3D Capsule Networks for Brain MRI Segmentation

Authors

Arman Avesta, MD,[1,2,3] Yongfeng Hui, BS, MPH,[2,3] Harlan Krumholz, MD, MS,[3,4] Sanjay Aneja, MD.[2,3,5]

[1] Department of Radiology and Biomedical Imaging, Yale School of Medicine, New Haven, CT 06510
[2] Department of Therapeutic Radiology, Yale School of Medicine, New Haven, CT 06510
[3] Center for Outcome Research and Evaluation, Yale School of Medicine, New Haven, CT 06510
[4] Division of Cardiovascular Medicine, Yale School of Medicine, New Haven, CT 06510
[5] Department of Statistics and Data Science, Yale University, New Haven, CT 06511

Abstract

INTRODUCTION: Segmenting brain structures around a tumor on brain images is important for radiotherapy and surgical planning. Current auto-segmentation methods often fail to segment brain anatomy when it is distorted by tumors.

OBJECTIVE: To develop and validate 3D capsule networks (CapsNets) that can segment brain structures with novel spatial features that were not represented in the training data.
Methods: We developed, trained, and tested 3D CapsNets using 3430 brain MRIs acquired in a multi-institutional study. We compared our CapsNets with U-Nets using multiple performance measures, including accuracy in segmenting various brain structures, segmenting brain structures with spatial features not represented in the training data, performance when the models are trained using limited data, memory requirements, and computation times.

RESULTS: 3D CapsNets can segment third ventricle, thalamus, and hippocampus with Dice scores of 94%, 94%, and 91%, respectively. 3D CapsNets outperform 3D U-Nets in segmenting brain structures that were not represented in the training data, with Dice scores more than 30% higher. 3D CapsNets are also remarkably smaller models compared to 3D U-Nets, with 93% fewer trainable parameters. This led to faster convergence of 3D CapsNets during training, making them faster to train compared to U-Nets. The two models were equally fast during testing.

CONCLUSION: 3D CapsNets can segment brain structures with high accuracy, outperform U-Nets in segmenting brain structures with features that were not represented during training, and are remarkably more efficient compared to U-Nets, achieving similar results while their size is one order of magnitude smaller.

Introduction

In patients with brain tumors undergoing radiotherapy, important brain structures such as thalamus and hippocampus should be avoided to prevent organ toxicity and preserve brain functions.[1] Therefore, it is important to segment brain structures on brain images in these patients.[1,2] Brain image segmentation is also important in surgical planning, image-guided interventions, and disease progress monitoring.[2] Manual segmentation is impractical because it requires radiologist-level expertise, is time-consuming, and is prone to inter- and intra-operator variability.[2] Currently-available software packages for image segmentation often fail when the brain anatomy is distorted by tumors, hemorrhages, and other space-occupying lesions.[2,3] These packages work by constructing distributions for the shape and location of each brain structure. When a space-occupying lesion distorts the brain anatomy, it changes the shape and location of brain structures to the extent that they fall out of their expected distributions, resulting in segmentation failure.[2] Therefore, the key to segmenting brain structures when their spatial features are changed by space-occupying lesions is to develop a method that can generalize to unseen spatial features.

Segmentation using deep learning methods, such as U-Nets, has two major drawbacks: 1) the unique anatomy of each patient might not be represented in the training data; and 2) when a tumor distorts the shape and location of brain structures, they fall out of the distributions represented in the training data. Notably, the full distribution of the anatomical variations and distortions caused by brain tumors cannot be represented in any training data, no matter how large the training data might be. While the convolutional operators in U-Nets generalize knowledge from one part of the image to other parts, they cannot generalize to segment a structure when it is rotated, squeezed, or otherwise distorted by space-occupying lesions. In summary, U-Nets cannot generalize to unseen spatial features. Data augmentation is commonly used to remedy this shortcoming, but the augmented data cannot represent the full distribution of anatomical variations and distortions either. As a result, efforts to segment brain structures (e.g. hippocampus) in the presence of space-occupying lesions have been largely unsuccessful.[4–6]

Capsule networks (CapsNets) are the potential solution to this problem.[4,5] The main idea behind CapsNets is that rotation, size, shear, and other spatial information about each structure on the image can be encoded and propagated in the network. If a structure rotates, changes in size, or undergoes other spatial changes, the capsule encoding that structure can still recognize it while encoding the changed spatial features. CapsNets can achieve this level of knowledge generalization without data augmentation. LaLonde et al developed 2D CapsNets that outperformed U-Nets in segmenting lungs on CT slices, and segmenting muscle and fat tissues on leg MRI slices, with Dice scores higher than 95%. They also showed that CapsNets outperform other segmentation methods, including U-Nets, when the model was fed with rotated inputs that were not represented during training. However, 2D CapsNets achieved less impressive results in segmenting heart and brain MRI slices, with Dice scores less than 70%. Subsequently, 2.5D CapsNets were introduced that analyzed five consecutive slices as input, resulting in Dice scores closer to 85%. Because most brain structures such as hippocampus and thalamus traverse more than five slices, 2D and 2.5D segmentation paradigms expectedly achieve suboptimal results. Therefore, there was a need to develop 3D CapsNets for volumetric segmentation.

In this study, we developed and validated 3D CapsNets for volumetric segmentation of brain MRIs.[7] The 3D CapsNets were trained using more than 3000 brain MRIs acquired in a multi-institutional study, and

were evaluated using a battery of performance measures including segmentation accuracy, out-of-distribution performance, performance when the models were trained using limited data, memory requirements, and computation times. For comparison, 3D U-Nets were also coded, trained, and tested.

Methods

The Data

We downloaded 3430 T1-weighted brain MRI volumes, belonging to 841 patients, from the Image and Data Archive (IDA).[8] These patients were enrolled in the multi-institutional Alzheimer's Disease Neuroimaging Initiative (ADNI) study.[9] Patients in this dataset range from mild cognitive impairment to Alzheimer's dementia. On average, each patient underwent four MRI acquisitions. Details of MRI acquisition parameters are provided in Supplemental Table 1.

We randomly split the patients (not the MRIs) into training, validation, and test sets. Therefore, all MRIs belonging to a patient ended up in the same set. We assigned 30 patients to the validation set (117 MRI volumes), 30 patients to the test set (114 MRI volumes), and the remaining 781 patients to the training set (3199 MRI volumes). The demographics for each of the training, validation, and test sets are provided in Table 1.

FreeSurfer-segmented brain images were also downloaded from IDA.[3,10,11] While Alzheimer's dementia is associated with degenerative changes in the brain, it does not cause brain anatomy distortions (as seen with tumors, hemorrhages, or other space-occupying lesions). FreeSurfer is shown to have expert-level segmentation accuracy for non-distorted brain images, including in patients with Alzheimer's dementia.[12] Therefore, FreeSurfer segmentations were used as the ground truth to train and test our models.

Pre-Processing

To make data loading faster, we converted the DICOMs of each brain MRI into a 3D NIfTi file.[13] We used FreeSurfer to correct for intensity inhomogeneities including B1-field variations.[3,14] We also used FreeSurfer to remove the skull, face, and neck, only leaving the brain.[15] The resultant 3D images were then cropped around the extracted brain.

To overcome memory limitations, we cropped 64×64×64-voxel boxes of the MRI volume that contained each segmentation target. The position of each box (e.g. for segmenting the right hippocampus) was determined by the first author (board-eligible radiologist with 9 years of experience in neuroimaging research) was fixed for all volumes.

3D CapsNet

We built on the 2D CapsNets introduced by LaLonde et al[7] to develop 3D CapsNets for volumetric segmentation. CapsNets are composed of three main ingredients: 1) capsules that each encode a structure together with the *pose* of that structure: the pose is an n-dimensional vector that learns to encode orientation,

size, curvature, location, and other spatial information about the structure; 2) a supervised learning paradigm that learns the transforms between the poses of the parts (e.g. head and tail of hippocampus) and the pose of the whole (e.g. the entire hippocampus); and 3) a clustering paradigm that detects a whole if the poses of all parts (after getting transformed) vote for matching poses of the whole. Therefore, any CapsNet architecture requires procedures for: 1) creation of the first capsules from the input; 2) learning transforms between the poses of parts and wholes; and 3) clustering the votes of the parts to detect wholes.

Figure 1.A shows the architecture of our 3D CapsNet. The first layer, Conv1, performs 16 convolutions (5×5×5) on the input volume to generate 16 feature volumes, which are reshaped into 16D vectors at each voxel. The 16D vector at each voxel provides the first pose that can learn to encode spatial information at that voxel. The next layer, PrimaryCaps2, has two capsule channels that learn two 16D-to-16D convolutional transforms (5×5×5) from the poses of the previous layer to the poses of the next layer. Likewise, the next *convolutional* capsule layers (green layers in Figure 1.A) learn m-to-n-dimensional transforms between the poses of the previous layer and the poses of the next layer. The number of transforms at each layer matches the number of capsule channels (shown by stacks of capsules in Figure 1.A). Our CapsNet has downsampling and upsampling limbs. The downsampling limb learns *what* structure is present at each voxel, and the skip connections from downsampling to upsampling limbs preserve *where* each structure is on the image. Downsampling is done using 5×5×5 convolutional transforms with stride = 2. The poses in the deeper parts of the downsampling limb have more pose components (up to 64) to be able to encode more complex spatial information. Additionally, layers in the deeper parts of the model contain more capsule channels (up to 8) to be able to encode more structures at each voxel, since each voxel in these layers corresponds to multiple voxels in the input that can each represent a separate structure. Upsampling is done using 4×4×4 transposed convolutional transforms with stride = 2 (turquoise layers on Figure 1A). The final layer, FinalCaps13, contains one capsule channel that learns to activate capsules within the segmentation target and deactivate them outside the target. Activation of a capsule is determined by the length of its pose vector, which is a number between 0 and 1. Further details about the activation of capsules are provided in the supplemental material.

To find clusters of the agreeing votes of the parts, we used the inner products between the poses of the parts and the aggregate pose of the whole.[5] We used Dice loss to train our models and to evaluate segmentation accuracy.[16] Further details about the clustering method, loss function calculation, and activation of each capsule are provided in the supplemental material.

3D U-Net

Figure 1.B shows the architecture of our 3D U-Net. The input image undergoes 64 convolutions (3×3×3) to generate 64 feature volumes. These volumes then undergo batch normalization and ReLU activation. Similar operations are carried out once more before downsampling using max-pooling (2×2×2). The downsampling limb includes four downsampling units, each composed of a max-pooling layer followed by two convolutional layers. The layers in the deeper parts of the downsampling limb have more channels (up to 1024). The upsampling limb includes four upsampling units, each composed of an upsampling layer, concatenation with the skip connection, and two convolutional layers. Upsampling is done using 2×2×2 transposed convolutions with stride = 2. The final layer carries out a 1×1×1 convolution to aggregate all 64 channels, followed by soft thresholding using the sigmoid function. The model learns to output a number

close to 1 for each voxel inside the segmentation target, and a number close to 0 for each voxel outside the target.

Training

We used Dice loss for training our models. Adam optimizer was used with the following hyperparameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. Training was done using 50 epochs, each consisting of all 3199 brain MRIs in the training set, and with the batch size of four. Because of the large epoch size, we split each epoch into mini-epochs that each comprised 30 batches (120 MRIs). After each mini-epoch during training, the Dice loss was computed for the validation set (117 MRIs).

We used dynamic paradigms for learning rate scheduling and for selecting the best models. The initial learning rate was set at 0.002. The validation set Dice loss was monitored after each mini-epoch, and if it did not decrease over 10 consecutive mini-epochs, the learning rate was decreased by half. The minimum learning rate was set at 0.0001. The model with the lowest Dice loss *over the validation set* was selected as the best model and was used for testing.

Performance Evaluations

We compared CapsNets and U-Nets using the following performance measures:

1) Segmentation accuracy for three brain structures: third ventricle, thalamus, and hippocampus. These three structures represent easy, medium, and difficult structures for segmentation. Third ventricle is an easy structure because it is a cerebrospinal fluid (CSF) filled cavity with clear boundaries. Thalamus is a medium-difficulty structure because it is abutted by CSF on one side and brain parenchyma on the other side. Hippocampus is a difficult structure because it has a complex shape and is abutted by multiple brain structures with indistinct borders. Segmentation accuracies were quantified using Dice scores.

2) Out-of-distribution segmentation accuracy: to evaluate the performance of our models in segmenting images that were not represented during training, we trained our models to segment right thalamus and hippocampus. Then, we compared the performance of our models in segmenting the contralateral left thalamus and hippocampus. Notably, we did not use data augmentation during training.

3) Training using small datasets: in addition to training our models on the full training set (3199 MRIs), we also trained them on smaller training sets comprised of 600, 240 and 120 MRI volumes. These smaller training sets were randomly selected. Then, the compared the performance of our models on the test set.

4) Model size and computation times: the number of trainable parameters, the model size in megabytes, and the computation times were compared between CapsNets and U-Nets, both during training and testing. Computation times were calculated per example.

For all experiments, the mean segmentation accuracies over the test set were compared between CapsNets and U-Nets using paired-samples t-tests. The mean Dice scores together with their 95% confidence intervals

were also tabulated for the two models and the three brain structures that were segmented. While our main measure of segmentation accuracy was Dice score, we also tested our final models using additional measures of segmentation accuracy. Details about these additional measures and the performance of our models using these measures are provided in the supplemental material.

Implementation

We implemented our models in Python (version 3.9). The models were coded in PyTorch (version 1.9). The statistical analyses were conducted using the SciPy package within Python. Pre-processing, statistical analyses, and visualization of the results were done on a local computer (iMac, 3.1 GHz Intel Core i7 processor, 16 GB DDR3 RAM, 1GB NVIDIA graphics). Training and testing of the models were run on AWS p2xlarge instances (4 vCPUs, 61 GB RAM, 12 GB GPU).

Results

As mentioned in Methods, this study included 3430 brain MRIs belonging to 841 patients that were enrolled in the multi-institutional Alzheimer's Disease Neuroimaging Initiative (ADNI) study.[9] Patient demographics are provided in Table 1.

The accuracy of 3D CapsNets in segmenting various brain structures is above 90% and is within 1.5% of to the accuracy of U-Nets. Figure 1 shows the segmentation of various brain structures by both models in a patient. Table 2 compares the segmentation accuracy of the two models, measured by Dice scores. Supplemental Table 2 compares additional measures of segmentation accuracy between the two models.

The 3D CapsNets achieved better out-of-distribution segmentation accuracy compared to 3D U-Nets. When both models were trained to segment right-sided brain structures and tested on contralateral left left-sided brain structures, 3D CapsNets significantly outperformed 3D U-Nets with Dice scores more than 30% higher. Figure 3 illustrates segmentation of the contralateral left thalamus and hippocampus by both models in a patient. Table 2 compares out-of-distribution segmentation accuracy between the two models.

The 3D CapsNets and 3D U-Nets achieved comparable segmentation accuracy when trained on smaller datasets. When the size of the training set was decreased from 3199 to 600 brain MRIs, both CapsNet and U-Net were minimally affected. Further decrease in the size of the training set down to 120 brain MRIs caused a decrease in the accuracy of both models down to 85%. Figure 4 shows the performance of both models when trained on smaller datasets.

Our 3D CapsNets are remarkably smaller models compared to 3D U-Nets. The 3D CapsNet has 7.4 million trainable parameters, while the corresponding 3D U-Net has 90.3 million trainable parameters. In addition, the 3D CapsNet has fewer layers and fewer steps of image propagation in forward and backward passes, leading to a smaller cumulative size of the feature volumes in the entire model. The 3D CapsNet and 3D U-Net respectively hold 228 and 1364 megabytes of cumulative feature volumes in the entire model. Figure 5.A compares the size of 3D CapsNet and 3D U-Net models.

The 3D CapsNets train slightly faster compared to U-Nets. When we compared the training time between the two models (on an AWS p2xlarge instance with 12GB of GPU memory), our 3D CapsNets and 3D U-Nets respectively took about 1.5 and 2 seconds per example per epoch to train. The two models are equally fast during testing, taking 0.9 seconds to segment the MRI volume. Figure 5.B compares the training and testing times between the two models.

Discussion

In this study, we developed and validated 3D CapsNets for volumetric segmentation of brain MRIs. We also coded, trained, and tested 3D U-Nets as the main competitor. We compared the two models using a battery of performance measures. Our results showed that 3D CapsNets have high segmentation accuracy for segmenting various brain structures with Dice scores above 90%. While 3D CapsNets are one order of magnitude smaller than U-Nets, their segmentation accuracy is within 1.5% of U-Nets.

In out-of-distribution segmentation, 3D CapsNets outperformed U-Nets with Dice scores more than 30% higher. This is expected, given that the main idea behind the CapsNets is generalization to novel viewpoints and spatial features. In object recognition, 2D CapsNets were already shown to ourperform convolutional neural networks (CNNs) when the objects are imaged from viewpoints that were not represented during training.[4] In 2D image segmentation, 2D CapsNets are shown to outperform 2D U-Nets in segmenting rotated images.[7] Our study extends the literature by showing that 3D CapsNets outperform 3D U-Nets in segmenting contralateral brain structures that were not represented during training. Notably, we did not use data augmentation during training. Given our results, we propose that 3D CapsNets can segmenting brain structures with out-of-distribution spatial features, such as scenarios in which the brain anatomy is distorted by space-occupying lesions. We propose that a model with out-of-distribution segmentation capabilities will be able to segment such deformed brain structures.

CapsNets are slightly faster to train compared to U-Nets. While clustering of vote vectors between capsule layers slows down CapsNets, they converge faster than U-Nets given that they have 93% fewer parameters to train. The net effect of these opposing factors leads to slightly faster training of CapsNets. The two models are equally fast during testing. Given that the forward pass through the fixed, trained parameters during testing is much faster compared to the forward *and* backward passes during training, the larger size of the U-Net does not slow it down as much during testing as it does during training. At the same time, clustering between the capsule layers slows down the CapsNet during testing same as training. As a result, the two models end up being equally fast during testing.

To develop 3D CapsNets and make them work for volumetric brain MRI segmentation, we explored a large space of design questions, hyperparameters, loss functions, and implementation details to find optimal solutions. We used the validation set to explore these questions and tested our final models on the test set only once. While our model performs well for volumetric segmentation of T1-weighted brain MRIs, we did not evaluate its performance for segmentation of other organs or other imaging modalities. We assume that our model would need modifications to perform well on other segmentation tasks or on brain MRIs that are pre-processed differently. We have described the experiments that helped us find optimal solutions for our

design questions in the supplemental material, and we welcome further research to generalize our 3D CapsNet to other imaging modalities and segmentation tasks.

This study extends the literature by developing and validating 3D CapsNets for volumetric segmentation. We also showed that 3D CapsNets are promising for out-of-distribution segmentation of brain structures that were not represented during training, which will open the door to use them in segmenting distorted brain anatomy in patients with brain tumors, hemorrhages, and other space-occupying lesions. Our future work will focus on using 3D CapsNets to segment the distorted anatomy in patients with brain tumors, with the aim of improving radiotherapy, surgical planning, and disease progress monitoring in these patients.

Conclusion

In this study, we developed and validated 3D Capsule Networks and used them for volumetric segmentation of brain MR images. While this model is one order of magnitude smaller than the equivalent 3D U-Net, it achieves comparable performance in segmenting various brain structures. Additionally, 3D CapsNets can segment brain images that are not represented during training, outperforming 3D U-Nets in out-of-distribution segmentation.

References

1. Feng CH, Cornell M, Moore KL, et al. Automated contouring and planning pipeline for hippocampal-avoidant whole-brain radiotherapy. *Radiat Oncol Lond Engl* 2020;15:251.

2. Despotović I, Goossens B, Philips W. MRI Segmentation of the Human Brain: Challenges, Methods, and Applications. *Comput Math Methods Med* 2015;2015:e450341.

3. Fischl B, Salat DH, Busa E, et al. Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain. *Neuron* 2002;33:341–55.

4. Hinton GE, Sabour S, Frosst N. Matrix capsules with EM routing. In: *International Conference on Learning Representations 2018*.

5. Sabour S, Frosst N, Hinton GE. Dynamic routing between capsules. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Red Hook, NY, USA: Curran Associates Inc.; 2017:3859–69.

6. Afshar P, Mohammadi A, Plataniotis KN. Brain Tumor Type Classification via Capsule Networks. In: *2018 25th IEEE International Conference on Image Processing (ICIP)*.; 2018:3129–33.

7. LaLonde R, Xu Z, Irmakci I, et al. Capsules for biomedical image segmentation. *Med Image Anal* 2021;68:101889.

8. Crawford KL, Neu SC, Toga AW. The Image and Data Archive at the Laboratory of Neuro Imaging. *NeuroImage* 2016;124:1080–3.

9. Weiner MW, Veitch DP, Aisen PS, et al. The Alzheimer's Disease Neuroimaging Initiative 3: Continued innovation for clinical trial improvement. *Alzheimers Dement J Alzheimers Assoc* 2017;13:561–71.

10. Fischl B. FreeSurfer. *NeuroImage* 2012;62:774–81.

11. Fischl B, van der Kouwe A, Destrieux C, et al. Automatically parcellating the human cerebral cortex. *Cereb Cortex N Y N 1991* 2004;14:11–22.

12. Clerx L, Gronenschild EHBM, Echavarri C, et al. Can FreeSurfer Compete with Manual Volumetric Measurements in Alzheimer's Disease? *Curr Alzheimer Res* 2015;12:358–67.

13. Li X, Morgan PS, Ashburner J, et al. The first step for neuroimaging data analysis: DICOM to NIfTI conversion. *J Neurosci Methods* 2016;264:47–56.

14. Ganzetti M, Wenderoth N, Mantini D. Quantitative Evaluation of Intensity Inhomogeneity Correction Methods for Structural MR Brain Images. *Neuroinformatics* 2016;14:5–21.

15. Somasundaram K, Kalaiselvi T. Automatic brain extraction methods for T1 magnetic resonance images using region labeling and morphological operations. *Comput Biol Med* 2011;41:716–25.

16. Taha AA, Hanbury A. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC Med Imaging* 2015;15:29.

Figure 1: CapsNet (A) and U-Net (B) architectures. Both models process 3D volumes in all layers, with dimensions shown on the left side. *D*, *H*, and *W* respectively represent the depth, height, and width of the image in each layer. In (A), the number over the Conv1 layer represents the number of channels. The numbers over the capsule layers (ConvCaps, DeconvCaps, and FinalCaps) represent the number of pose components. The stacked layers represent capsule channels. In (B), the numbers over each layer represent the number of channels. In the 3D U-Net, the convolutions have stride=1 and the transposed convolutions have stride = 2. Please note that the numbers over capsule layers show the number of pose components, while the numbers over non-capsule layers show the number of channels.

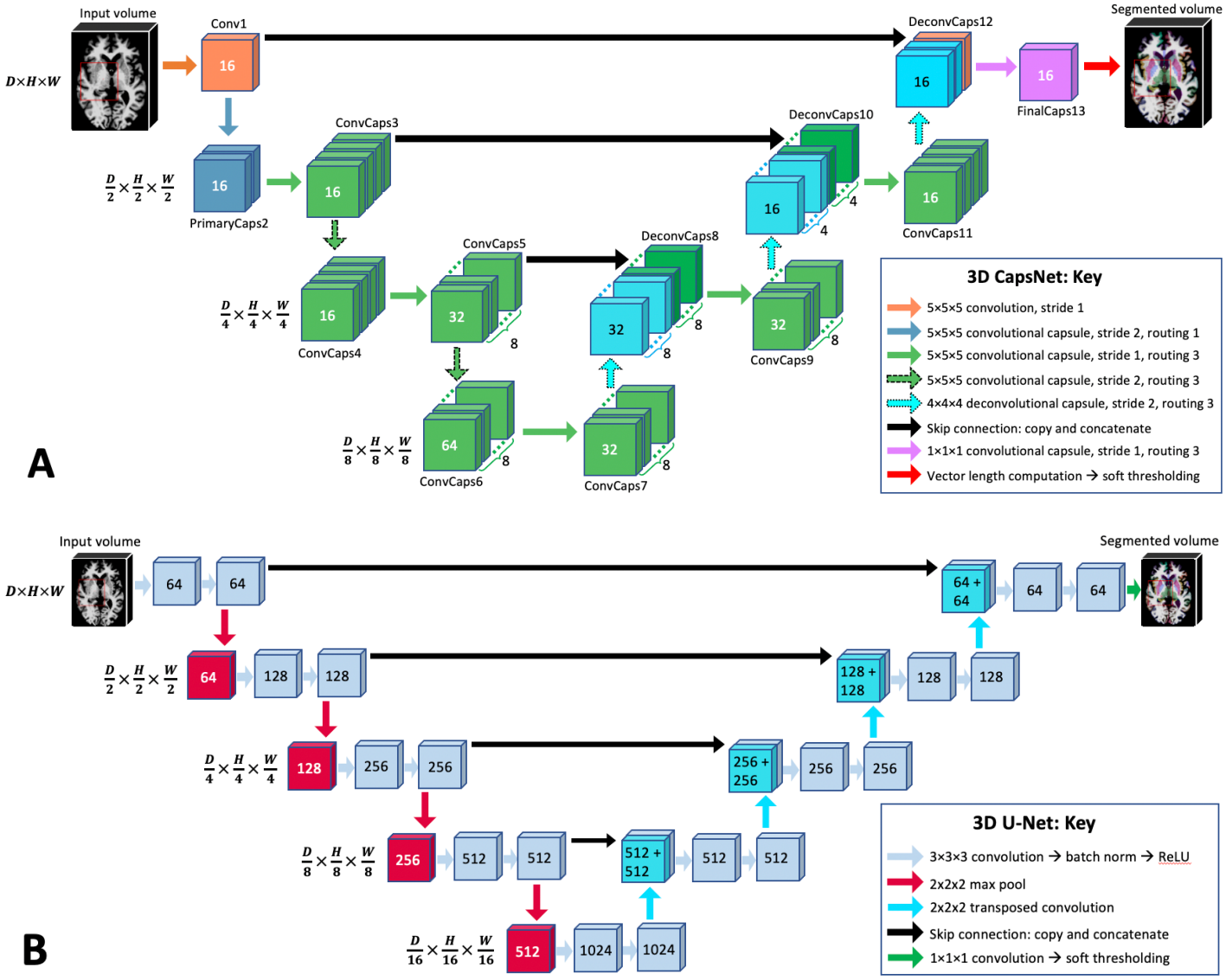

Table 1: Study participants. These accuracies were computed on the test set (114 brain MRIs). The 3rd ventricle, thalamus, and hippocampus respectively represent easy, medium, and hard structures to segment.

Data Partitions	Number of MRI volumes	Number of patients	Age mean ± SD	Gender % female	Ethnicity
Training set	3199	841	? ± ?	?%	?
Validation set	117	30	? ± ?	?%	?
Test set	114	30	? ± ?	?%	?

Figure 2: CapsNet vs U-Net in segmenting brain structures that were represented in the training data. Segmentations for three structures are shown: 3rd ventricle, thalamus, and hippocampus. Target segmentations and model predictions are respectively shown in white and red. Dice scores are provided for the entire volume of the segmented structure *in this case* (this case was randomly chosen from the test set).

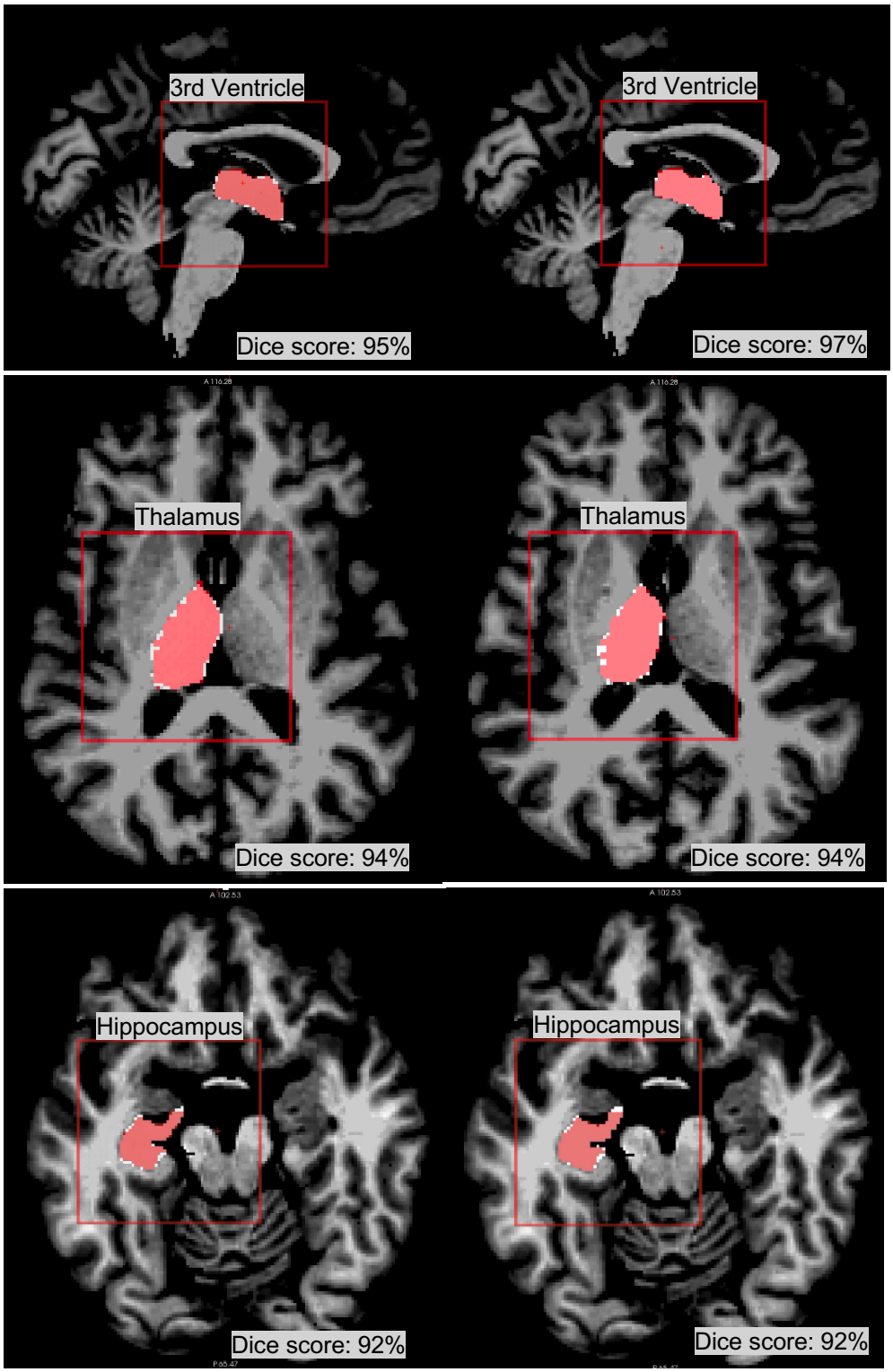

Table 2: CapsNet vs U-Net in segmenting brain structures that were represented in the training data. The segmentation accuracy was quantified using Dice scores on the test (114 brain MRIs). The 3rd ventricle, thalamus, and hippocampus respectively represent easy, medium, and difficult structures to segment.

Brain structure	CapsNet Dice score (95% CI)	U-Net Dice score (95% CI)	P-value†
3rd ventricle	93.6 (93.2 to 94.0) %	95.3 (95.0 to 95.6) %	< 0.01
Thalamus	93.6 (93.4 to 93.8) %	94.4 (94.3 to 94.6) %	< 0.01
Hippocampus	91.0 (90.7 to 91.3) %	92.5 (92.1 to 92.9) %	< 0.01

† Paired-samples t-test, degrees of freedom = 114 - 1 = 113

Figure 3: CapsNet outperforms U-Net in out-of-distribution segmentation. Both models were trained to segment right-sided brain structures, and were tested to segment contralateral left-sided brain structures. Target segmentations and model predictions are respectively shown in white and red. Dice scores are provided for the entire volume of the segmented structure *in this case*. While CapsNet partially segmented the contralateral thalamus and hippocampus, U-Net poorly segmented thalamus and entirely missed the hippocampus.

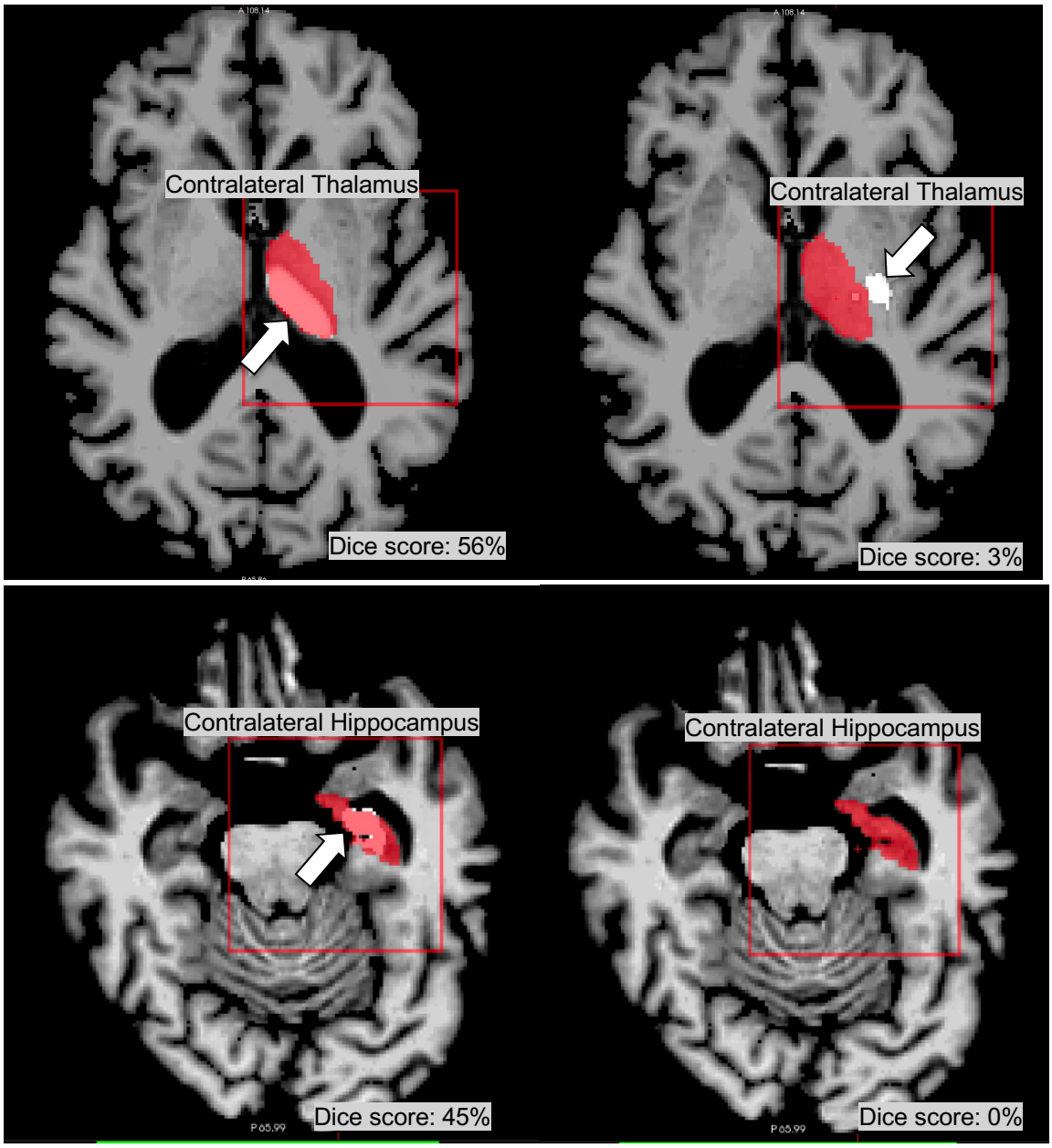

Table 3: CapsNet vs U-Net out-of-distribution segmentation accuracy. Both models were trained to segment the right thalamus and hippocampus. Then, they were tested on segmenting the contralateral left thalamus and hippocampus.

Brain structure	CapsNet Dice score (95% CI)	U-Net Dice score (95% CI)	P-value†
Thalamus	52 (46 to 58) %	16 (11 to 21) %	< 0.01
Hippocampus	43 (38 to 48) %	10 (6 to 14) %	< 0.01

† Paired-samples t-test, degrees of freedom = 114 - 1 = 113

Figure 4: CapsNet vs U-Net segmentation accuracy as a measure of training set size. When the size of the training set was decreased from 3199 to 600 brain MRIs, both models maintained their segmentation accuracy above 90%. Further decrease in the size of the training set down to 120 MRIs led to worsening of their segmentation accuracy down to 85% (measured by Dice scores).

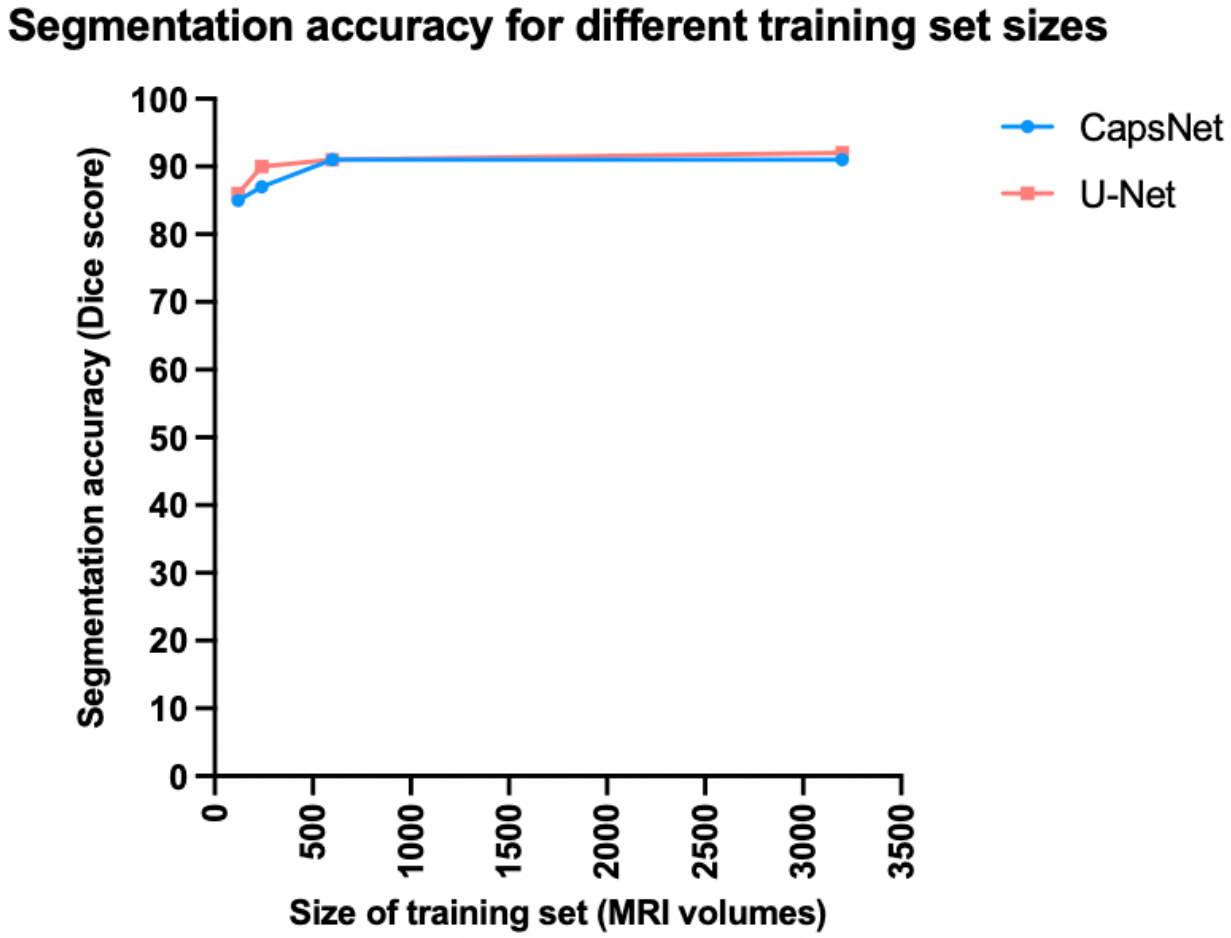

Figure 5: Model size (A) and computation times (B) compared between CapsNet and U-Net. The model size bars in (A) represent parameter size (28 and 345 MB for CapsNet and U-Net, respectively) plus the cumulative size of the forward and backward pass feature volumes (228 and 1364 for CapsNet and U-Net, respectively). The CapsNets train train slightly faster (B), given that they have 93% fewer trainable parameters. However, clustering between the capsule layers slows down CapsNets, making them only slightly faster than U-Nets during training. The two models are equally fast during testing.