2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)
Venice, Italy, April 8-11, 2019

ACCURATE SEGMENTATION OF DENTAL PANORAMIC RADIOGRAPHS WITH U-NETS

Thorbjørn Louring Koch[⋆†◇] *Mathias Perslev*[⋆] *Christian Igel*[⋆] *Sami Sebastian Brandt*[⋆†]

[⋆] Department of Computer Science, University of Copenhagen
[†] Department of Computer Science, IT University of Copenhagen
[◇] X1 Software Development, 3Shape Medical A/S

ABSTRACT

Fully convolutional neural networks (FCNs) have proven to be powerful tools for medical image segmentation. We apply an FCN based on the U-Net architecture for the challenging task of semantic segmentation of dental panoramic radiographs and discuss general tricks for improving segmentation performance. Among those are network ensembling, test-time augmentation, data symmetry exploitation and bootstrapping of low quality annotations. The performance of our approach was tested on a highly variable dataset of 1500 dental panoramic radiographs. A single network reached the Dice score of 0.934 where 1201 images were used for training, forming an ensemble increased the score to 0.936.

Index Terms— Dental radiography, Segmentation, Pantomogram, Deep learning, Fully convolutional neural network

1. INTRODUCTION

Convolutional Neural Networks (CNNs) have become state-of-the-art in medical image segmentation. For example, all top-ranking solutions in the *Medical Segmentation Decathlon*,[1] were based on Fully Convolutional Networks (FCNs). The arguably most prominent FCN architecture for medical semantic segmentation is the U-Net proposed by Ronneberger et al. [1, 2]. Likewise, we study FCNs based on U-Nets for semantic segmentation of dental panoramic radiographs. A dental panoramic radiograph, or pantomogram for short, is a 2D X-ray imaging modality for investigating the entire dentition. In contemporary scanners, the panoramic view is formed by fusing a set of narrow 2D projections along the tooth line.

While deep learning has been successfully applied within medical radiography, a little work exists in the dental domain [3], most notable the study by Ronneberger et al. [2] on dental X-ray segmentation of bitewings using a U-Net. So far only small benchmark datasets have been available within dental radiography (e.g., 39 patients in [2]). To address this problem, Gil Silva et al. [3] created a dataset consisting of 1500 pantomograms. In their paper, they reviewed the usage of classical segmentation methods, such as local thresholding and clustering, for segmenting dental radiographs. They also tested these methods on the pantomograms and argue that none of the classical methods reached the accuracy level needed for clinical practice. They propose a deep learning solution based on the Mask R-CNN architecture [4] as an alternative. In [5], they continue the study with Mask R-CNN and expand to instance segmentation, using a modified version of the dataset.

In this paper, we show how to compute high quality segmentations of the pantomograms using a U-Net architecture. We investigate how training data augmentation, network ensembling[6], choice of loss function, and *test-time data augmentation* affect the performance. For augmenting the training dataset we considered horizontal reflection. We tested different loss functions for network training including the Tversky loss [7]. Averaging prediction scores from a single network over multiple views of the input is denoted as test-time augmentation in this paper. This idea was already used in the seminal AlexNet [8] for classification tasks, where Krizhevsky et al. averaged over five $224\times$ 224 patches and their horizontal reflections to classify a $256\times$ 256 image. Ronneberger et al. [2] averaged over horizontal and vertical reflections when segmenting dental bitewings, and recent work [9] investigated the use of test-time augmentation for uncertainty estimation of the predictions. However, systematic studies of test-time data augmentation for medical segmentation have been missing so far.

[1] A challenge at *MICCAI* 2018 evaluating algorithms on 10 semantic 3D segmentation tasks, see `http://medicaldecathlon.com`.

2. METHODS

2.1. Dataset

We used the dataset of 1500 pantomograms created by Gil Silva et al. [3]. This dataset is the largest freely available for research within dental radiography. These radiographs are pixelwise classified into three classes: outside region of interest (ROI), teeth and implants (the foreground), not teeth and not implants (the background).

The images are divided into 10 patient categories. Eight of them are defined in Table 1. The last two are more than 32 teeth (6) and dental implants (5). The category size varies from 45 to 457 images. A training set were formed by randomly selecting 80% of the images from each category. The last 20% formed the test set.

2.2. Network Architecture

We used a FCN architecture based on the default U-Net as described in [1]. Because the input images are large, the network does not receive a full scan as input, instead it is applied to subregions, which we denote as *patches*. The size of the pantomograms in the dataset are (1127×1991). We modified the original U-Net by as follows. Firstly, we did not use dropout during training. Instead we used batch normalization [10] before each max pooling, up-sampling and concatenation layer. Secondly, for the optimization we used the Nadam optimizer [11] with default parameters. Finally, the weights were initialized using uniform Xavier initialization [12].

Category	1	2	3	4	7	8	9	10
32 teeth	Yes	Yes	Yes	Yes	No	No	No	No
Restoration	Yes	Yes	No	No	Yes	Yes	No	No
Appliance	Yes	No	Yes	No	Yes	No	Yes	No

Table 1. Definition of Patient Categories.

978-1-5386-3640-4/19/$31.00 ©2019 IEEE

We tested using zero padding in the convolutions for more efficient evaluation of an entire pantomogram. In our experiments, zero padding caused an asymmetry in the predictions. The network using padding made confident predictions whenever the full receptive field, i.e. all the input pixels the segmentation of a single pixel depends on, was part of the non-padded area of the patch. The performance dropped whenever the receptive field contained padded values. This issue were solved by test-time augmentation as described below. Reaching the same performance as without padding required a higher level of test-time augmentation. For example, by reducing the patch stride to an extend where it had a notable effect on computation time. The results in the following were computed without any padding. Unlike the standard U-Net, the 2×2 convolutions in the upsampling path were also without any padding.

2.3. Patching Scheme

Patching, as proposed in [1] uses two hyperparameters, the size and the stride. In an FCN architecture, the predicted segmentation is expected to be independent of the patch size. However, as will be discussed later, the patch standardization and the batch normalization depend on the actual patch size. While a large patch size allows for fast evaluation of a complete pantomogram, reducing the number of different pantomograms represented in calculating the optimization step increases the randomness of the steps. If we want a mini batch to fit into GPU memory, there is a trade-off between patch size and the number of pantomograms represented.

We found that 512×512 patches worked well with our hardware configuration. Without padding, only the central 309×309 pixels of the patch were segmented. The boundary of the pantomograms were reflected as suggested by Ronneberger et al. [1]. This boundary condition gives the most similar average intensity and intensity variance for the patches including pixels outside the pantomogram. The reflection boundary were also used during training to mirror the test scenario. In the preprocessing step the patches were standardized (zero mean, unit variance).

2.4. Loss Functions

The default loss function for U-Net and most other CNNs is cross entropy (CE). Our first U-Net configuration used CE with the three semantic classes described above. Looking through the dataset we realized that the ground truth annotation is inconsistent. The annotation had been performed by marking the tooth boundary at certain anchor points and interpolating between them. For most of the dataset the interpolated boundary is smooth. However, for the categories 7 and 8, the annotations have visible corners, see Fig. 2 and 3 for an example of the varying annotation quality. Furthermore, the ROI has sometimes been annotated by a smooth curve following the jawline, sometimes just by a rectangle. Due to this inconsistency, learning the ROI class was image specific. Therefore, we trained not only 3-class networks, but also 2-class networks that focus on the teeth segmentation. For the 2-class networks all ROI pixels were considered as background pixels. However, even though the 3-class networks learned the ROI class we did *not* use it while testing. During testing, the segmentation problem was turned into a binary segmentation problem within the ROI as in [3]. With test-time augmentation and ensembling, the final prediction was the average over all predictions for that pixel.

The segmentation learned by the CE loss is affected by the class imbalance. Reducing the impact of class imbalance is usually achieved by either the weighted CE loss, or by sampling from the less represented classes more frequent. However, recent work within medical deep learning suggest changing the loss function instead. For binary segmentation the Tversky loss [7] has shown promising results. This loss is defined as

$$\mathcal{T}(\boldsymbol{p}, \boldsymbol{g}|\alpha, \beta) = 1 - \frac{\boldsymbol{p} \cdot \boldsymbol{g}}{\boldsymbol{p} \cdot \boldsymbol{g} + \alpha \boldsymbol{p} \cdot \overline{\boldsymbol{g}} + \beta \overline{\boldsymbol{p}} \cdot \boldsymbol{g}}, \tag{1}$$

where $\boldsymbol{p}$ is the prediction score from the final softmax layer, $\boldsymbol{g}$ is a binary vector containing the ground truth. The bar indicates inversion; $\overline{\boldsymbol{p}} = \mathbf{1} - \boldsymbol{p}$. The parameters α and β specify the weighting between false positives (FP) and false negatives (FN) respectively. For $\alpha = \beta = 0.5$, the Tversky loss reduces to the Dice loss. For $\beta > \alpha$, FNs are penalized more heavily than FPs hence preferring increasing sensitivity before precision. In the experiments by Salehi et al. [7], $\alpha = 0.3$ and $\beta = 0.7$ gave the best Dice score for a highly imbalanced dataset, and we adopt this setting.

2.5. Data Augmentation

During training the patches were formed by selecting a random window of the specified size. The number of patches available for each image were equal to the resolution: $1127\times\ 1991 \approx 2.24 \cdot\ 10^6$ so the training set had more than a billion possible patches. With the chosen patching scheme at least 28 patches were needed for segmenting the entire pantomogram. The reason for using random patch selection and not a structured scheme, such as a grid, for generating patches is that this leads to variability in the standardization and the batch normalization. The huge number of slightly different patches helps to reduce overfitting.

A U-Net using patching and standardization learns to predict the pixel class within a range of average intensities and intensity variance. The range of the variance and the range of the mean depend on all pixels within the patch window, not only those in the receptive field. Hence, the predictions are influenced by the entire patch, and not only the receptive field. Larger patches limit the fluctuation of the mean and the variance. Smaller patches will make the trained network less susceptible to local variations. Reducing the patch size can be used as data augmentation during training. We found that increasing the patch size at test-time, compared to the patch size used at training, did not affect performance. Whereas decreasing the size reduced performance. The results presented in this paper were computed with the same patch size as during the training. In testing, the patches were created from a grid with a stride determined from the desired number of predictions for each pixel.

Perfectly aligned pantomograms of a symmetric face are symmetric over the central vertical axis. Flipping the patches along the horizontal axis therefore produced realistic patches. We tested whether using this symmetry improves performance.

3. RESULTS

3.1. Comparing the Configurations

The previous work [3, 5] on the panoramic dataset was tested with five evaluation measures: *accuracy, specificity, precision, sensitivity* and *Dice score* (DSR). For a proper comparison, we tested our network using the same measures.

We trained six configurations of U-Net for evaluation within the ROI. These configurations are: three class CE with bootstrapping and horizontal reflections (A: 3Cl+HR+BS), binary CE with horizontal reflections (B: 2Cl+HR), three class CE (C: 3Cl), binary

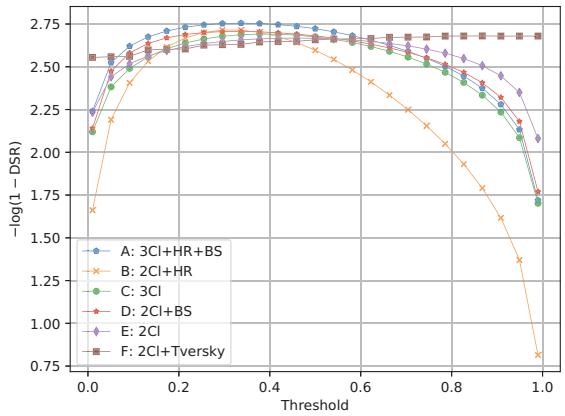

Fig. 1. Logarithmic Dice score as a function of softmax score threshold value. The flatter the curve the less the precision and sensitivity are affected by the selected threshold. A model trained with the Tversky loss varies little with the threshold because its predictions are approximately binary.

CE with bootstrapping (D: 2Cl+BS), binary CE (E: 2Cl) and Tversky loss (F: 2Cl+Tversky).

For the bootstrapping case we did not use the foreground segmentation masks from the dataset. Instead the foreground masks were found by evaluating the training data with Configuration C. Bootstrapping was tested because of the varying quality of the annotations. Since the U-Net does not have sufficient capacity for overfitting the entire dataset the low quality segmentations are improved by bootstrapping.

The upper part of Table 2 shows the results of these six U-Nets. The softmax scores were averaged over eight predictions during test-time. Horizontal reflections, bootstrapping, and using the ROI class during training increase the performance compared to Configuration E. As expected, combining all the three augmentation methods into a single network configurations gave the best single network result. Notably, E performs worse than C. We expect this is due to the increased class imbalance when the ROI class is treated as background. The best previous result on the dataset from [3] is also reported. However the comparison is not entirely fair, as [3] used an unknown 50% part of the data for training. The U-Net configurations reported here used 80%. The Mask R-CNN performs significantly worse compared with all the U-Nets. Notably, the significantly higher specificity than precision and sensitivity is troubling. It indicates that the true positive rate is low compared to the true negative rate. Hence, the classifier is not finding the difficult part of the teeth, but only the easy part. For this dataset, the end of the tooth-root is the difficult part.

In the follow-up study [5], the dataset was modified with an artificial boundary between all teeth, and only 193 images from the first four categories were used during training. The segmentations in these four categories are generally high quality, unlike the ones found for instance in Category 7 and 8. Categories 1-4 also only contain patients with 32 teeth and no dental implants. The values for the measures reported in [5] are for the entire image, not only the ROI. Replicating their training scenario is difficult without access to the modified segmentations. Instead, we trained B with 298 images in the training set. Our training set is still composed of images from all ten categories. The results of testing this network on the rest of the data are found in the lower part of Table 2 together with the result from [5]. Their Mask R-CNN is successful at detecting the

Model	Acc.	Speci.	Preci.	Sensi.	DSR
Mask R-CNN	*0.9208*	*0.9612*	*0.8373*	*0.7619*	*0.7944*
A: 3Cl+HR+BS	**0.9506**	0.9633	**0.9357**	0.9371	**0.9342**
B: 2Cl+HR	0.9486	**0.9636**	0.9356	0.9315	0.9315
C: 3Cl	0.9474	0.9543	0.9223	0.9420	0.9301
D: 2Cl+BS	0.9485	0.9607	0.9314	0.9351	0.9310
E: 2Cl	0.9460	0.9510	0.9176	0.9428	0.9278
F: 2Cl+Tversky	0.9454	0.9401	0.9041	**0.9572**	0.9277
Mask R-CNN	*0.98*	*0.99*	*0.94*	*0.84*	*0.88*
2Cl+HR	0.9720	0.9831	0.9294	0.9285	0.9266

Table 2. Results for single U-Net and previous Mask R-CNN.

background class while our U-Net is better for the foreground class. U-Net is inherently better at circumventing the class imbalance as can be seen by the higher DSR. The U-Net trained with 20% of the data achieved better DSR than both Mask R-CNN results. Due to the class imbalance the most important measure is the Dice score, and hence the precision and sensitivity.

Figure 1 shows a rescaled Dice score as a function of threshold for the U-Nets using the ROI during testing. The threshold is the value of the softmax score separating background and foreground pixels. After training, we selected threshold values that gave good Dice scores on the training data. For all CE configurations and all ensembles a threshold of 0.35 was used. For the Configuration F a threshold of 0.5. Unlike the CE loss, the Tversky loss (1) gives a strong incentive to push the output of the softmax layer to either one or zero. Accordingly, using the Tversky loss resulted in almost a binarization of the softmax scores. While the Tversky loss was successful at achieving a high sensitivity, the binarization of the predictions makes it more difficult to control the trade-off between sensitivity and specificity as compared to the cross entropy loss.

3.2. Test-Time Augmentation

Table 3 show the effect of test-time augmentation for Configuration C. N is the number of predictions averaged for each pixel. Due to the grid patching scheme N represent the number of times the images is *covered* by patches. The performance increases until 25 covers, but is minor after 9-12 covers. This result agrees with the method used by [8] for image classification. The performance impact is subtle. Accuracy only changes drastically between one cover and two covers. Specificity and precision drops and returns slowly to their original value as the augmentation increases. The major impact is for sensitivity. The sensitivity increasing indicates fewer pixels within the foreground class are classified as background.

Looking at the predicted segmentations we find the class boundary less jagged when using test-time augmentation. A segmentation with jagged edges appears worse to the human eye than missing the correct position of an edge by a pixel. Fig. 4 show an example of how test-time augmentation improves the segmenta-

N	Accuracy	Specificity	Precision	Sensitivity	DSR
1	0.9458	**0.9545**	**0.9226**	0.9371	0.9278
2	0.9468	0.9540	0.9220	0.9407	0.9292
4	0.9472	0.9542	0.9222	0.9416	0.9298
6	0.9473	0.9543	0.9223	0.9416	0.9298
9	0.9473	0.9543	0.9224	0.9416	0.9299
12	0.9475	0.9543	0.9225	0.9420	0.9301
16	0.9475	0.9543	0.9225	0.9422	0.9302
25	**0.9476**	0.9544	0.9226	**0.9424**	**0.9303**
36	0.9476	0.9544	0.9226	0.9423	0.9303

Table 3. Performance of U-Net C with Test-time Augmentation.

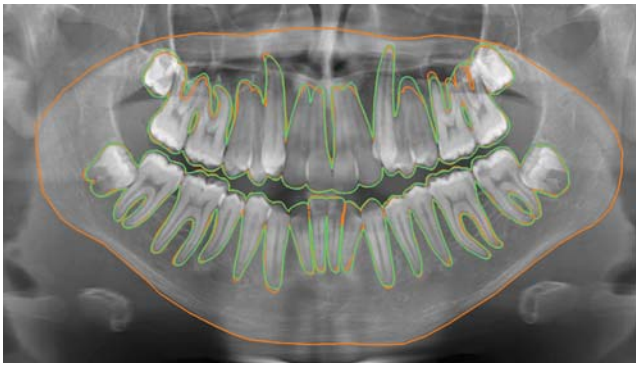

Fig. 2. Result with the highest DSR (0.97) for Category 4. The green curve is from ensemble A-D, and the orange is the ground truth. The curve along the jaw is the boundary between ROI and background.

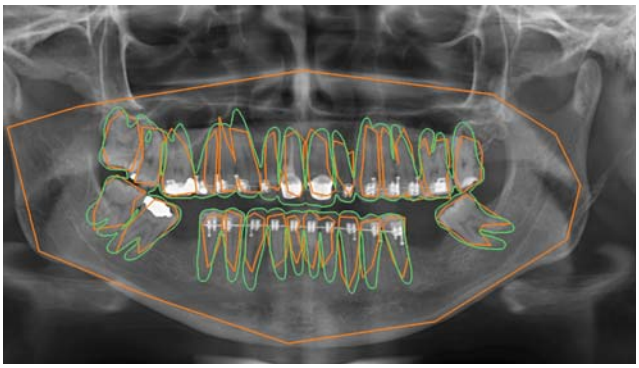

Fig. 3. Ensemble A-D result with the lowest DSR (0.78) in the test set. The low Dice scores are mostly due to crude annotations such as this one. For this pantomogram the U-Net segmentation (green) is far superior to the ground truth (orange).

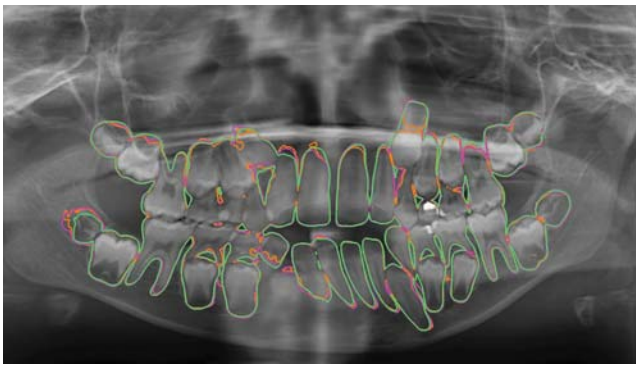

Fig. 4. Comparison between ensemble A-D (green, DSR=0.934), Configuration C with 36 covers (purple, DSR=0.932) and Configuration C with one cover (orange, DSR=0.921).

tion. Trusting the segmentation in a clinical setting requires not only a good score but also a realistic looking segmentation. The visual results indicate test-time augmentation has higher impact than the performance measures suggests. The reason is that the exact position of the class boundary is important visually, but the boundary is only a tiny fraction of all the pixels. This result were confirmed by the other U-net configurations.

Ensemble	Acc.	Speci.	Preci.	Sensi.	DSR
A-B	0.9517	**0.9640**	**0.9369**	0.9391	0.9358
A-C	0.9519	0.9612	0.9328	0.9438	0.9361
A-D	**0.9521**	0.9614	0.9331	0.9437	**0.9363**
A-E	0.9519	0.9596	0.9305	0.9460	0.9360
A-F	0.9509	0.9518	0.9199	**0.9546**	0.9347

Table 4. Results of U-Net Ensembles

DSR 1-5	0.937	0.952	0.960	0.955	0.950
DSR 6-10	0.943	0.839	0.932	0.957	0.954

Table 5. DSR for each Category with A-D Ensemble.

3.3. Model Ensembling

Table 4 contains the result of averaging the prediction scores from the six U-Nets reported in 2 before thresholding. Combining model A,B,C and D in an ensemble gives the best result on the test set. Including the Configuration E reduces performance. The best configuration is a merge of results with similar DSR. Configuration E and F has significantly lower precision than the other three, and hence DSR. Our results indicate ensembling a small number of models is only beneficial if the results are of similar quality. Otherwise the *worse* results are too influential. Including the Tversky configuration significantly decreases performance due to the binarization of the softmax scores. The difficult areas, i.e. where the softmax scores are near the threshold are dominated by the seemingly confident Tversky scores. Fig. 2 show a best case and 3 a worst case of the A-D ensemble. As can be seen in Fig. 3, the U-Net predictions surpasses the crude manual annotation. In Fig. 4, there is an example of how ensembling improved the smoothness of the boundaries beyond what the single network with test-time augmentation achieved.

The average DSR of the A-D ensemble for each of the 10 categories are found in Table 5. Only Category 7 and 8 have any Dice scores below 0.90. There is no correlation between the category size and DSR. Category 5 (implants) and 6 (supernumerary) are the most difficult categories, due to the higher internal variance. These both have lower performance as expected. The much lower performance for Category 7 and 8 originates in the segmentation quality. In Category 7, all the images are crudely annotated. For Category 8 it is only a fraction. Category 8 contains almost a third of the entire dataset. Hence the average over all the categories is largely affected by the low performance of this category. For Category 1 the annotations of the root are often cut a tiny bit short, compared to the other categories. The precision is therefore only 0.901 resulting in the lower DSR.

4. CONCLUSION

Teeth segmentation is the first step towards guided analysis of dental radiographs. Our U-Nets reached better segmentations of pantomograms than previous work using the same data, while relying on a smaller and simpler network architecture. The study confirmed that (i) exploiting symmetries in the input by training data augmentation, (ii) using an ensemble of networks, (iii) test-time augmentation, and (iv) bootstrapping can improve segmentation performance. We reached a Dice score of 0.936 by ensembling four end-to-end trained U-Nets with test-time augmentation, symmetry exploitation and bootstrapping. Future work could be to study test-time augmentation and ensembling with more recent architectures such as [13] or [14].

5. REFERENCES

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2015, pp. 234–241.

[2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "Dental X-ray image segmentation using a U-shaped deep convolutional network," in *International Symposium on Biomedical Imaging (ISBI)*, 2015.

[3] Gil Silva, Luciano Oliveira, and Matheus Pithon, "Automatic segmenting teeth in X-ray images: Trends, a novel data set, benchmarking and future perspectives," *Expert Systems with Applications*, 2018.

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask R-CNN," in *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 2980–2988.

[5] Gil Jader, Jefferson Fontinele , Marco Ruiz , Kalyf Abdalla , Matheus Pithon , and Luciano Oliveira , "Deep instance segmentation of teeth in panoramic X-ray images," in *Graphics, Patterns and Images (SIBGRAPI)*, 2018.

[6] Lars Kai Hansen and Peter Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.

[7] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour, "Tversky loss function for image segmentation using 3D fully convolutional deep networks," in *International Workshop on Machine Learning in Medical Imaging*. Springer, 2017, pp. 379–387.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[9] Murat Seckin Ayhan and Philipp Berens, "Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks," in *Medical Imaging with Deep Learning (MIDL)*, 2018.

[10] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning (ICML)*, 2015, pp. 448–456.

[11] Timothy Dozat, "Incorporating Nesterov momentum into Adam," in *International Conference on Learning Representations (ICLR) Workshop Track*, 2016.

[12] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Artificial Intelligence and Statistics (AISTATS)*, 2010, pp. 249–256.

[13] Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*. IEEE, 2017, pp. 1175–1183.

[14] Rodney LaLonde and Ulas Bagci, "Capsules for object segmentation," *arXiv preprint arXiv:1804.04241*, 2018.