Voxels Intersecting along Orthogonal Levels Attention U-Net (viola-Unet) to Segment Intracerebral Haemorrhage Using Computed Tomography Head Scans *

Qinghui Liu[1], Bradley J MacIntosh[1], Till Schellhorn[1], Karoline Skogen[1], Kyrre Eeg Emblem[1], and Atle Bjørnerud[1]

Oslo University Hospital (OUS), Rikshospitalet
0372 Oslo, Norway
{qiliu, bramac, uxscti, kaskog, kemblem, abjorner}@ous-hf.no
https://oslo-universitetssykehus.no/

Abstract. We implemented two distinct 3-dimensional deep learning neural networks and evaluate their ability to segment intracranial hemorrhage (ICH) seen on non-contrast computed tomography (CT). One model, referred to as "Voxels-Intersecting along Orthogonal Levels of Attention U-Net" (viola-Unet), has architecture elements that are amenable to the INSTANCE 2022 Data Challenge. A second comparison model was derived from the no-new U-Net (nnU-Net). Input images and ground truth segmentation maps were used to train the two networks separately in supervised manner; validation data were subsequently used for semi-supervised training. Model predictions were compared during 5-fold cross validation. The viola-Unet outperformed the comparison network on two out of four performance metrics (i.e., NSD and RVD). An ensemble model that combined viola-Unet and nnU-Net networks had the highest performance for DSC and HD. We demonstrate there were ICH segmentation performance benefits associated with a 3D U-Net efficiently incorporates spatially orthogonal features during the decoding branch of the U-Net. The code base, pretrained weights, and docker image of the viola-Unet AI tool will be publicly available at https://github.com/samleoqh/Viola-Unet.

Keywords: U-Net · intracranial hemorrhage · head computed tomography · deep learning · semi-supervised training.

1 Introduction

A spontaneous intracranial hemorrhage (ICH) is the second most common cause of stroke, following ischemic stroke, but with disproportionate high mortality and long-term disability. Bleeding might occur in the brain parenchyma or in the surrounding anatomical spaces. Subdural, epidural, and subarachnoid bleedings

* Supported and funding provided by Helse Sør-Øst Regional Health Authority.

are examples of ICH that occur in close proximity to layers near the skull and tend to be trauma-related. Parenchymal and subarachnoid ICH can lead to blood in the ventricles, which is a poor prognostic marker [1].

Non contrast computed tomography is effective at detecting hemorrhage. Robust radiological ICH estimates are desirable. For example, a subdural hematoma with a width ¿1 cm typically warrants neurosurgery [2]. The risk of poor outcomes scales with each mL increase in stroke hematoma volume [1]. Current clinical practise is to calculate hematoma volume "by-hand", following the ABC/2 method that uses the maximum hematoma diameter "A", the orthogonal in-place diameter "B", and "C" the number of slices where the hematoma is visible to produce a volume estimate. Although the ABC/2 can be done quickly (i.e. minutes) it is desirable to develop automated ICH segmentation methods [3] .

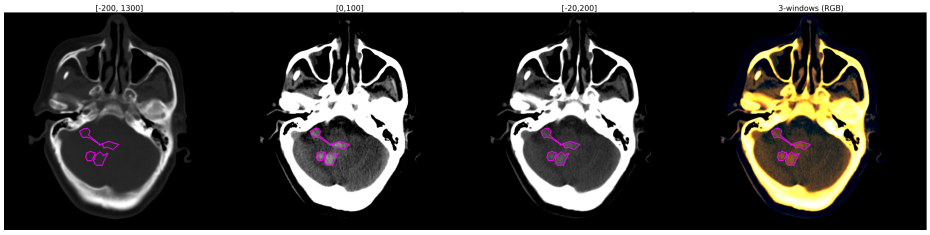

Fig. 1. An INSTANCE2022 example (case: 088) from the training dataset. Each images shows different Hounsfield Unit (HU) windowing levels to take advance of an RGB-style 3-window data input combination (from left to right: $[-200 \sim 1300]$, $[0 \sim 100]$, $[-20 \sim 200]$, and 3-window combination), with the ICH labeled segmentation regions highlighted by the pink edge lines.

Deep learning (DL) algorithms have recently received increasing attention in computer-aided automatic methods for medical data analysis. The state-of-the-art medical image segmentation models tend to rely on the popular U-Net [4] architecture, an encoder-decoder convolutional neural network (CNN) based approach with end-to-end training pipeline for pixel- or voxel-wise segmentation. Several U-Net-like models have tackled ICH segmentation using head CT scans [5,6,7,8,9] and these successes are mirrored in other brain imaging fields such as tumor segmentation of multi-modal MRI scans [10,11]. Isensee et al., in particular, used the nnU-Net framework [12] to present a winning model for the BraTS20 challenge [13], with a self-configuring method for various DL-based biomedical image segmentation tasks. Thus, we chose nnU-Net as the strong baseline model in the current work.

In this paper, we propose a novel solution for the ICH segmentation task of the INSTANCE 2022 challenge [14]. We demonstrate that there is a deep learning model that is fast, accurate, robust, and computational efficient in segmenting the ICH lesion. Section 2 presents our methodology in detail. Experimental procedure, evaluation, and test results are performed in Section 3. Finally, we draw

conclusions based on our participation in the INSTANCE 2022 challenge in the last Section 4.

2 Methods

We first describe the proposed Viola U-Net (viola-Unet) framework and its variants. We then present our design choices and provide detailed information about viola attention module for the ICH segmentation task.

2.1 Model architectures

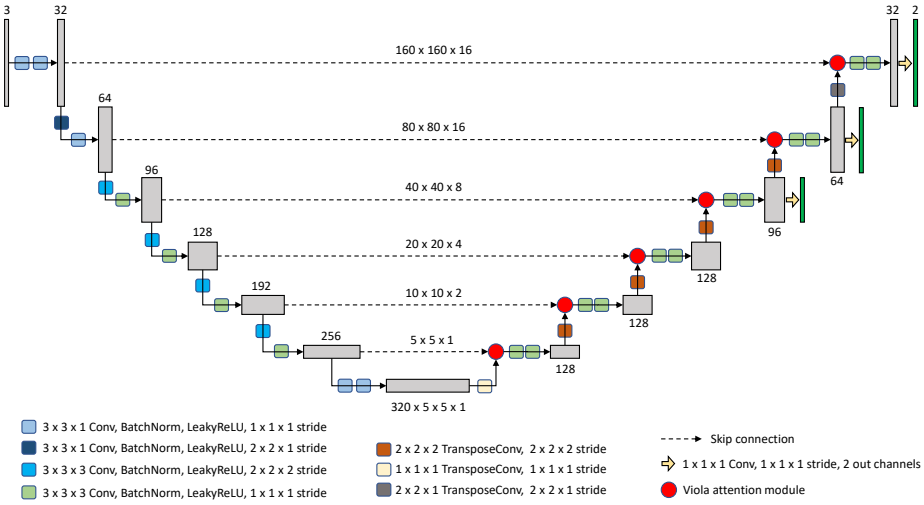

Fig. 2. The Viola U-Net (viola-Unet) architecture powered by the proposed Voxels Intersecting along Orthogonal Levels Attention (viola) module. Additional two output heads are only used for deep supervision [15] training.

Baseline nnU-Net: As a baseline, we used a self-configured U-Net architecture from the official open source nnU-Net framework [1]. The nnU-Net had a depth of 6. The number of channels at each encoder and decoder (symmetric) level were: 32, 64, 128, 256, 320 and 320. The input path size was $1 \times 320 \times 320 \times 16$ with 5 scales of deep supervision training outputs.

[1] https://github.com/MIC-DKFZ/nnUNet.

Viola U-Net: Our solution is called "viola-Unet" as it relies on Voxels in feature space that Intersect along Orthogonal Levels to provide an Attention U-Net, which is an asymmetric encoder-decoder architecture with 7-depth layers (shown in Figure 2). The number of channels at each encoder was 32, 64, 96, 128, 192, 256 and 320, while the channel-numbers at each corresponding decoder layer were 32, 64, 96, *128*, *128* and *128*. In addition, the input patch size was $3 \times 160 \times 160 \times 16$ with 2 extra scales of deep supervision outputs.

Architecture considerations: The viola-Unet is flexible and configurable, i.e. strides and kernel sizes at each layer, number of features in both encoder and decoder layers, symmetric or asymmetric, the number of deep supervision outputs. We can also incorporate other attention blocks such as gated attention [16]. The final submission used a larger version of viola-Unet configured as following:

```
ViolaUNet
(
  in_channels=3,
  out_channels=2,
  spatial_dims=3,
  kernel_size
    =[[3,3,1],[3,3,1],[3,3,1],[3,3,3],[3,3,3],[3,3,3],[3,3,3]],

  strides
    =[[1,1,1],[2,2,1],[2,2,1],[2,2,2],[2,2,2],[2,2,1],[1,1,1]],

  upsample_kernel_size
    =[[2,2,1],[2,2,1],[2,2,2],[2,2,2],[2,2,1],[1,1,1]],

  filters=(32,64,96,128,192,256,320),

  # can use different number of feature in each decoder layer
  dec_filters=(32,64,96,128,192,256),

  norm_name=("BATCH", {"affine":True}),
  act_name=("leakyrelu", {"inplace":True, "negative_slope"
    :0.01}),
  dropout=0.2,
  deep_supervision=True,
  deep_supr_num=4,
  res_block=True,
  trans_bias=True,
  viola_att=True,  # turn on or off viola attention
  gated_att=False, # turn on or off gated attention
  sum_deep_supr=False # can sum all deep-supervision outputs
    during inference
)
```

Listing 1.1. viola-Unet-l model configurations

Viola attention module: Squeeze-and-Excitation (SE) networks are able to recalibrate channel-wise feature responses by explicitly modelling interdependencies between channels on 2D feature planes[17]. The viola-Unet attention method is similar; Fig. 3 shows how the viola attention module incorporate features along orthogonal directions, which is an efficient way to incorporate through-plane features.

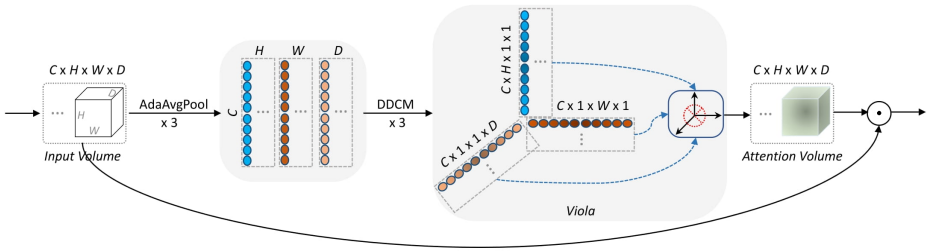

Fig. 3. The illustration of Voxels Intersecting along Orthogonal Levels Attention (Viola) pipeline. Here AdaAvgPool denotes adaptive average pooling, and DDCM denotes dense dilated convolutions' merging network [18].

Overall Viola module is composed of three key blocks, i.e., the adaptive average pooling (AdaAvgPool) module that squeezes the input feature volume (e.g., $\mathbf{X} \in \mathbb{R}^{C\times H\times\times W\times D}$, where C, H, W, and D represent channel, height, width, and depth for a given feature volume.) into three latent representation spaces (e.g., $\mathbf{X}_h \in \mathbb{R}^{C\times H}$, $\mathbf{X}_w \in \mathbb{R}^{C\times W}$, and $\mathbf{X}_d \in \mathbb{R}^{C\times D}$) along each axis of the input feature patch. The customized dense dilated convolutions merging (DDCM) networks fuses cross-channel and non-local contextual information on each orthogonal direction with adaptive kernel sizes (i.e., $k = [2 * (C//32) + 3, 1]$), dilated ratios (i.e., $dilation = [1, k, 2 * (k-1) + 1, 3 * (k-1) + 1]$) and strides (i.e, $strides = [(2,1),(2,1),(4,1),(4,1)]$).The Viola unit constructs the voxels intersecting along orthogonal level attention volume (i.e. $\mathbf{A}_{viola} \in \mathbb{R}^{C\times H\times\times W\times D}$) based on fused and reshaped cross-channel-direction latent representation spaces (i.e., $\mathbf{X}_h \in \mathbb{R}^{C\times H\times 1\times 1}, \mathbf{X}_w \in \mathbb{R}^{C\times 1\times W\times 1}$, and $\mathbf{X}_d \in \mathbb{R}^{C\times 1\times 1\times D}$), see footnote[2].

$$\tilde{\mathbf{X}}_h, \tilde{\mathbf{X}}_w, \tilde{\mathbf{X}}_d = \sigma\left(\mathbf{X}_h, \mathbf{X}_w, \mathbf{X}_d\right), \qquad \hat{\mathbf{X}}_h, \hat{\mathbf{X}}_w, \hat{\mathbf{X}}_d = \varphi\left(\mathbf{X}_h, \mathbf{X}_w, \mathbf{X}_d\right) , \tag{1}$$

$$\hat{\mathbf{X}}_h, \hat{\mathbf{X}}_w, \hat{\mathbf{X}}_d = 0.5 \cdot \left(\tilde{\mathbf{X}}_h + \hat{\mathbf{X}}_h\right), 0.5 \cdot \left(\tilde{\mathbf{X}}_w + \hat{\mathbf{X}}_w\right), 0.5 \cdot \left(\tilde{\mathbf{X}}_d + \hat{\mathbf{X}}_d\right) , \tag{2}$$

$$\tilde{\mathbf{A}}_{tt} = \left(\tilde{\mathbf{X}}_h \otimes \tilde{\mathbf{X}}_w\right) + \left(\tilde{\mathbf{X}}_w \otimes \tilde{\mathbf{X}}_d\right) + \left(\tilde{\mathbf{X}}_d \otimes \tilde{\mathbf{X}}_h\right) + \left(\tilde{\mathbf{X}}_h \otimes \tilde{\mathbf{X}}_w \otimes \tilde{\mathbf{X}}_d\right) , \tag{3}$$

$$\hat{\mathbf{A}}_{tt} = \mathrm{ReLU}\left(\hat{\mathbf{X}}_h + \hat{\mathbf{X}}_w + \hat{\mathbf{X}}_d\right) , \tag{4}$$

[2] Unless particularly specified, we use bold capital characters for matrices and tensors, lowercase and capital characters in italics for scalars and bold italics for vectors.

$$\mathbf{A}_{viola} = 0.1 \cdot (\tilde{\mathbf{A}}_{tt} + \hat{\mathbf{A}}_{tt}) + 0.3, \qquad \mathbf{A}_{viola} = \mathbf{A}_{viola} + \text{L2Norm}\left(\mathbf{A}_{viola}\right) \ , \quad (5)$$

$$\mathbf{X} = \mathbf{X} \odot \mathbf{A}_{viola} \ . \quad (6)$$

where σ denotes the Sigmoid activation function, φ denotes a combination function of group normalization [19] ($G = 2$ in this work) and Tanh non-linearity, $\otimes$ denotes the tensor product and $\odot$ denotes the element-wise multiplication.

3 Data, experiments and results

3.1 Dataset and evaluation metrics

The INSTANCE 2022 challenge dataset [20,14] consists of 200 non-contrast 3D head CT scans of clinically diagnosed patients with ICH of various types, such as subdural hemorrhage (SDH), epidural hemorrhage (EDH), intraventricular hemorrhage (IVH), intraparenchymal hemorrhage (IPH), and subarachnoid hemorrhage (SAH). N=100 of the publicly available cases were used for training; the remaining N=100 cases were held-out for the validation set (N=30 for the public leaderboard, and N=70 for the competitor rankings). The CT images had a matrix size of: $512 \times 512 \times N$, where N lies in $[20, 70]$. The average pixel spacing was around $0.45 \times 0.45 \times 5$ mm.

Model performance was evaluated by four measures: Dice Similarity Coefficient (DSC), Hausdorff distance (HD), Relative absolute Volume Difference (RVD), and the Normalized Surface Dice (NSD).

3.2 Implementation

Our code for this study were written in Python3 and PyTorch [21] with use of the open source Monai [3] library version 0.9.0. We adopted and modified Monai's network codes to implement the proposed models (both viola-Unet and modified nnU-Net).

3.3 Training details

Guided by our empirical results, we trained all networks with randomly sampled patches of fixed size ($3 \times 160 \times 160 \times 16$) as input and a batch size of 2. Each network was trained with 5-fold cross validation for up to 72,000 steps using stochastic gradient descent (SGD) and an optimizer with Nesterov momentum of 0.99. The initial learning rate was 7×10^{-3} with applying a cosine annealing scheduler [22] to reduce the learning rate over epochs. We used a linear warm-up learning rate during the first 1000 steps. A sliding window inference method [4] was applied to evaluate the model on the local validation set after every 200 training steps. We stored the checkpoint with the highest mean dice score on the validation set of the current fold during the training phase. Based on our

[3] https://monai.io/.
[4] MONAI sliding window implementation was used.

training observations to achieve fast and stable convergence for each network, we applied a combination loss function of the dice loss [23] and Focal loss [24] for all our experiments.

3.4 Data pre-processing and augmentations

CT image and ground truth labels were reoriented into "RAS" format (i.e., Right, Anterior and Superior), then resized to a standard spacing of $1 \times 1 \times 5$ mm^3 using trilinear interpolation for the image and nearest-neighbor interpolation for the label. Each CT image was windowed into three image intensity ranges (i.e., $[0, 100], [-20, 200]$, and $[-200, 1300]$ as shown in Fig. 1), and re-scaled to the range $[0, 1]$ by min-max normalization and then stacked as 3-channel (RGB-style) volumes to serve as inputs with the (C, H, W, D) shape where C-channels (e.g.3), H-height (e.g. 160), W-width (e.g. 160) and D-depth (e.g. 16), and then the 3-channel 3D volume was normalized on only non-zero values with calculated mean and std on each channel separately. In addition, the following data augmentation steps were taken during training phase:

- **Random Crop:** A fixed sized patch ($3 \times 160 \times 160 \times 16$) was randomly cropped with probability of 0.5. And the center was either a foreground or background voxel based on the Positive and Negative Ratio ($1 : 1$).
- **Random Zoom:** A random value was sampled uniformly from $(0.9, 1.2)$ with a probability of 0.15.
- **Gaussian Noise:** Random Gaussian noise with mean 0 and standard deviation of 0.01 was added to the input volume with a probability of 0.15.
- **Gaussian Smooth:** Gaussian smoothing with standard deviation of the Gaussian Kernel sampled uniformly from $(0.5, 1.15)$ was applied to the input volume with probability of 0.15.
- **Rotation:** With probability of 0.1, input volume was rotated by 90 degrees along either the x- or y-axis.
- **Random Shift:** Randomly shifted intensity for the entire volume was performed by uniformly sampled offset value from $[-0.1, 0.1]$ with a probability of 0.5.
- **Random Scale:** Randomly scale the intensity of the volume with a probability of 0.15 by a factor uniformly picked from $[-0.3, 0.3]$.
- **Flips:** Volumes were randomly flipped along each x, y, and z axis with a probability of 0.25 independently.
- **Random Contrast:** Randomly change volume intensity by a value sampled uniformly from $(0.78, 1.25)$ with a probability of 0.15.

3.5 Semi-supervised learning

In this work, we utilised self-training strategy to do semi-supervised fine-tune learning. The semi-supervised learning principle with self-training algorithms is to train a model iteratively by assigning pseudo-labels to the set of unlabeled training samples in conjunction with the labeled training set [25]. In practice,

we manually select the best prediction on each validation example from each submission as the pseudo-label and put them into our training set to fine-tune our models repeatedly.

During the self-training stage, we also optimized our hyperparameters gradually. First, based on our experimental observations, we optimized our model configurations and decided to use a larger version of viola-Unet, as shown in table 1. Second, to ensure a fair comparison, we reimplemented a comparable version of nnU-Net using the Monai library. Finally, we fine-tuned our models using minor updated parameters, such as 1) learning rate of 5×10^{-3} with a warm-up of $10,000$ steps, 2) windowing levels of [[0, 100], [-15, 200], [-100, 1300], 3) spacing of $[0.902, 0.902, 4.997]$, and 4) batch size of 3.

Table 1. Model configurations are provided for three networks with increasing architecture complexity (i.e., DS: number of deep-supervisions, Residual: used residual connections in the encoder layers, as per [26], Dec-filters: the number of decoder filters for the bottom two layers, and Z-strides: if down-size the z-slice for the last two encode layers). The number of parameters (Params) is provided in millions. The inference time (Inf-Time) was measured in seconds on a GeForce RTX 2080TI GPU with input patch size of $1 \times 3 \times 160 \times 160 \times 16$). Note that inference time is fast for each model. All three networks used the same number of features for encoder layers: $[32, 64, 96, 128, 192, 256, 320]$. The r in nnU-Net-r denotes that we re-implemented the nnU-Net after some optimization.

Models	DS	Residual	Dec-filters	Z-strides	Params (M)	Inf-Time (s)
viola-Unet-s	2	✗	$[128, 128]$	$[2, 1]$	12.77	0.051
nnU-Net-r	4	✓	$[192, 256]$	$[1, 1]$	22.01	0.026
viola-Unet-l	4	✓	$[192, 256]$	$[1, 1]$	22.12	0.052

3.6 Results

Table 2 shows the average Dice similarity coefficient (DSC) scores for each 5-folds with the nnU-Net baseline models and viola-Unet models, respectively. The viola-Unet outperforms the baseline nnU-Net by a significant margin (mean DSC +2.18%).

Table 3 shows online validation results with nnU-Net-base models and our viola-Unet-s models before applying semi-supervised learning methods. In terms of DSC and RVD, our models outperformed nnU-Net-base by about 1.2% and 3.9%, respectively, while underperformed by about 0.03% in terms of NSD.

In table 4, we show the top 10 ranking scores for INSTANCE 2022 online validation phase. Our semi-supervise trained viola-Unet-l models outperformed the comparison networks on two out of four performance metrics (i.e., NSD and RVD). An ensemble model that combined viola-Unet-l and re-implemented nnU-Net-r networks had the highest performance for DSC and HD.

Table 2. Average Dice Similarity Coefficient (DSC) for each of the 5-folds. Results for a base nnU-Net configuration (i.e. using the official nnU-Net framework and training pipeline without any modification) are shown along with a smaller-sized version of viola-Unet (s denotes small).

Model	**nnU-Net-base**	**viola-Unet-s**
Fold 0	0.7562	**0.7786**
Fold 1	0.7345	**0.7530**
Fold 2	0.7796	**0.7990**
Fold 3	0.7555	**0.8058**
Fold 4	**0.7746**	0.7730
Mean DSC	0.7601	**0.7819** (+2.18%)

Table 3. Online validation results without semi-supervised training with the nnU-Net-base and the viola-Unet-s models, i.e., DSC, HD (hausdorff distance), RVD (Relative absolute Volume Difference) and NSD (normalized surface dice).

Models	**DSC**	**HD**	**NSD**	**RVD**
nnU-Net-base	0.7251 ± 0.289	*null*	$\mathbf{0.5151} \pm 0.202$	0.2674 ± 0.281
viola-Unet-s	$\mathbf{0.7373} \pm 0.260$	$\mathbf{20.613} \pm 19.810$	0.5148 ± 0.187	$\mathbf{0.2284} \pm 0.194$

Table 4. Top 10 ranking scores for INSTANCE 2022 online validation phase [data extracted on 7-Aug-2022]. Note that 3 submissions provided by our team scored in the top-3. A larger version of the viola-Unet (l denotes large) was fine-tuned with semi-supervised training and achieved highest validation performance for NSD and RVD scores, while an ensemble of nnU-Net-r with viola-Unet-l was top for DSC and HD scores.

Models	**DSC**	**HD**	**NSD**	**RVD**
arren	0.7435 ± 0.236	31.616 ± 33.221	0.5201 ± 0.153	0.3580 ± 0.450
asanner	0.7456 ± 0.257	21.805 ± 21.735	0.5239 ± 0.175	1.1381 ± 0.112
dongyuDylan	0.7503 ± 0.237	29.072 ± 26.121	0.5280 ± 0.165	0.2301 ± 0.218
testliver	0.7537 ± 0.236	35.843 ± 28.453	0.5289 ± 0.165	0.2208 ± 0.206
L_Lawliet	0.7640 ± 0.213	34.323 ± 29.207	0.5381 ± 0.145	0.2044 ± 0.175
yangd05	0.7645 ± 0.237	25.725 ± 23.801	0.5403 ± 0.169	0.2322 ± 0.235
amrn	0.7821 ± 0.184	32.296 ± 30.039	0.5528 ± 0.127	0.2027 ± 0.182
nnU-Net-r (our)	0.7943 ± 0.174	22.799 ± 25.423	0.5673 ± 0.129	0.1952 ± 0.182
viola-Unet-l (our)	0.7951 ± 0.171	24.038 ± 29.236	$\mathbf{0.5693} \pm 0.125$	$\mathbf{0.1941} \pm 0.179$
Ensemble (our)	$\mathbf{0.7953} \pm 0.172$	$\mathbf{21.557} \pm 25.021$	0.5681 ± 0.125	0.1980 ± 0.180

The graph results for each of the 30 validation cases with the prediction estimates correspond to our best submission, i.e., the ensemble of nnU-Net-r and viola-Unet-l models, are shown in Fig. 4.

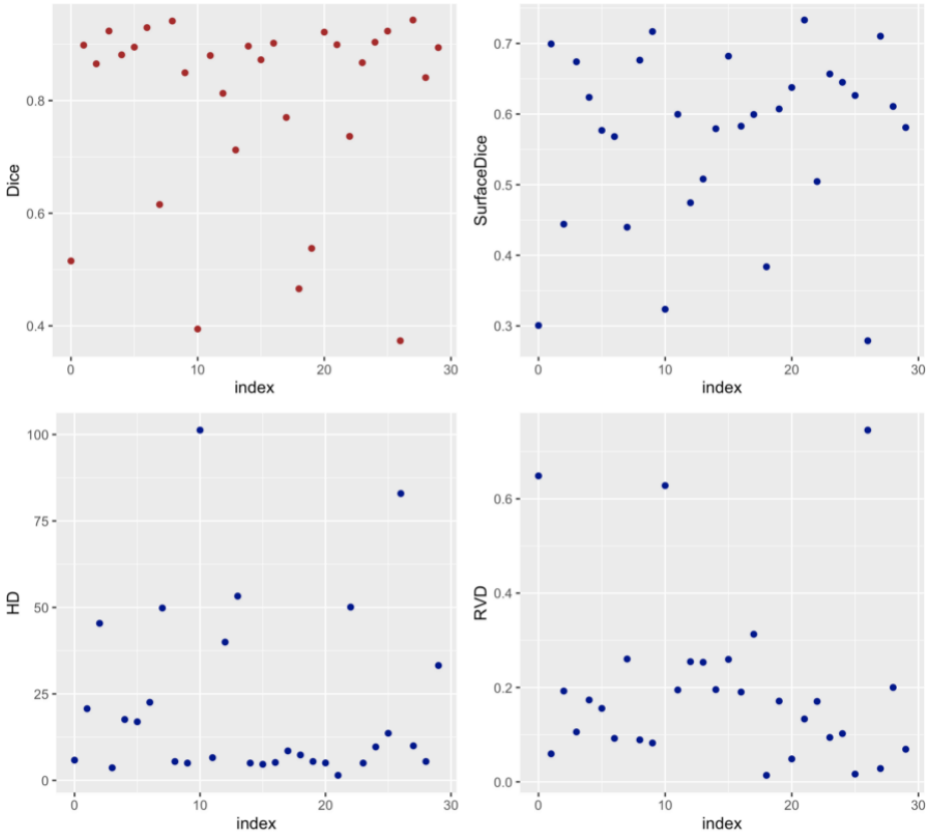

Fig. 4. The four graphs demonstrate the results for each of the 30 validation cases [denoted as index]. The prediction estimates correspond to one of the models that produced a high score on the leaderboard (i.e., an ensemble of nnU-Net-r and viola-Unet-l models).

4 Conclusions

We demonstrate that it is feasible to segment a range of ICH lesions on CT imaging by training a conventional nnU-Net and an architecture that we developed that is referred to as a viola U-Net deep learning model. The viola U-Net architecture is a novel conception. The flexible configurations were designed to

achieve high performance despite a limited training sample size. Notably, the model relied on image inputs that retained the 3-dimensional information from the CT images and orthogonal projections in the feature space were used to increase the between-plane information during the decoder layers of the U-Net. This design produced better segmentation results compared to the nnU-Net. The viola-Unet architecture did not incur additional computation costs and converged more rapidly than the nnU-Net despite comparable in the number of trainable parameters.

References

1. Rodrigues, M.A., E Samarasekera, N., Lerpiniere, C., Perry, L.A., Moullaali, T.J., J M Loan, J., Wardlaw, J.M., Al-Shahi Salman, R.: Association between Computed Tomographic Biomarkers of Cerebral Small Vessel Diseases and Long-Term Outcome after Spontaneous Intracerebral Hemorrhage. Ann Neurol 89(2), 266–279 (02 2021)
2. Zumkeller, M., Behrmann, R., Heissler, H.E., Dietz, H.: Computed tomographic criteria and survival rate for patients with acute subdural hematoma. Neurosurgery 39(4), 708–712 (Oct 1996)
3. Kothari, R.U., Brott, T., Broderick, J.P., Barsan, W.G., Sauerbeck, L.R., Zuccarello, M., Khoury, J.: The ABCs of measuring intracerebral hemorrhage volumes. Stroke 27(8), 1304–1305 (Aug 1996)
4. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
5. Arab, A., Chinda, B., Medvedev, G., Siu, W., Guo, H., Gu, T., Moreno, S., Hamarneh, G., Ester, M., Song, X.: A fast and fully-automated deep-learning approach for accurate hemorrhage segmentation and volume quantification in non-contrast whole-head CT. Scientific Reports 10(1), 1–12 (2020)
6. Hssayeni, M.D., Croock, M.S., Salman, A.D., Al-khafaji, H.F., Yahya, Z.A., Ghoraani, B.: Intracranial hemorrhage segmentation using a deep convolutional model. Data 5(1), 14 (2020)
7. Patel, A., Schreuder, F.H., Klijn, C.J., Prokop, M., Ginneken, B.v., Marquering, H.A., Roos, Y.B., Baharoglu, M., Meijer, F.J., Manniesing, R.: Intracerebral haemorrhage segmentation in non-contrast CT. Scientific reports 9(1), 1–11 (2019)
8. Sharrock, M.F., Mould, W.A., Ali, H., Hildreth, M., Awad, I.A., Hanley, D.F., Muschelli, J.: 3D deep neural network segmentation of intracerebral hemorrhage: Development and validation for clinical trials. Neuroinformatics 19(3), 403–415 (2021)
9. Yu, N., Yu, H., Li, H., Ma, N., Hu, C., Wang, J.: A Robust Deep Learning Segmentation Method for Hematoma Volumetric Detection in Intracerebral Hemorrhage. Stroke 53(1), 167–176 (01 2022)
10. Futrega, M., Milesi, A., Marcinkiewicz, M., Ribalta, P.: Optimized U-Net for Brain Tumor Segmentation. arXiv preprint arXiv:2110.03352 (2021)
11. Luu, H.M., Park, S.H.: Extending nn-UNet for brain tumor segmentation. arXiv preprint arXiv:2112.04653 (2021)
12. Isensee, F., Jaeger, P.F., Kohl, S.A., Petersen, J., Maier-Hein, K.H.: nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. Nature methods 18(2), 203–211 (2021)

13. Menze, B.H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., et al.: The multimodal brain tumor image segmentation benchmark (BRATS). IEEE transactions on medical imaging 34(10), 1993–2024 (2014)
14. Li, X., Wang, K., Liu, J., Wang, H., Xu, M., Liang, X.: The 2022 Intracranial Hemorrhage Segmentation Challenge on Non-Contrast head CT (NCCT) (Mar 2022), https://doi.org/10.5281/zenodo.6362221
15. Zhu, Q., Du, B., Turkbey, B., Choyke, P.L., Yan, P.: Deeply-supervised cnn for prostate segmentation. In: 2017 international joint conference on neural networks (IJCNN). pp. 178–184. IEEE (2017)
16. Oktay, O., Schlemper, J., Folgoc, L.L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N.Y., Kainz, B., et al.: Attention U-Net: Learning Where to Look For the Pancreas. arXiv preprint arXiv:1804.03999 (2018)
17. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
18. Liu, Q., Kampffmeyer, M., Jenssen, R., Salberg, A.B.: Dense Dilated Convolutions' Merging Network for Land Cover Classification. IEEE Transactions on Geoscience and Remote Sensing 58(9), 6309–6320 (2020)
19. Wu, Y., He, K.: Group normalization. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018)
20. Li, X., Luo, G., Wang, W., Wang, K., Gao, Y., Li, S.: Hematoma expansion context guided intracranial hemorrhage segmentation and uncertainty estimation. IEEE Journal of Biomedical and Health Informatics 26(3), 1140–1151 (2021)
21. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems 32 (2019)
22. Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts. ICLR (2017)
23. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 fourth international conference on 3D vision (3DV). pp. 565–571. IEEE (2016)
24. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
25. Amini, M.R., Feofanov, V., Pauletto, L., Devijver, E., Maximov, Y.: Self-training: A survey. arXiv preprint arXiv:2202.12040 (2022)
26. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)