PATTER, Volume *2*

Supplemental Information

Toward human intervention-free clinical diagnosis of intracranial aneurysm via deep neural network

Zi-Hao Bo, Hui Qiao, Chong Tian, Yuchen Guo, Wuchao Li, Tiantian Liang, Dongxue Li, Dan Liao, Xianchun Zeng, Leilei Mei, Tianliang Shi, Bo Wu, Chao Huang, Lu Liu, Can Jin, Qiping Guo, Jun-Hai Yong, Feng Xu, Tijiang Zhang, Rongpin Wang, and Qionghai Dai

Table S1. | Ablation study on the internal test dataset.

Metrics		GLIA-Net w/o global positioning network	GLIA-Net w/o pyramid weighted loss	GLIA-Net
Voxel-wise	Precision↑	**49.8 (38.6-60.9)**	60.2 (50.2-70.2)	48.8 (44.5-53.0)
	Recall↑	36.4 (23.8-49.0)	60.5 (56.2-64.7)	**72.9 (66.9-78.9)**
	DSC↑	40.3 (29.5-51.1)	**60.0 (53.0-67.0)**	57.9 (56.4-59.5)
	95%HD↓	9.80 (7.85-11.8)	**7.91 (6.57-9.25)**	9.07 (7.84-10.3)
	AUC↑	90.0 (81.5-98.4)	92.1 (90.8-93.5)	**98.2 (97.6-98.8)**
	AP↑	35.7 (23.3-48.2)	57.3 (50.5-64.1)	**61.9 (59.4-64.4)**
Target-wise	Recall↑	39.7 (17.6-61.8)	44.0 (39.4-48.6)	**82.1 (78.2-86.0)**
	FPs per case↓	**1.10 (0.36-1.84)**	3.72 (1.38-6.06)	4.38 (2.91-5.85)

FPs per case is the number of false positive predictions per case. 95%HD is given in mm. Other Values are given in units of %.

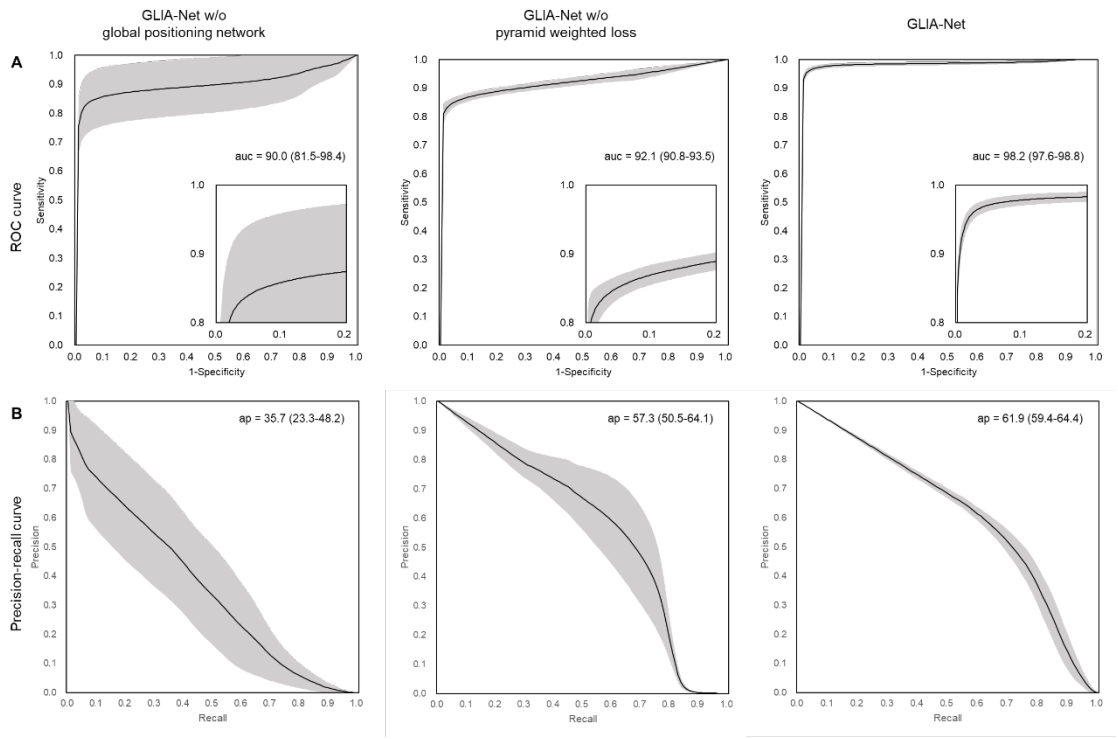

Figure S1. Segmentation performance of the ablation study on the internal test dataset. (A) ROC curve and (B) precision-recall curve of our GLIA-Net without global positioning network, our GLIA-Net without pyramid weighted loss, and our final GLIA-Net are shown. The AP and AUC values are given in “mean (95%CI)”. Most of the evaluation metrics get much worse without the support of our global positioning network.

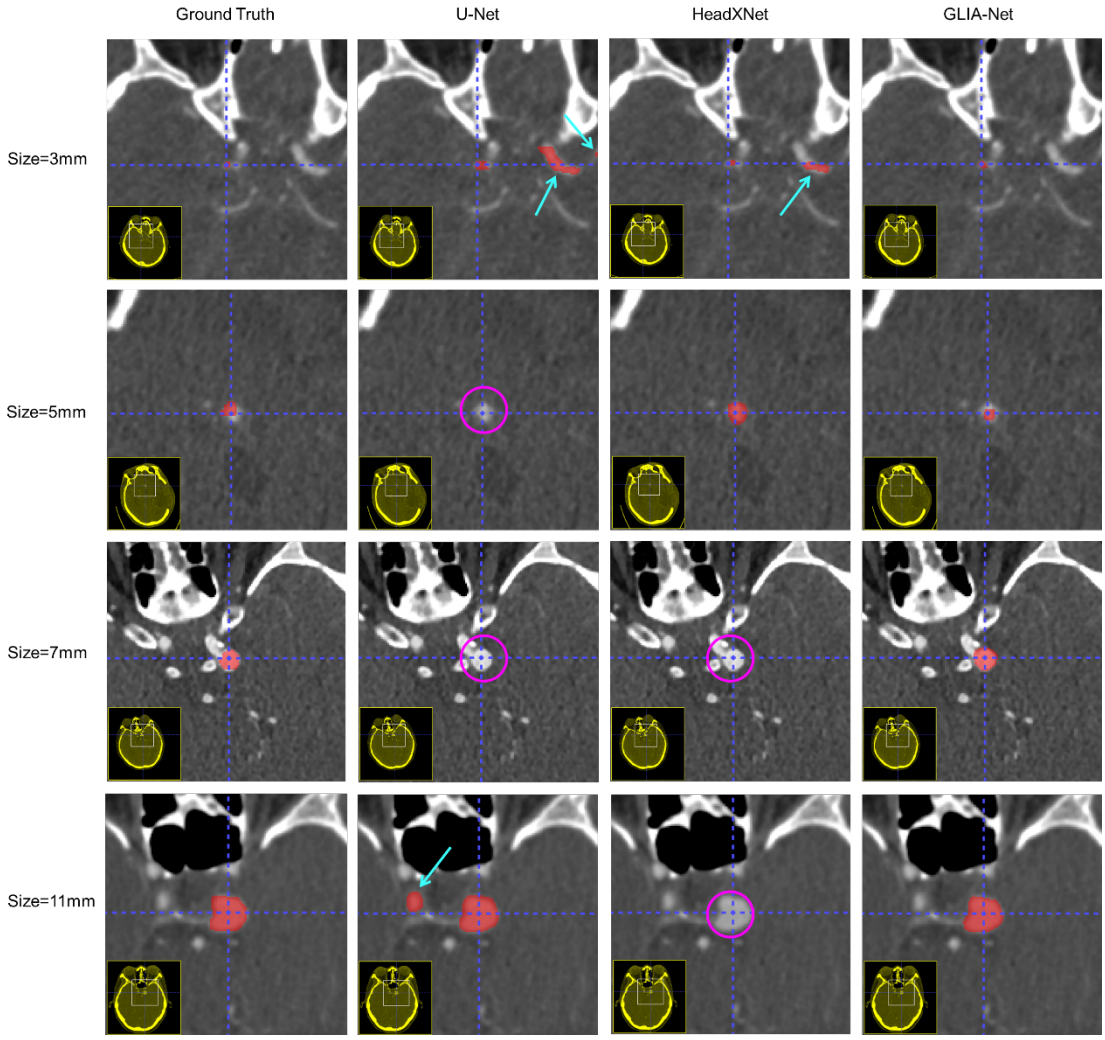

Figure S2. Segmentation results for 4 IAs of different sizes in the internal test dataset. The blue arrow points out the false positive predictions. The pink circle means the model fails to find the lesion area. The blue crosshair indicates the position of IAs.

Table S2. Clinical study performance of different institutions.

			Voxel-wise	Target-wise			Case-wise	
		Time↓	DSC↑ (95%CI)	Precision↑ (95%CI)	Recall↑ (95%CI)	Specificity↑ (95%CI)	Sensitivity↑ (95%CI)	ACC↑ (95%CI)
Institution α	Without Assist	147 (134-159)	70.9 (56.4-85.4)	90.9 (75.5-100)	83.3 (66.0-100)	83.3 (55.0-100)	**97.2 (92.5-100)**	93.8 (87.0-100)
	With assist	**123 (108-138)**	**74.8 (70.5-79.2)**	**93.2 (81.6-100)**	**95.8 (91.8-99.9)**	**100 (100-100)**	94.4 (89.0-99.9)	**95.8 (91.8-99.9)**
Institution β	Without assist	133 (115-151)	41.3 (23.8-58.8)	79.7 (73.2-86.3)	56.3 (49.5-63.0)	**100 (100-100)**	75.7 (62.8-88.6)	**83.3 (75.2-91.5)**
	With assist	**120 (105-134)**	**54.4 (44.7-64.1)**	**85.2 (74.2-96.3)**	**78.1 (70.3-86.0)**	85.4 (70.8-100)	**77.3 (69.9-84.7)**	79.2 (72.1-86.2)
Institution γ	Without assist	161 (144-179)	47.0 (32.4-61.5)	89.3 (71.1-100)	62.5 (37.8-87.2)	91.7 (77.5-100)	71.5 (55.9-87.2)	77.1 (62.5-91.7)
	With assist	**154 (136-172)**	**53.9 (34.9-72.9)**	**91.7 (77.5-100)**	**85.4 (76.5-94.3)**	**100 (100-100)**	**88.5 (80.8-96.3)**	**91.7 (85.9-97.4)**

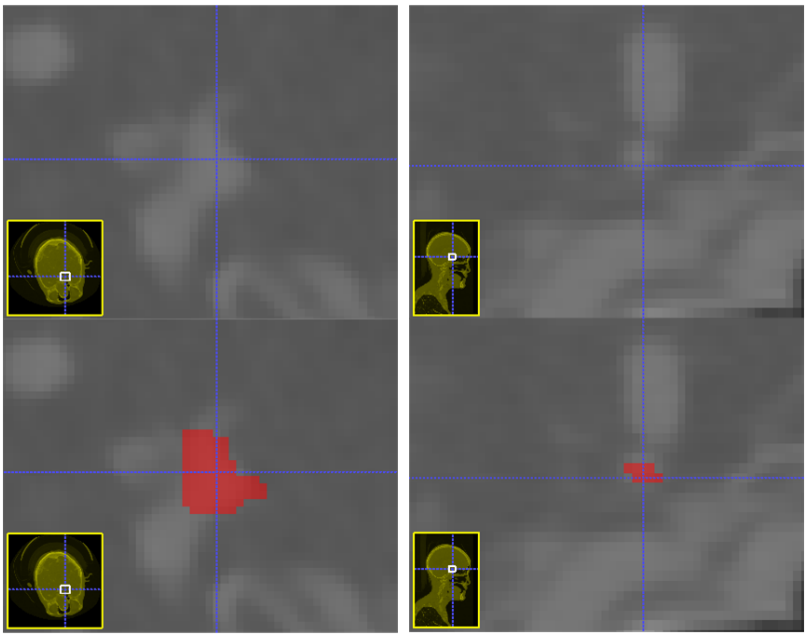

Figure S3. Annotation details for aneurysm segmentation. There are two label annotations achieved by radiologists in which the red mask is the annotation label. The boundary between aneurysms and brain tissue is very blurry, let alone that between aneurysms and their attached vascular, especially for the small case from the right figure. This is not a labeling error, but a result of the low resolution of CTA images and the definition of lesion regions. So we propose a pyramid weighted loss strategy to overcome this phenomenon.

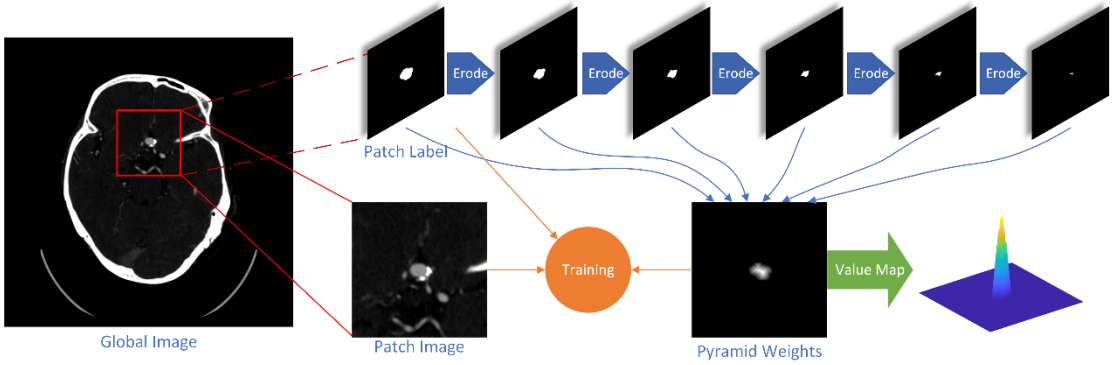

Figure S4. The pipeline of building the pyramid weights. The patch label map is eroded recursively and summed up to build the pyramid weights, which has high values in the target center, and low values on the edge. The highest and lowest weight values are fixed and the values in between are linearly scaled.

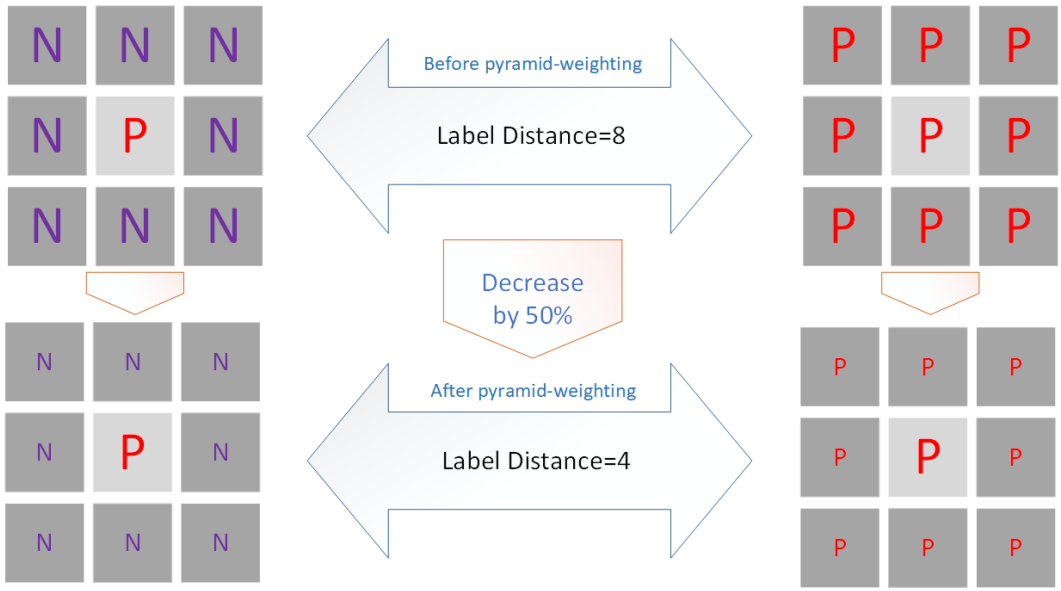

Figure S5. Label consistency before and after using the pyramid-weighted strategy. Each of the two image samples contains 9 pixels, which represents a small aneurysm with a ”certain area” in the center and “uncertain areas” on the edge. P and N represent positive and negative labels. Both the left and right samples may occur in the training set because of the different labeling standards. Before pyramid-weighting, all the pixels have the same loss weights of 1.0, leading a label distance between the two images to 8. After pyramid weighting, the weights of the neighboring pixels are decreased to 0.5, which decreases the label distance by 50%. With a low label consistency, the model may think these training samples to be noise and will not learn anything from them. This weighting strategy can increase the labeling consistency in the training set, thus enhancing our model's training procedure.

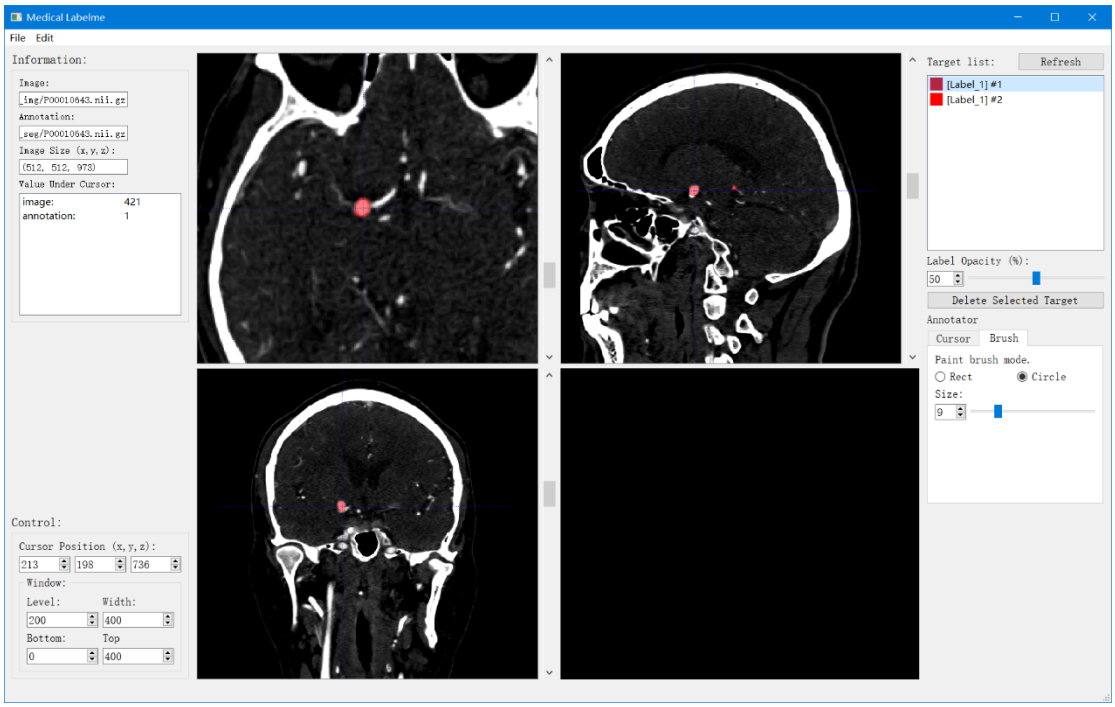

Figure S6. CTA viewing and annotation tool to assist radiologists in the IA diagnosis procedure. The software supports common scan viewing functions like the adjustment of the HU window and image statistics display. It also offers functions like IA annotation and target navigation, which is helpful to deal with large 3D images.

Supplemental Experimental Procedures

Implementation Details of GLIA-Net

Model Structure

Our segmentation model consists of a global positioning network and a local segmentation network whose inputs are the resized global CTA image and the local image patch with the same size of 96x96x96. They share some similar basic blocks in the building design. We use residual blocks[1] as the unit blocks in all architectures, which has 3 3D-convolution layers and a residual connection. All the convolution layers are followed by a group normalization layer[2] and a leaky relu activation layer[3]. We use bottleneck design for our residual block in which the kernel size for the first and last convolutions is 1 and that for the second is 3. Depending on the residual block, we design a universal encode block, which consists of a max-pooling layer if needs down-sampling and a residual block.

The global positioning network contains a global feature generator and a local feature generator. The global feature generator takes the resized global CTA image as input and has 5 encode blocks to build the global feature map whose output channels are 8, 16, 32, 64, and 128 separately, the 2nd and 3rd of which use down-sampling. Then the global feature map is cropped by a roi-pooling layer and reshaped to 6x6x6 whose bounding-box is the position of the current local patch. Finally, the local feature generator will be applied, which contains 2 encode blocks with 64 and 32 output channels. This local feature of the global positioning network for the current patch will be used to (1) compute the global positioning loss and (2) guide the local segmentation network through skip-connections. In (1), there are a 3D-convolution layer with group normalization and leaky relu, a global max-pooling layer, and a fully connected layer before the softmax computation of the global positioning loss. In (2), the output feature of the global positioning network is average-adaptive-pooled to different sizes in different scales of the local segmentation network and is activated by a sigmoid layer after a 3D-convolution layer.

The local segmentation network uses the encoder-decoder design with skip-connections between them, like U-net[4]. The encoder consists of 4 encode blocks with output channels 16, 32, 64, and 128. Except for the 1st encode block, all the other blocks contain down-sampling. So, the output feature map sizes of them are 96x96x96, 48x48x48, 24x24x24, and 12x12x12. The skip-connections is composed of the output feature maps of the first 3 encode blocks and are element-wise multiplied by specific adapted-sized local feature maps of the global positioning network. Then the enhanced skip-connections are conveyed to the decoder. The decoder of our local segmentation network consists of 3 decode blocks with output channels 64, 32, and 16 that can restore the feature map size to the original size of the input image step by step. The decode block takes the output of the former decode block (output of the encoder for the first decode block) and a skip-connection feature as input. It contains a 3D-transposed-convolution layer and a residual block. Then a final convolution layer whose output channel is 2 is applied to generate the final local segmentation probability map. The pipeline of building

pyramid weighted loss is shown in Figure S4.

Input Transformation

All the CTA images in our dataset are loaded as 3D images. The resolution of original images is Dx512x512, where D indicates the number of 2D images in each CTA scan. We clip the HU (Hounsfield unit scale) value of the images into 3 input channels before sending them to the network, each with a range of 0-100, 100-200, and 200-800. All the values in the 3 input channels are then normalized to 0-1. The clipping strategy is inspired by the diagnostic procedure of clinical practice.

The global image is resized from the original CTA image to 96x96x96 while keeping the same aspect ratio (with zero-padding) before fed into the global positioning network. The local image with a size of 96x96x96 is cropped from the original CTA image using a 3D tiling method. In clinical usage, we use a sliding window to generates local image patches from the global image with an overlap of 64 voxels, making sure that no possible target is lost. But when training, because of the severe label unbalance, we collect the training patches into a positive group and a negative group with the same numbers. For the positive group, we locate all lesion region centers and extract patches with a random deviation that is a maximum of 38 pixels from the center points, together with data augmentation of random flipping and rotation. For the negative group, we randomly select the patch centers from the global image.

Training Details

We train our end-to-end model using a deep learning framework PyTorch on RTX2080ti with 11GB memory. Adam optimizer is adopted with an initial learning rate of 0.0002 and the learning rate is decayed by 0.95 every 10000 steps. The training batch size is set to 3 and we train the model for about 200k steps. ω_{Global} and ω_{Local} in the total loss function of the training is set to 0.1 and 1.0. ω_{Dice} and ω_{Cross} in the local loss are 0.8 and 0.2. γ_{Dice} and γ_{Cross} in cross-entropy loss are both 0.3. In the cross-entropy loss of local loss, the pyramid weight for targets larger than 400 voxels is set to 3.0~20.0, and that for small targets is fixed to 11.5. The loss weight for negative voxels is set to 1.0.

Implementation Details of Other Methods

U-Net

Because the memory consumption in the 3D convolution network is heavy, the original U-net cannot be transferred to a 3D version while keeping the same parameter scale. We modify the original U-net to a similar parameter scale to our model. The encoder and decoder use the same structure as our model, except that it has no global feature to guide the skip-connections. This modified U-net uses batch-normalization and softmax cross-entropy loss to train.

HeadXNet

We follow the model structure described in the HeadXNet paper[5]. Because the specific model structure like the output channels for each block is not given, we design it to fit a similar parameter scale as ours. The HeadXNet model takes the same local images as ours and the training batch size is also 3. The output channels for each encoder block are set to 8, 16, 32, 64 and the output channel for the ASPP block is 64. There is only one max-pooling layer in the model as described in the paper, and we follow it. We also test the version that all the encoder blocks have a max-pooling layer, but the performance gets worse. We use softmax cross-entropy loss in the training period because the training using the combination of dice loss and softmax cross-entropy loss always leads to an unstable result that generates all-black outputs.

Supplemental References

1. He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778.

2. Wu, Y., and He, K. (2018). Group normalization. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 3-19.

3. Maas, A.L., Hannun, A.Y., and Ng, A.Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In Proc. icml, p. 3.

4. Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. Medical Image Computing and Computer-Assisted Intervention, Pt Iii *9351*, 234-241.

5. Park, A., Chute, C., Rajpurkar, P., Lou, J., Ball, R.L., Shpanskaya, K., Jabarkheel, R., Kim, L.H., McKenna, E., Tseng, J., et al. (2019). Deep Learning-Assisted Diagnosis of Cerebral Aneurysms Using the HeadXNet Model. JAMA Network Open *2*, e195600.