Head and Neck CT Image Segmentation

Mrinal Jain
New York University
mrinal.jain@nyu.edu

Eelis Virtanen
New York University
ev933@nyu.edu

Abstract

Annotating organs at risk in CT images is very time-consuming process radiologists need to complete before performing radiotherapy. Therefore this task has received much attention among researchers to create automatic methods of creating these segmentations to save time. Our 2D U-Net architecture is a fully automatic method of performing this task. It highlights the importance of using various techniques like data augmentation and mix-up training to improve generalizability in the small data environment typical of many medical segmentation tasks. Furthermore, we show that residual connections are critical in such architectures to prevent the network's collapse. Our results indicate that our solution is close to being competitive with state-of-the-art approaches on a general level while exceeding performance with some specific organs. Our code is publicly available.[1]

1 Introduction

Head and neck image segmentation involves annotating CT scans to identify non-cancerous organs before radiotherapy to minimize radiation to surrounding healthy tissue. Therefore, the delineation of organs at risk (OARs) forms a critical component of successful cancer treatment. Traditionally, this task has been performed manually by radiologists who have needed to annotate hundreds of slices of a single 3D CT scan and has been very time-consuming. Furthermore, individual variations exist among annotators leading to a lack of consistency among labeled CT scans created by different annotators.

Deep learning could provide a solution to automate the process of creating these manual annotations. Furthermore, it offers the possibility of creating a more uniform standard of annotations if they prove to match human performance. There have been several public competitions based upon this task, most notably with MICCAI 2015 [1], which provides the data for this project. However, despite recent advances in this task within the community, no approach has yet been able to achieve the necessary performance that radiologists require to be willing to use the results of a fully automatic system for creating annotations. Even the best approaches still require the radiologist to adjust the annotations manually.

We use a deep learning framework by leveraging a 2D U-Net to segment the organs at risk using a network trained end-to-end. The initial input is a 3D volume, which we split into 2D slices followed by specific transformation before feeding it into the network. The output of the network will then be the segmentation mask for all of the organs at risk. Our main contributions are using residual units within the 2D U-Net, a weighted mix-up training strategy, and leveraging the boundary loss along with other region-based loss functions.

Our results show that we are very close to the state-of-the-art when considering all ten classes combined within the commonly used Dice Loss. Furthermore, on certain classes such as the chiasm, which has been very difficult to predict traditionally, we can exceed the state-of-the-art (see *Section 4.3* for more details).

[1] https://github.com/MrinalJain17/CT-image-segmentation

The remainder of this paper is structured as follows:

- We give an overview of existing research in the field that inspired our work in *Section 2*.
- In *Section 3*, *Section 4.1*, and *Section 4.2* we describe the available data set and our approach to the problem, supported by a number of experiments.
- We present our final results and a comparison with the current state-of-the-art in *Section 4.3* and *Section 4.4*.

2 Related Work

Medical image segmentation has a vast literature, but the focus in this paper is on Head and Neck CT image segmentation. Within this domain, approaches can be roughly split into atlas-based methods and learning-based methods.

Before the recent success in applying Deep Learning to medical segmentation, atlas-based methods were the most popular method and still enjoy using today. Within this approach, new images are aligned to a fixed set of example images [10]. Usually, the input image undergoes some transformation in order to make it compatible with the atlas. However, the main issue with this method is that it cannot handle anatomical variations very well since the atlas is fixed beforehand.

Learning-based methods enjoyed much success after the U-Net architecture [3] was introduced. The main idea is to add up-sampling operations to a typical contracting network in a more or less symmetric manner, creating a U-shaped network. When combined with additional data augmentation to learn properties such as invariance, it can produce state-of-the-art results.

On the other hand, the MICCAI competition [1] aims to provide a principled way of evaluating segmentation algorithms and runs every few years. The latest one in 2015 was the MICCAI Head and Neck Auto Segmentation Challenge (whose data this paper used as well for benchmarking purposes) showed that when comparing atlas-based and learning-based approaches, a clear winner was hard to find since teams using similar approaches often had quite different results due to the importance of fine-tuning each approach. However, atlas-based initialization of the segmentation was found to be the right approach. Many successful submissions combined both approaches.

In 2018 the *AnatomyNet* [2] (a learning-based approach) achieved state-of-the-art results on the same data set by using a 3D U-Net combined with squeeze-and-excitation residual blocks as well as a new loss function combining dice scores and focal loss. This current paper adopts many ideas from this paper and extends them, for example, by adding the boundary loss to the loss function.

3 Problem Definition and Algorithm

3.1 Task

Image segmentation involves taking an image and generating a binary mask with multiple possible regions of interest segmented out. In our case, we have 3-dimensional CT scans of the head and neck region as our input, and we intend to generate segmentation masks that indicate the presence/absence of 9 different regions. We provide a detailed description of our data set in *Section 4.1*.

A CT scan can be thought of as a 3D volume composed of multiple 2D slices. We tackle the problem from a 2D perspective - by considering each 2D slice an independent sample in our data set instead of an entire 3D volume.

There are two benefits of this conversion:

- Using convolutional neural networks with 2D images is computationally a lot cheaper than working with 3D volumes (that require the use of 3D convolutions).
- We can effectively use the existing best practices for the task of 2D image segmentation, which is a well-researched topic in the domain of computer vision.

A potential disadvantage of this method is that we lose the volumetric information contained in the CT scans. Research has shown that this volumetric information could help better segmentation; however, we do not experiment with 3D models as part of this project.

3.2 Algorithm

The general framework that we use for the modeling procedure is represented in *Figure 1*, and can be summarized as follows:

1. Given an input CT slice, apply transformations and deformations as part of the data augmentation pipeline. Use the transformed slice as an input to the segmentation model.
2. Generate the prediction from the model. We use the U-Net architecture [3] as our primary baseline.
3. Compute the loss between the prediction and the actual ground truth mask (essentially indicating how "correct" is the prediction).
4. Use gradient-descent to iteratively train the model and evaluate it on the hold-out test set.

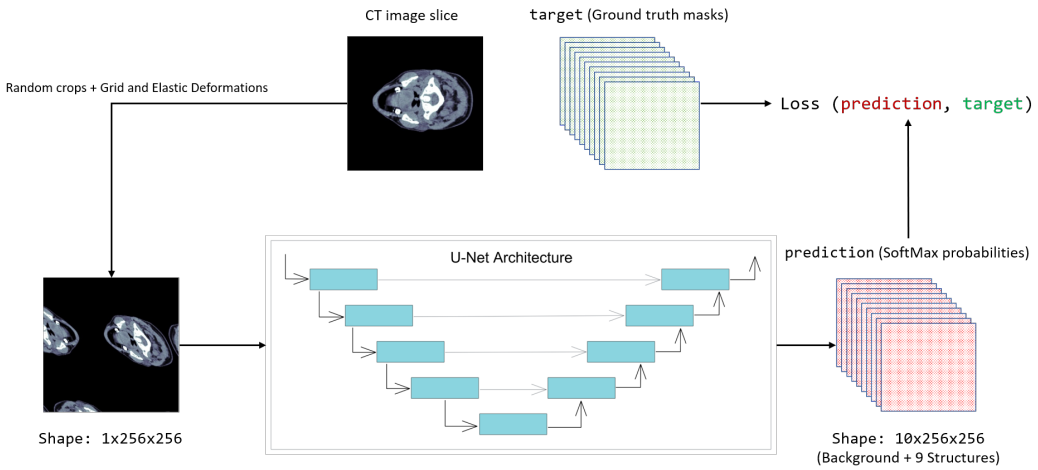


Figure 1: High-level overview of our training framework.

4 Experimental Evaluation

4.1 Data

We use the data released as part of the MICCAI 2015 Head and Neck Auto-segmentation Challenge [1]. In total, the data set consists of 48 CT scans (one for each of the 48 patients) with manual segmentations for nine regions: brain stem, mandible, chiasm, (left & right) optic nerves, (left & right) parotid glands, and (left & right) submandibular glands. Of the 48 total scans, 15 were set aside as the hold-out test set (precisely the same as the MICCAI challenge). The remaining were randomly split to form the training (25 scans) and validation (8 scans) sets.

Each CT scan is a 3D volume of dimension $N \times 512 \times 512$ - indicating it has N slices with a height and width of 512 pixels. The (binary) segmentation masks have the same dimensions as the CT scan, where a value of 1 indicates a specific structure's presence.

Traditionally, the numerical values of the voxels in CT scans are represented using the Hounsfield Scale.[2] For our data set, these values typically lie in the range $(-1024, 2000)$ as shown in *Figure 2*. All the structures except the mandible have HU values in the range $(-200, 200)$. Mandible, which is a bone, has a much broader range in comparison.

Pre-Processing We consider each 2D slice as an independent sample as opposed to an entire 3D volume. *Figure 3* summarises this process of conversion.

[2] https://en.wikipedia.org/wiki/Hounsfield_scale

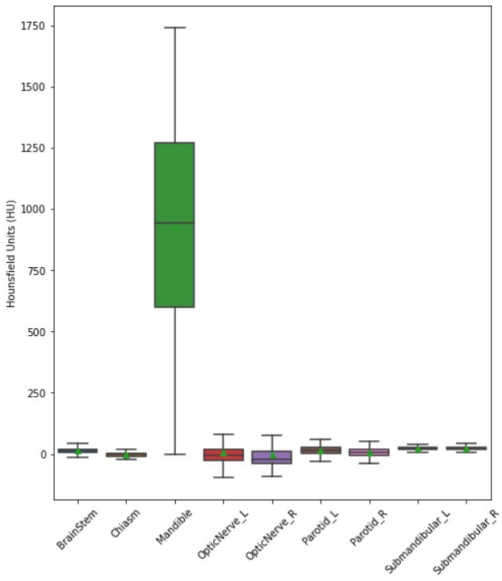


Figure 2: Distribution of HU values.

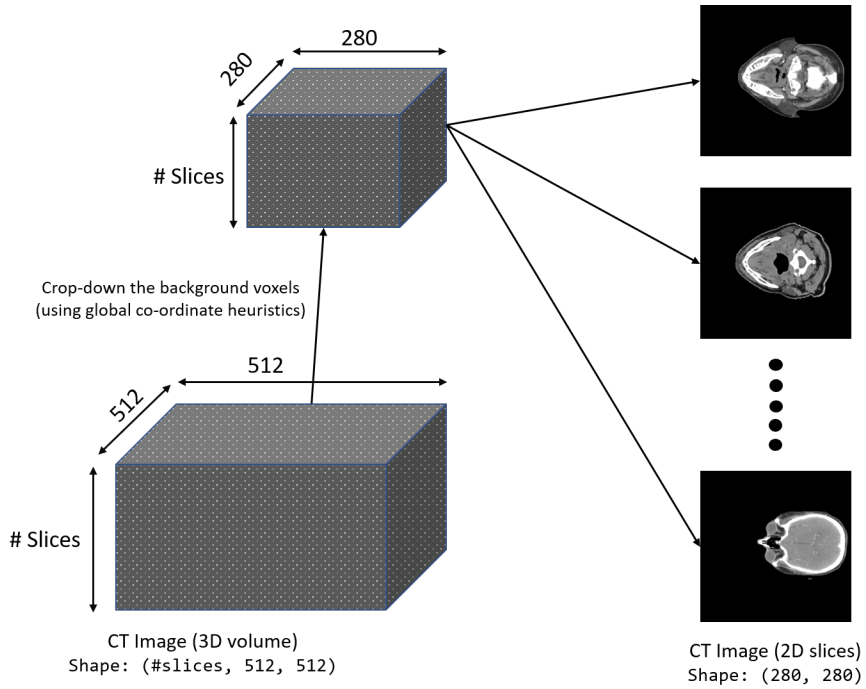


Figure 3: Conversion from 3D volumes to 2D slices.

4.2 Methodology

This section presents a detailed description of the set of techniques that led us to the final results. Note that we have the following three avenues to experiment with:

- The transformations and deformations that we apply within the data augmentation pipeline.
- The loss function used to train the network.
- Improvements/tweaks to the model architecture.

4.2.1 Data Augmentation and Pre-processing

Windowing *Figure 2* shows the pixel values of CT images span a wide range, and as it turns out, it is not easy to distinguish ≈ 3000 distinct gray scale levels - both for visualization and modeling. Windowing is a way to enhance the contrast of an image to highlight certain structures. We use the standard soft-tissue windowing heuristic[3], which essentially "clips" the pixel values to range $[-155, 195]$.

Random Cropping After the basic processing of CT scans as described in *Figure 3*, we end up with 2D slices of dimensions 280×280 pixels each. While training, we take random crops of size 256×256 pixels.

Deformations Geometric deformations like elastic transforms and grid distortions are extremely effective data augmentation techniques [3] that can significantly boost the model's performance and generalizability. We apply one of these transforms (selected at random) randomly with 50% probability to each slice.

Normalization Finally, we normalize the CT slices to have zero mean and unit standard deviation, which leads to better convergence of the models.

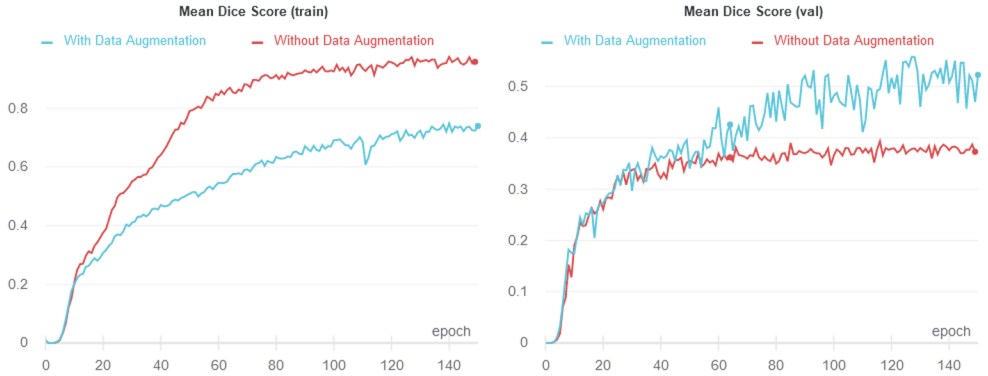


Figure 4: Improvements provided by data augmentation.

4.2.2 Loss Functions and Evaluation Metrics

We use the Dice coefficient (or Dice score) as the metric to evaluate the models' performance. In our experiments, we explore different combinations of the following loss functions:

Cross-entropy Loss CE is traditionally used for classification problems - and segmentation can be interpreted as per-pixel classification. However, CE does not address the problem of class imbalance. In the context of segmentation, the network cannot learn smaller structures as well as the others. Our data set particularly suffers a significant class imbalance as shown in *Figure 5*.

$$\text{Cross-entropy}(p,\ y) = \begin{cases} -\log p & y = 1 \\ -\log(1-p) & \text{otherwise} \end{cases}$$

Focal Loss [5, 2] At a high level, focal loss is an extension of the cross-entropy loss. It weighs down the "easy" examples, thereby implicitly making the network focus more on the "hard" ones.

$$\text{Focal}(p,\ y) = \begin{cases} -(1-p)^{\gamma} \log p & y = 1 \\ -(p)^{\gamma}\ \log(1\ -\ p) & \text{otherwise} \end{cases};\ \gamma = 2$$

[3] https://radiopaedia.org/articles/windowing-ct

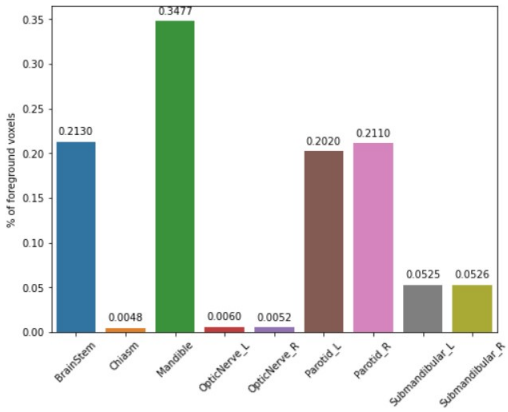


Figure 5: Distribution of the foreground voxels. Note that the foreground is only about 1% of the total volume, the remaining comprising the background (i.e., no structures)

Dice Loss [4] Based on the Dice coefficient, the (soft) dice loss measures the overlap between two samples. It is essentially a ratio: intersection over union, of the predicted and ground truth pixels.

$$\text{Dice} = 1 - \frac{2 \sum\limits_{\text{pixels}} p_i . y_i}{\sum\limits_{\text{pixels}} p_i + \sum\limits_{\text{pixels}} y_i}$$

Boundary Loss [8] Complementary to the other region-based losses, the authors of [8] framed boundary loss as a distance metric on the space of contours, not regions. Combining other region-based loss functions should produce more accurate boundaries, especially for imbalanced (small) regions. We compute the euclidean distance map[4] from the ground truth masks, and define:

$$\text{Boundary} = \sum_{\text{pixels}} p_i . d_i \; ; \; d = \text{Distance maps}$$

Missing Annotations There are some CT scans for which we do not have the labeled annotations for all the nine structures in our data set. More specifically, the annotations for the mandible and submandibular glands are missing from $\approx 20\%$ of the CT scans. We use the technique described in [2] to handle these missing annotations. Intuitively, we exclude those predictions in computing the loss, which do not have the corresponding ground truth mask.

4.2.3 Model Architecture

We use the standard U-Net architecture [3] as our primary baseline model. However, in our experiments, the network ended up collapsing after some iterations of the training procedure to predict "trivial" outputs. We replaced the traditional U-Net blocks with their residual counterparts - a Residual U-Net [6]. Residual connections provide significant benefits [7] for training deep networks, and in our case, they successfully prevented the network from collapsing. *Figure 6* shows how these residual units differ from the ones in a standard U-Net, and *Figure 7* provides the quantitative proof of them preventing the collapse.

4.2.4 Training Tricks

Overfitting is a significant issue when training neural networks, wherein the model "memorizes" the data that it is trained on but does not generalize well to new examples. This problem is even more pronounced when working with medical images because the available data sets are often relatively small compared to their "natural image" counterparts. For reference, we only have ≈ 1250 2D slices

[4] SciPy Implementation

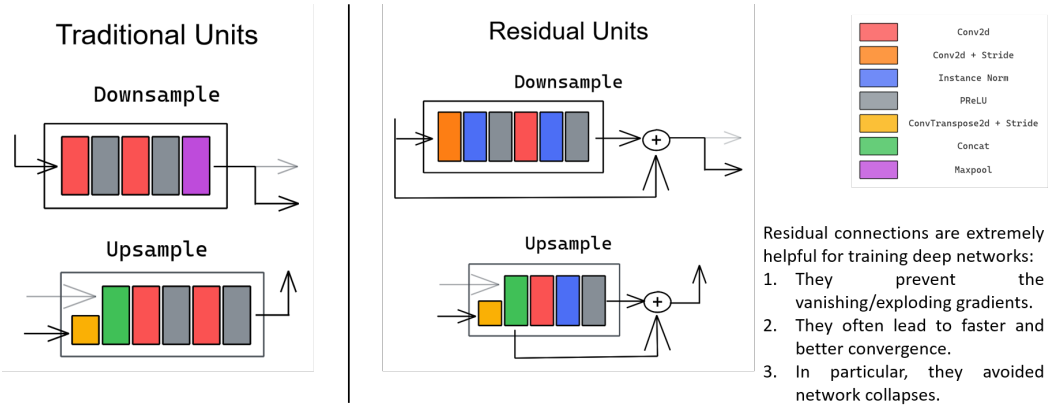


Figure 6: Traditional vs. Residual U-Net.

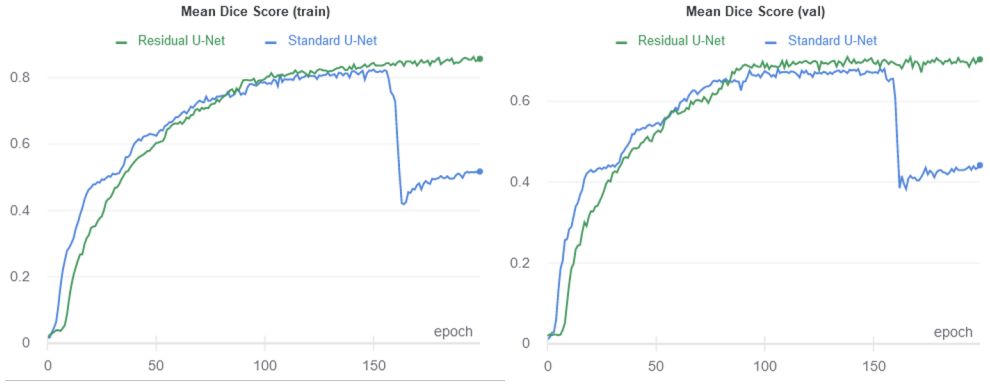


Figure 7: Improvement provided by the residual connections. Note the sudden drop in performance for standard U-Net after 150 epochs.

in our training data. We primarily rely on robust data augmentation to prevent overfitting; however, recent work has shown that a technique called "mixup" can be quite useful to help models generalize well.

Mixup Training [9] Mixup allows us to use a linear combination of two different samples when training the network. This *mixed-up sample* can be thought of as a super-imposed image and has led to improvements in various machine learning tasks. Note that the corresponding label is also a linear combination of the two ground truth segmentation masks, using the same coefficients.

$$x_{mixed} = \lambda x_1 + (1-\lambda)\, x_2 \; ; \; \lambda \in \beta(0.2,\, 0.2)$$

$$\begin{aligned} &\text{prediction } (p) = \text{model}(x_{mixed}) \\ &\text{Total Loss} = \lambda\, l_1 + (1-\lambda)\, l_2 \; ; \\ &l_1 = L(p,\, y_1),\; l_2 = L(p,\, y_2) \end{aligned}$$

Weighted Mixup To additionally deal with the class imbalance, we used a weighted mixup strategy. Instead of *mixing* two randomly sampled examples from a batch, we use the inverse of the number of annotations for each structure and virtually over-sample smaller structures (like chiasm) and under-sample larger structures (like mandible).

4.3 Results

Table 1 summarizes the final configurations that produced the best results on the validation set. For a fair and consistent comparison, all the experiments use the following hyper-parameters:

- The number of convolutional filters doubled every block in the U-Net starting from 64, all the way up to 1024 in the last down-sampling block.
- Models were trained for 200 epochs with an out-of-the-box Adam optimizer (learning rate was set to 0.001).
- Additionally, our final models' used learning rate scheduling - where the LR was halved if the validation dice scores did not improve for ten consecutive epochs. This further stabilized the training.

Table 1: Final model configurations. Both models produced similar results, with **Model M** being marginally better overall

	Model L (Large)	Model M (Mixup)
# Parameters	26 Million	13.5 Million
Loss function	Focal + Dice Loss	Focal + Dice + Boundary Loss
Training tricks	—-	Weighted Mixup

Comparison with State-of-the-art *Table 2* compares our model with the current state-of-the-art on MICCAI 2015 Challenge - Anatomy Net [2]. We present our hypothesis about these results further in *Section 4.4*. Moreover, we present some of the sample predictions[5] our model made on the test set in *Table 3*.

Table 2: Our Model vs. Anatomy Net

	Anatomy Net (3D)		Ours (Original Data - 2D)	
	Extended Data	Original Data	Model L	Model M
Brain Stem	86.65	58.6	86.37	85.53
Chiasm	53.22	39.93	57.52	55.05
Mandible	92.51	94.16	84.61	83.79
Optic Nerve L	72.1	74.62	66	65.87
Optic Nerve R	70.64	73.77	63.49	64.07
Parotid L	88.07	88.83	80.33	80.24
Parotid R	87.35	87.24	78.9	79.81
Submandibular L	81.37	78.56	66.6	70.81
Submandibular R	81.3	81.83	63.97	64.31
Average	**79.25**	**75.28**	**71.98**	**72.16**

4.4 Discussion

From *Figure 4* we can see that data augmentation helped a lot to increase performance. The data set is minimal, so we can significantly increase the size of the training set by doing this augmentation. Similarly, we allow the network to learn specific invariance properties that are common to medical images. Contrary to expectations, we did not find any benefit in using windowing as a pre-processing step despite not being shown individually. The main benefits were gained from the use of elastic transformations. On the other hand, *Figure 7* shows the impact of using residual units on performance. We can see that a catastrophic collapse of the network is avoided due to their inclusion, and learning can continue for longer. It was not entirely clear why they helped. Common explanations for their inclusion relate to preventing gradients from vanishing and retaining information from earlier, more extensive representations. In any case, they demonstrated a clear benefit.

From the results, we can see that our average dice score is a bit below the state-of-the-art-results (75.28 vs. 72.16). However, we can see that our model outperformed certain regions such as the Brain Stem and Chiasm. These are regions that are very small and have traditionally proved to be very difficult to predict. A common approach in dealing with small volumes is only to use a single downsampling layer since multiple layers make it challenging to capture tiny and granular details.

[5]We have setup interactive reports with a bunch of other visualizations: Model L, Model M

However, it appears the 2D network did a better job at predicting these organs. These results raise the possibility that in practice, the best implementation could potentially be made with an ensemble of a 2D and 3D network since these approaches should be slightly different and could hence benefit from a convex combination in their predictions. Nonetheless, generally speaking, it is plausible that we lose some critical 3D information when using 2D slices, which may explain why the 3D model performs slightly better. Furthermore, despite the 3D being more computationally demanding, this slight gain in performance will most likely be worth it for practitioners.

5 Conclusion

In summary, we have proposed an end-to-end framework that can segment head and neck CT images by leveraging a 2D U-Net. Our results show that we are very competitive with the state-of-the-art on a general level and outperform them on some individual structures. We used several techniques in order to achieve our results:

1. To alleviate issues from having a small data set, we used various pre-processing techniques along with a weighted mixup training to improve generalizability.
2. Residual units helped in preventing catastrophic collapses of the network.
3. By transforming the problem from 3D to 2D, we avoided a higher computational burden.

For the future, despite the computational challenges, exploring 3D techniques could be very interesting in order to avoid losing the 3D information in the problem. Furthermore, an ensemble of a 2D model and a 3D model could likely exceed state-of-the-art results. This work also showed the importance of having access to larger data sets to improve performance and is likely the final barrier preventing such models from being used in practice. Finally, the loss metric used was the Dice score, which is the commonly used metric to judge performance in segmentation tasks. However, it is not necessarily the one that radiologists care about the most in practice. Hence, one field of future research could explore new metrics that could potentially capture desired behavior in a more accurate manner.

6 Lessons Learned

Theory

- **With a small amount of data, it is important to use various techniques to prevent overfitting.** Since our training data was small, it was essential to augment the data with various transformations in the pre-processing stage, for example, by applying deformations and random cropping. Similarly, mix-up training was another useful tool that was used to help the model generalize better.
- **Residual units help.** Before introducing the residual units, the network suffered from catastrophic collapses after a certain number of epochs. After introducing the residual units, this phenomenon was significantly reduced.
- **3D is very expensive.** We looked at 3D implementations but realized it took at least an order of magnitude longer to train. The batch sizes had to be much smaller as well due to memory issues.

Practice

- **Higher level frameworks on top of PyTorch help with faster model prototyping.** Using PyTorch Lightning, running experiments was a lot easier since a lot of the boiler-plate code was abstracted away. Similarly, results were stored seamlessly, allowing more time to be used for new approaches instead of writing Python loops and keeping track of experiments.
- **Having an easy way of visualizing results is handy.** We used a framework called Weights and Biases.[6] It allowed the visualization of learning curves and predictions in real-time, which integrated very well with PyTorch-Lightning. This allowed us to diagnose the errors a new model made easily.

[6] https://www.wandb.com/

Acknowledgments

We would like to thank Prof. Narges Razavian for mentoring and guiding us throughout the project.

References

[1] Raudaschl, P. F., Zaffino, P., Sharp, G. C., Spadea, M. F., Chen, A., Dawant, B. M., & Jung, F. (2017), Evaluation of segmentation methods on head and neck CT: Auto-segmentation challenge 2015. *Medical Physics*, 44(5), 2020-2036. https://doi.org/10.1002/mp.12197 1, 2, 3

[2] Zhu, W., Huang, Y., Zeng, L., Chen, X., Liu, Y., Qian, Z., Du, N., Fan, W., & Xie, X. (2019), AnatomyNet: Deep learning for fast and fully automated whole-volume segmentation of head and neck anatomy. *Medical Physics*, 46: 576-589. https://doi.org/10.1002/mp.13300 2, 5, 6, 8

[3] Ronneberger O., Fischer P., Brox T. (2015), U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28 2, 3, 5, 6

[4] Sudre C.H., Li W., Vercauteren T., Ourselin S., Jorge Cardoso M. (2017), Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations. In: Cardoso M. et al. (eds) Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. DLMIA 2017, ML-CDS 2017. Lecture Notes in Computer Science, vol 10553. Springer, Cham. https://doi.org/10.1007/978-3-319-67558-9_28 6

[5] T. Lin, P. Goyal, R. Girshick, K. He, & P. Dollár (2017), Focal Loss for Dense Object Detection. IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 2999-3007. https://doi.org/10.1109/ICCV.2017.324 5

[6] Kerfoot E., Clough J., Oksuz I., Lee J., King A.P., & Schnabel J.A. (2019), Left-Ventricle Quantification Using Residual U-Net. In: Pop M. et al. (eds.) Statistical Atlases and Computational Models of the Heart. Atrial Segmentation and LV Quantification Challenges. STACOM 2018. Lecture Notes in Computer Science, vol 11395. *Springer, Cham*. https://doi.org/10.1007/978-3-030-12029-0_40 6

[7] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778. https://arxiv.org/abs/1512.03385 6

[8] Kervadec, H., Bouchtiba, J., Desrosiers, C., Granger, E., Dolz, J., & Ben Ayed, I. (2019), Boundary loss for highly unbalanced segmentation. Proceedings of The 2nd International Conference on Medical Imaging with Deep Learning, *PMLR* 102:285-296. http://proceedings.mlr.press/v102/kervadec19a.html 6

[9] Eaton-Rosen, Z., Bragman, F.J., Ourselin, S., & Cardoso, M. (2018), Improving Data Augmentation for Medical Image Segmentation. https://openreview.net/forum?id=rkBBChjiG 7

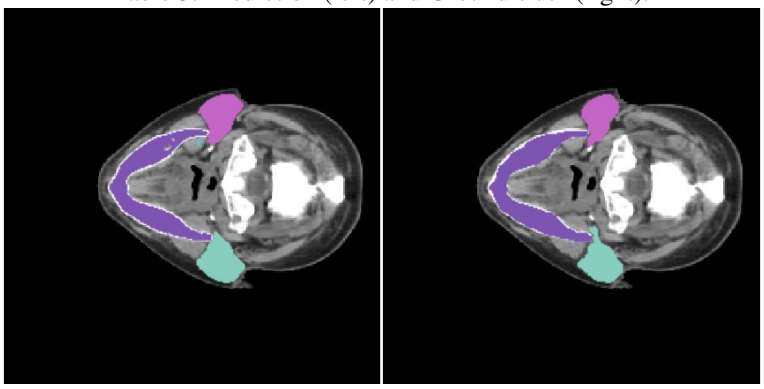
[10] Han, X., Hoogeman, M.S., Levendag, P.C., Hibbard, L.S., Teguh, D. N., Voet, P., Cowen, A.C., & Wolf, T.K., (2008) *MICCAI*, Atlas-based auto-segmentation of head and neck ct images. 2

Student Contributions

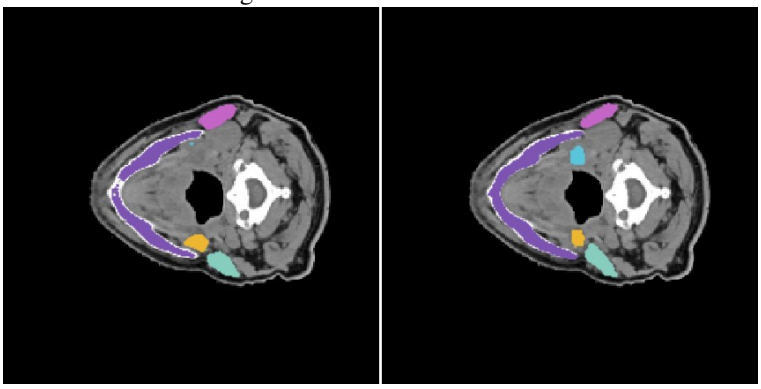
Mrinal: Implemented the end-to-end training and visualization framework, and worked on the report.

Eelis: Wrote minor portions of code and around half of the report.

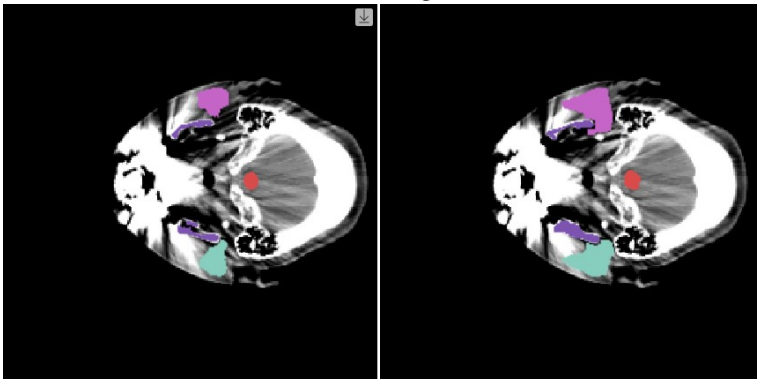
Table 3: Prediction (left) and Ground-truth (right).



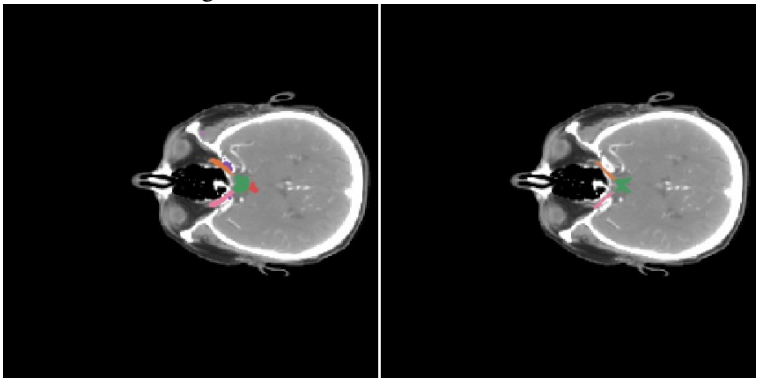
Mandible and Parotid glands



Mandible, Submandibular, and Parotid glands



Mandible, Parotid glands, and Brain Stem



Optic Nerves and Chiasm