

## A. MODEL SUMMARY

### A1. Background on you/your team

- Competition Name: SenNet + HOA - Hacking the Human Vasculature in 3D
- Team Name: Igor Krashenyi
- Private Leaderboard Score: 0.712188
- Private Leaderboard Place: 4th
- Name: Igor Krashenyi
- Location: Kyiv, Ukraine
- Email: [reachingxforward@gmail.com](mailto:reachingxforward@gmail.com)

### A2. Background on you/your team

- My background is in biomedical imaging, but for the last 9 years, I've been working with different kinds of domains and problems as a research engineer with a computer vision specialization. I did a PhD in Alzheimer's disease diagnosis, and after that, I switched to the industry as a research engineer/scientist.
- My previous experience was useful in both training and inference pipeline development, and its optimization.
- As I mentioned previously, medical imaging is my research interest, and, it is hard to miss a competition in this field.
- The competition took me around 150 hours in total
- <https://www.linkedin.com/in/igor-krashenyi-38b89b98/>

### A3. Summary

The solution is also presented in the competition [discussion section](#).

The main feature of my solution is to combine 3d and 2d models trained with [BoundaryDOU](#) loss which improved the score significantly compared to standard Dice Loss and BCE. BoundaryDOU loss is focused on the boundaries specifically, which was helpful for the competition metric.

The training pipeline was built using PyTorch Lightning, segmentation\_models.pytorch (SMP), and monai libraries. The first simplified the scaling process and significantly reduced the amount

of boilerplate code while keeping flexibility. SMP and monai helped not to spend much time on model development and inference, focusing more on the pipeline itself. The training time for the 3d model (2 folds) is around 2.5-3 days using 2xRTX4090, while the 2d model (2 folds) is trained in 12-16 hours.

## A4. Features Selection / Engineering

One of the most important parts of my solution was normalization based on the percentiles of the data. As it was suggested in the discussion section percentile normalization works better compared to the standard min-max (organ-level and slice level). I used the default suggested values min=10 and max=99.8.

I also used external data from <https://human-organ-atlas.esrf.eu/>. This data was pseudo-labeled and used as additional during the training. The data and pseudo labels are available [here](#).

## A5. Training Method(s)

I've created a standard training pipeline for the segmentation models for both 2d and 3d images. The training was performed using distributed data-parallel mode to scale the training across 2 GPUs and enlarge the batch size.

For the inference, I used the regional ensemble of 2d and 3d models. My 3d model had issues with false positives outside the kidney volume. Based on my observation, I decided to ensemble only inside the kidney volume (ROI). To perform this I've created a bounding polygon based on the predictions of the 2d model slice by slice and applied the masking procedure on the top of 3d predictions.

## A6. Interesting findings

The most important trick was the early discovery of BoundaryDOU loss, which was unique and wasn't used in other top solutions (based on the top solutions' descriptions). Another finding was an interesting dependency in lowering the threshold caused a better score for both validation and leaderboard. I've noticed this behavior for the first time. I guess it's more related to sparsity in data annotation and not the competition metric.

## A7. Simple Features and Methods

Dropping the tta could slightly reduce performance while reducing the inference time significantly. Another option is to use a lighter backbone. Another option is to train instead of a 2 fold model just a single fold model with the training parameters from the 2-fold setup. My top score, which was not selected for the final submission, was trained in that manner and scored 0.774246 on the private leaderboard (just 2 2d models).

## A8. Model Execution Time

The training time for the 3d model (2 folds) is around 2.5-3 days using 2xRTX4090, while the 2d model (2 folds) is trained in 12-16 hours.

The inference time on 2xT4 GPU including the metric computation which is around 1.5 hours is ~9 hours total.

The simplified model could be trained in 16-20 hours. The inference time is around 4-5 hours including the metric calculation.

## A9. References

<https://arxiv.org/abs/2308.00220>

<https://github.com/sunfan-bvb/BoundaryDoULoss>

<https://human-organ-atlas.esrf.eu/>