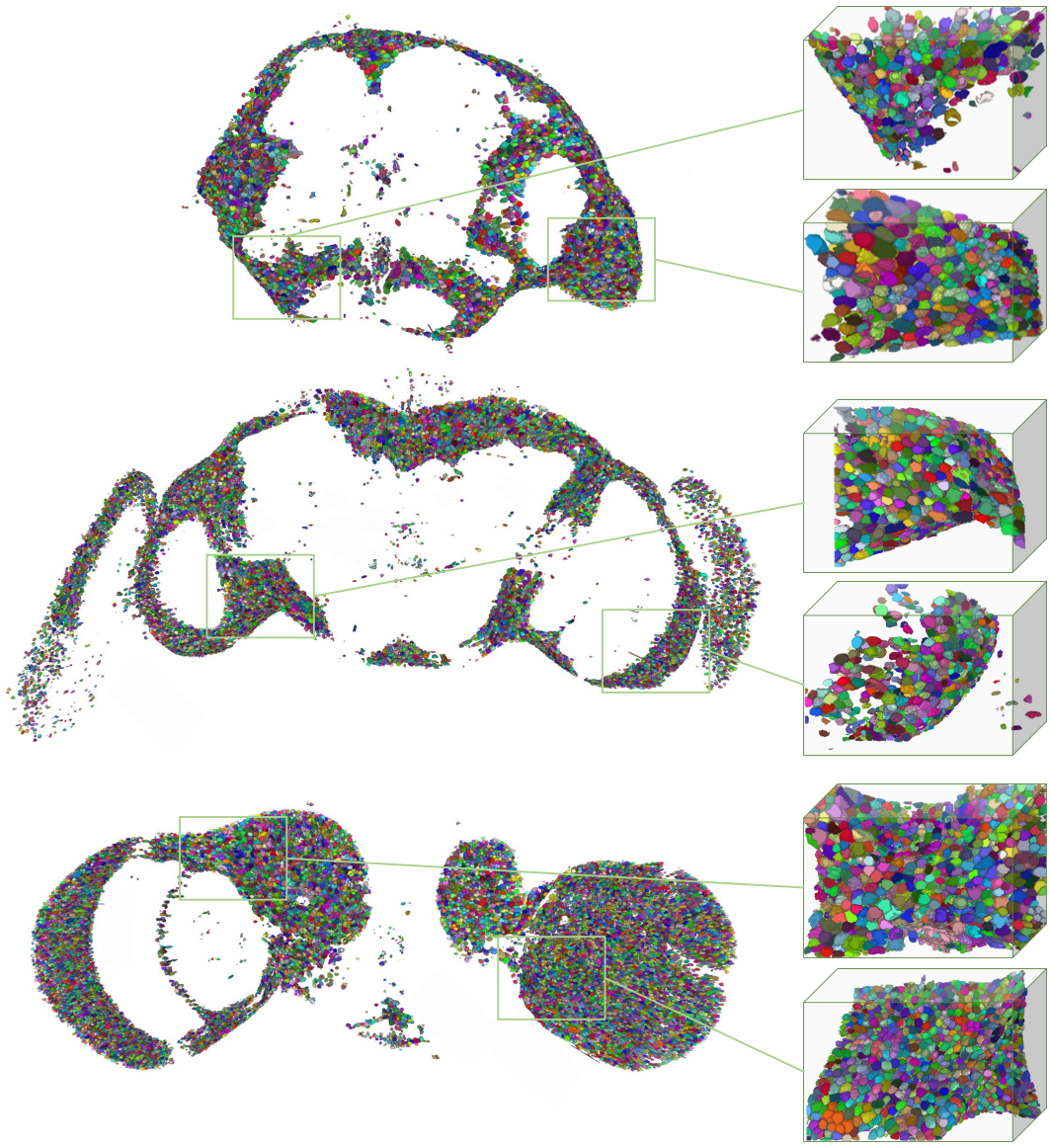

Figure 2. Visualization of our soma reconstruction for a full adult drosophila brain in three 3D sections. Each color instance corresponds to a reconstructed soma.

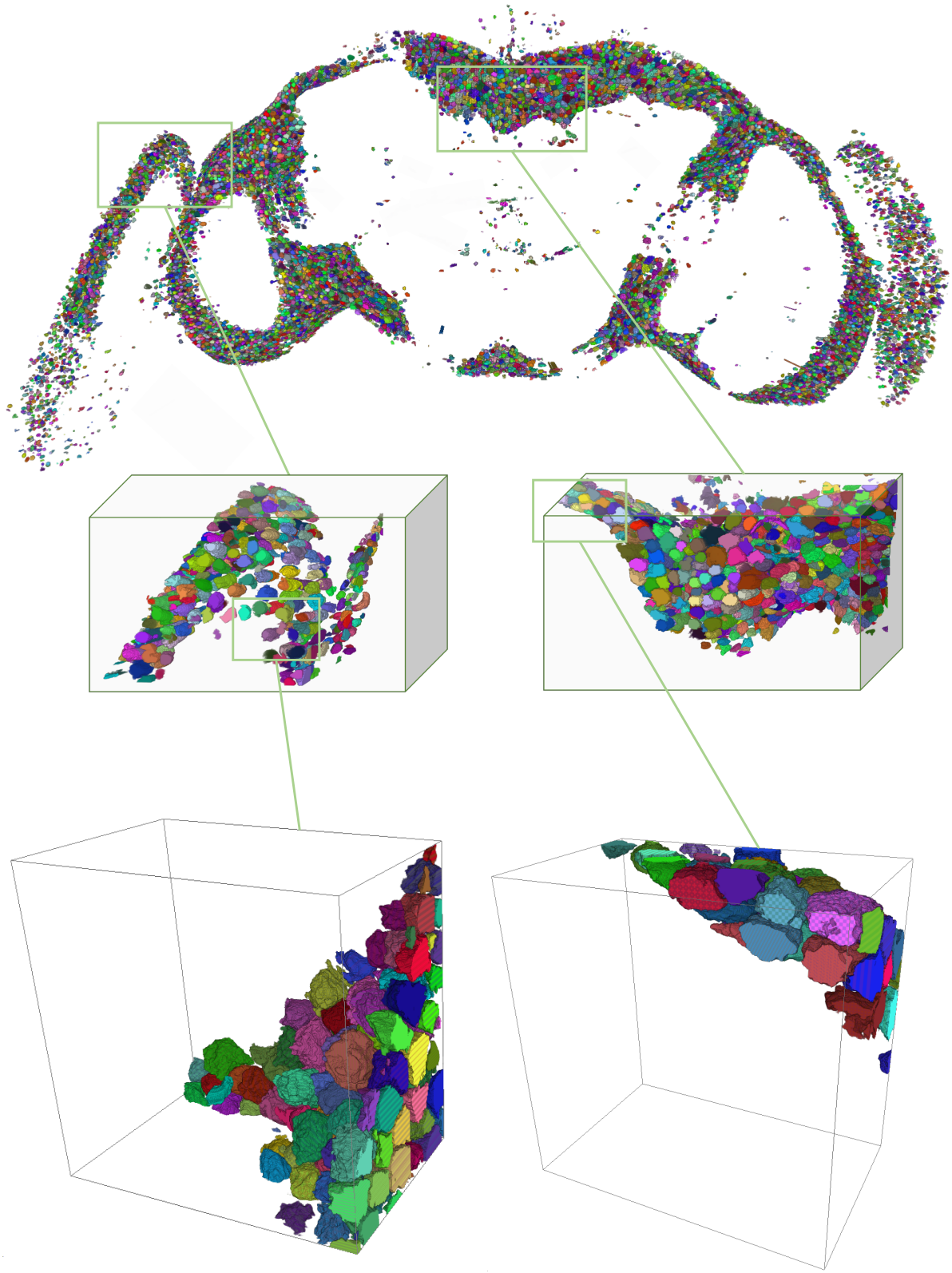

Figure 3. A detailed visualization example with two times zooming in.

## 2. Network Structures and Training Details

We illustrate the detailed structures of our deep networks in Figure 4 which are introduced in Section 3 in our main paper. Both the localization and segmentation networks are based on 3D U-Net, and they have the same structure except for the last convolution layer. Moreover, each convolution layer except for the last one is followed by a ReLU layer [1].

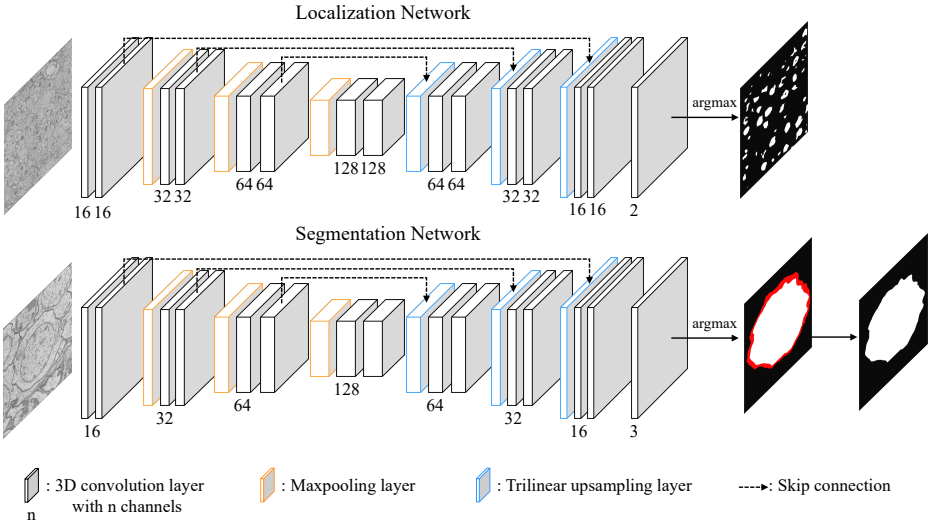


Figure 4. Illustration of the detailed structures of our deep networks. For simplicity and clarity, we illustrate the input and the output image as a slice of the 3D volume.

We train the localization network using a learning rate of $5 \times 10^{-4}$ and a batch size of 4 on four NVIDIA TITAN XP GPU for 500 epochs, and train the segmentation network using a learning rate of $1 \times 10^{-4}$ and a batch size of 1 on one NVIDIA TITAN XP GPU for 200 epochs. We optimize both of the networks by the Adam optimizer [2] with $\beta_1 = 0.9$ and $\beta_2 = 0.99$.

## 3. Parallelized Large-scale Data Processing Pipeline

The parallelized large-scale data processing pipeline is shown in Figure 5.

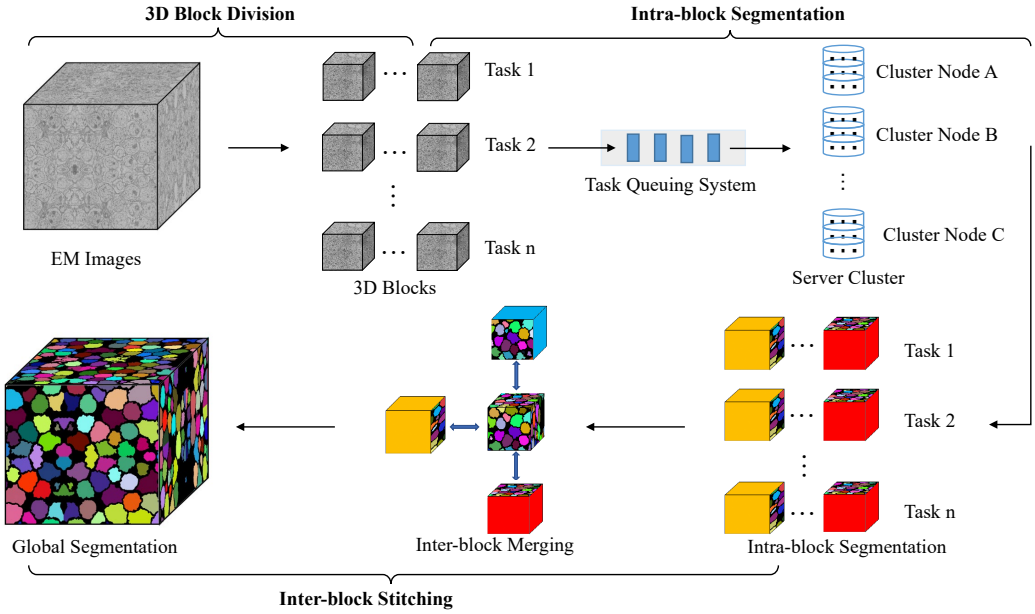


Figure 5. Overview of our parallelized large-scale data processing pipeline.

## 4. Details of Parallel Inter-block Stitching

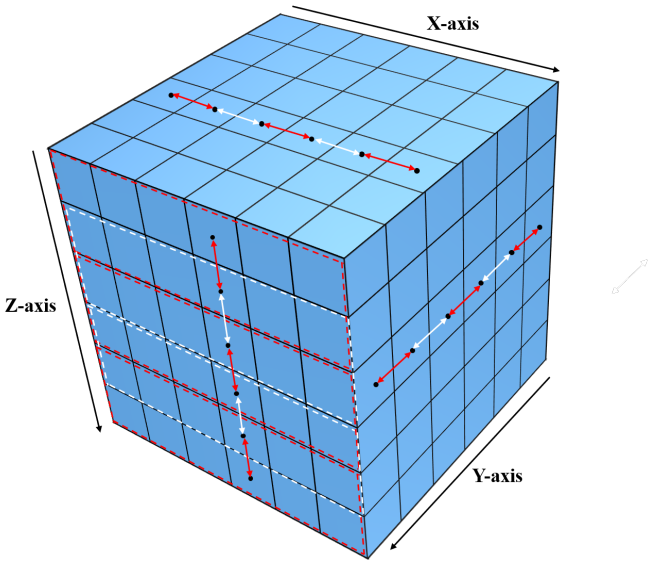


Figure 6. Illustration of our stitching algorithm. The red and white double-headed arrows indicate the pairwise blocks in the odd and even sequences, respectively.

This section elaborates our hierarchical block stitching algorithm, which is used for parallel inter-block stitching in the large-scale data processing pipeline. As shown in Figure 6, we divide the relations of these neighboring 3D blocks in the full brain into three types according to their relative positions. We stitch these blocks in different directions pair by pair. Specifically, for each direction, we divide neighboring blocks into two sequences, the odd and the even, which are indicated by the red and white double-headed arrows, respectively. We first stitch the blocks from the odd sequence, and then we stitch the blocks from the even sequence. The two sequences can be processed independently without interfering with each other to accelarate the pipeline. As an example of the volume consisting of $6 \times 6 \times 6$ blocks in Figure 6, we stitch 180 block pairs in each direction to obtain the final result. With sufficient computing resource (enough CPUs), we can first stitch 90 block pairs in the odd sequence in parallel, and then stitch the other 90 block pairs in the even sequence in parallel.

## 5. Visual comparison examples with the two baseline methods

Visual comparison examples with the two baseline methods is shown in Figure 7.

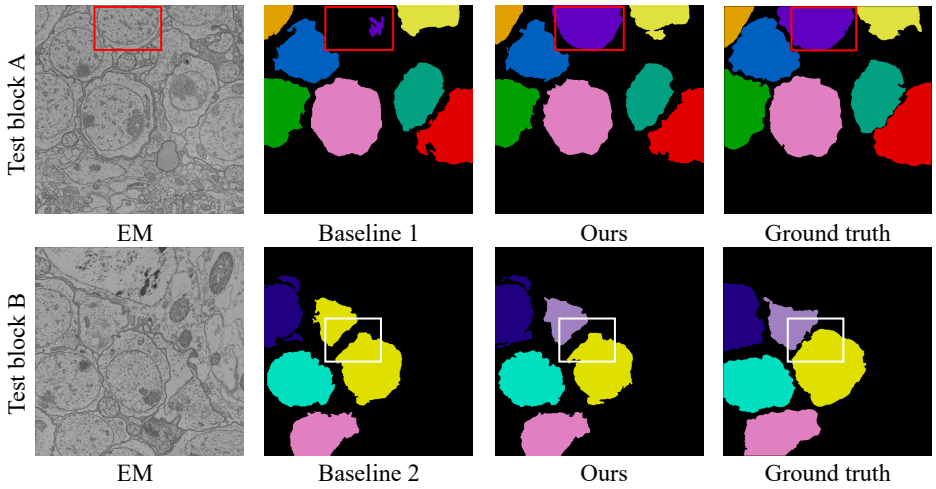


Figure 7. Visual comparison results of our method and two baseline methods.

## 6. Full Brain Partition

As introduced in Section 6 in our main paper, we separate the full brain into four types of cubic regions for statistics. As introduced in Section 2 in our main paper, the full brain data is stored by our defined 3D image blocks, and the image resolution of each block is $1836 \times 1836 \times 186$. The full brain data is stacked by $41 \times 22 \times 45$ blocks ($x \times y \times z$). Our sequence is listed in the $z-$axis order with 45 images, and the size of each image is $41 \times 22$. Each pixel in an image denotes a block. The pink, blue and purple color denotes that the block belongs to the central brain A, B and C region, respectively. The green color denotes that the block belongs to the optical lobes. We illustrate one image of this sequence as an example and the stacked image sequence for brain partition in Figure 8.

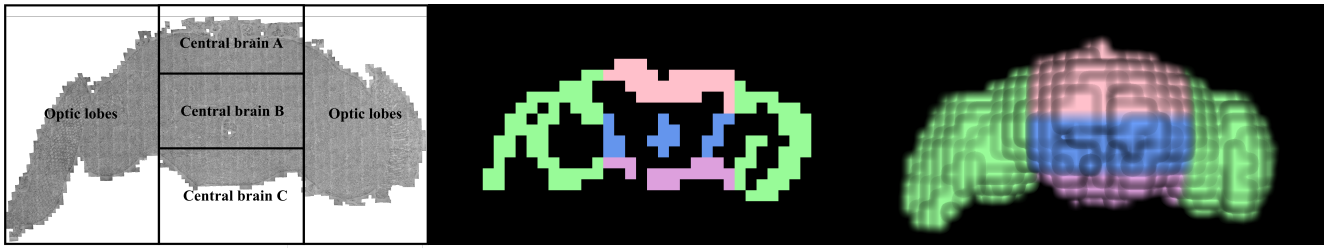


Figure 8. Illustration of our full brain partition. Left: our brain partition; Middle: an image example in the sequence for brain partition ; Right: the stacked 3D image sequence for brain partition.

## 7. Code and Dataset

Our project link is: https://github.com/liuxy1103/EMADS. We have released our trained localization and segmentation models along with their corresponding code, as well as our annotated dataset which has been organized and made available for public use.

## References

[1] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011. 4

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv*, 2014. 4