Use of attention blocks to improve U-Net architectures for MRI reconstruction

Aleksander Jan Netzel, Can Candan, Nam Pho
Georgia Institute of Technology
Atlanta, GA
{anetzel3, ccandan3, npho3}@gatech.edu

Abstract

*Convolutional Neural Networks (CNNs) have a track record of adequate performance for image segmentation. However, they have inherent weaknesses (*e.g.*, need a lot of data, limited focal scope) that recent architectural innovations such as U-Net have sought to address. In the years since, many have attempted to incorporate advancements elsewhere in machine learning (*e.g.*, attention) towards U-Net architectures but these advancements have only explored CT scans so it remains to be seen how they will perform against other medical imaging modalities such as MRI scans.* ***To the best of our knowledge, this would be the first attempted exploration of U-Net improved with attention and other transformer-based features towards MRI reconstruction (using U-Net as a baseline)****. We also sought interpretability of the model performance using layer visualizations (*e.g.*, GradCAM). Our findings show that U-Transformer is a modest improvement over U-Net baseline with a SSIM score of 0.9058 vs 0.9026. Layer visualization suggests that this improvement comes through additional reconstruction of background regions in the MRI scan (*i.e.*, noise) and not specific medical tissue. Further study is recommended with a medical expert to better focus the attention mechanisms trained towards biologically relevant areas to make this model architecture clinically viable.*

1. Introduction

The Magnetic Resonance Image (MRI) requires patients to spend long periods of time in a low-throughput scanner. This ties up limited hospital equipment, a contributing factor for its high cost in clinical care. Physicians may therefore be reluctant to recommend an MRI scan in anything but the most exceptional cases despite their general diagnostic utility. Decreasing MRI procedure time by taking fewer scans while maintaining the image quality would lower healthcare costs, improve the patient procedural experience, and accelerate medical diagnoses. The 2019 fastMRI challenge made U-Net architecture its baseline for MRI reconstruction [7], we aim to improve on this architecture's performance through the use of attention mechanisms.

The U-Net architecture was developed by Ronneberger, *et al*. [11] in 2015 to address weaknesses in the previous state of the art by Ciresan, *et al*. [2]. The main contribution was to supplement the usual contracting convolutional network with a subsequent upsampling half that mirrors its contracting operations and skip connections between equivalent levels of each phase, creating a "U" shape to this architecture. Follow up work by Oktay, *et al*. [8] in 2018 improved upon the original U-Net architecture by adding attention gates to the upsampled half of the model. The most recent innovation upon U-Net was the work by Petit, *et al*. [10] in 2021 to add a transformer-based enhancement to U-Net through the use of multi-headed self and cross attention modules.

Image segmentation is a widely useful methodology across multiple domains [6]. Notable successes include demonstrated applications in autonomous driving [5], cellular microscopy [1, 11], CT scan processing [8], and MRI reconstruction [14]. Any improvement upon image segmentation will have impacts across all these areas given its general applicability and the increased availability and use of images across many fields. Recent innovations upon the original U-Net architecture [8, 10] have shown segmentation improvements for CT scans. We focused on MRI reconstruction as it is unknown how these U-Net innovations improve other medical imaging techniques.

The fastMRI data set from NYU Langone Health [14] consists of single and multi-coil MRI scans of knees and brains, including the raw k-space data and reconstructions (inverse fast Fourier transforms of the k-space data). For our analysis we focused on single-coil k-space data of knees with modifications for computational tractability on modest compute infrastructure (*i.e.*, a single NVIDIA RTX 3090 GPU) including selecting 25% of the total provided MRI scans for study at random and reducing their associated images (by center cropping to 80x80 sizes) for learning. Each scan consists of multiple slices for recreating the Z-axis volume and to reduce the amount of data to process we fur-

ther selected only the middle 15 slices from each MRI scan. For each slice of k-space data, we performed an inverse fast Fourier transform followed by an element-wise magnitude operation to get our new corresponding reconstruction image. Finally, we added the max and Euclidean norm of the reconstruction as attributes to our data so that we can work with the fastMRI libraries.

2. Approach

We started from the Facebook fastMRI code[1] repository, which included a reference implementation of a baseline U-Net model in PyTorch [9] and the training framework in PyTorch lightning [4]. Since we used a subset of the total MRI scans available and transformed (*e.g.*, cropped) the inputs, we had to run our own baseline model for reference in this paper. From the U-Net reference model we created an improved version using attention blocks called U-Transformer. Our U-Transformer model and all supporting code and results are publicly available[2].

The ideal metric for comparing a reconstruction with the reference baseline would be with the assistance of a board certified radiologist who could understand the nuances regarding the quality of image reconstruction. However, that would pose a problem to this study and something many who attempt this challenge anticipate. In lieu of a trained expert, the Structural Similarity Index Measure (SSIM) is used as the primary quantitative metric to judge model performance in addition to several others as described in the following section.

2.1. Metrics

Normalized mean square error (NMSE) is a metric for the sum of the squared differences between two sets of numbers. In our study we are interested in how closely related our model reconstruction (K) is compared to the provided reference reconstruction (I). The closer the reconstruction is to the reference, the smaller the NMSE value becomes (*i.e.*, smaller NMSE scores are preferred).

$$NMSE = \frac{1}{nm}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \qquad (1)$$

Peak signal-to-noise ratio (PSNR) is a metric for quantifying the quality of an image reconstruction (K) compared to a reference standard (I). One would prefer a strong signal-to-noise ratio and therefore PSNR approaches infinity as the reconstruction more closely matches the reference standard (*i.e.*, higher PSNR scores are better).

$$PSNR = 20 \cdot \log_{10}(max(I)) - 10 \cdot \log_{10}(NMSE) \qquad (2)$$

[1] https://github.com/facebookresearch/fastMRI
[2] https://github.com/dys129/fastmri

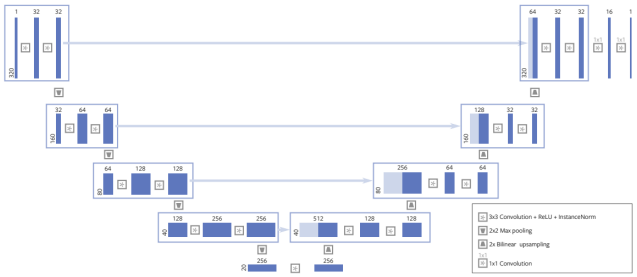

Figure 1. fastMRI baseline U-Net architecture from Figure 7 in Zbontar, *et al*. [14].

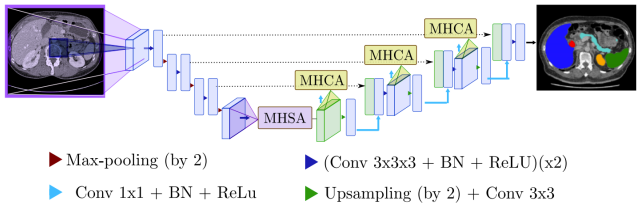

Figure 2. A U-Net model with transformer (i.e., attention block) modifications that Petit, *et al*. called "U-Transformer" from Figure 2 in their paper [10].

Structural similarity index measure (SSIM) quantifies the similarity between two images.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \qquad (3)$$

μ_x and μ_y are the average of x and y while σ_x and σ_y are their variance. σ_{xy} is the co-variance between x and y while c_1 and c_2 are stabilization variables.

2.2. U-Net

The baseline U-Net model architecture used is shown in Figure 1 borrowed from the recent fastMRI challenge paper [14]. As originally described by Ronneberger, *et al*. the model uses traditional contracting convolutional layers followed by a mirroring set of expanding convolutional layers with skip connectings between equivalent layers on either side [11].

2.3. U-Transformer

The main contribution behind the U-Transformer work of Petit, *et al*. is to use attention to "model long-range contextual interactions and spatial dependencies" [10]. In order to do so, two new blocks were introduced into a base U-Net architecture: Multi-Head Self-Attention (MHSA, Figure 3) and Multi-Head Cross-Attention (MHCA, Figure 4). The complete U-Transformer architecture is presented in Figure 2. While not a transformer in the strict sense as described by Vaswani, *et al*. [13], U-Transformer does make use of attention block improvements and a pseudo encoder-decoder architecture if one considers the contracting layers to be the

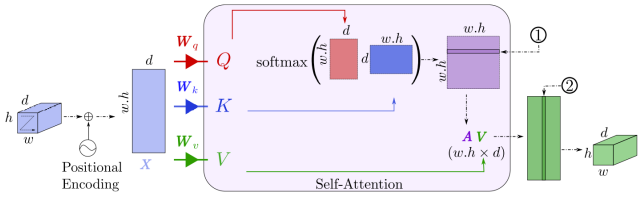

Figure 3. Multi-Headed Self-Attention (MHSA) is Figure 3 in the Petit, *et al*. U-Transformer paper [10].

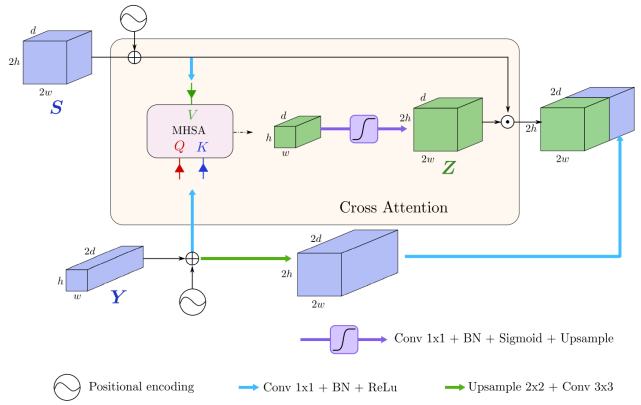

Figure 4. Multi-Headed Cross-Attention (MHCA) is Figure 4 in the Petit, *et al*. U-Transformer paper [10].

"encoder", the expanding layers to be the "decoder", and the layer in between these phases as a sort of "embedding".

Multi-Head Self-Attention (MHSA) modules are designed to extract long range structural information from the images (Figure 3). In the context of the entire U-Transformer architecture (Figure 2), it's positioned at the bottom of the U-Net. MHSA is responsible for connecting every element in the highest feature map with each other, thereby giving access to a receptive field of the entire input image.

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right)\boldsymbol{V} \qquad (4)$$

Attention was first proposed by Vaswani, *et al*. when the team introduced the concept of transformers in 2017 [13]. We use their original formulation as written in equation 4 where $\boldsymbol{Q} \in \mathbb{R}^{n \times d_k}$ is the matrix of queries, $\boldsymbol{K} \in \mathbb{R}^{n \times d_k}$ is the matrix of keys, and $\boldsymbol{V} \in \mathbb{R}^{n \times d_k}$ is the matrix of values.

Multi-Head Cross-Attention (MHCA) modules are used to turn off irrelevant or noisy areas from the skip connection features and highlight useful ones (Figure 4). Skip connections in U-Net architectures are used to connect high-resolution information laterally in conjunction with the information from the lower level from expanding layers. As shown in Figure 4, the skip connection S is first gated by the attention given to high level feature map Y then scaled to lie in (0,1) by going through a sigmoid function activation. The output shown as Z then can act as a filter if we element-wise multiply it with S. The filtered S and Y are concatenated before exiting the module.

3. Experiments and Results

All experiments in this section were performed using a single NVIDIA RTX 3090 with 24GB of GPU memory. The U-Transformer architecture was implemented in PyTorch 1.7.1 with CUDA 11.0 [9] using the base U-Net model provided in the facebook fastMRI repository as starter code. All experiments were run with RMSProp as the optimizer and batch size of 16. Each model was trained for a total of 50 epochs with an adaptive learning rate $\alpha = 0.001$ for the first 40 epochs and $\alpha = 0.0001$ for the last 10 epochs. Results of all experiments are presented in Table 1.

U-Net baseline. To have a baseline model for comparison against our architectural improvements, we trained the provided U-Net model using 64 channels at the top level and four downsampling layers. The same parameters were used for all U-Transformer experiments as well. The baseline U-Net model provided in the fastMRI library (Figure 1) [14] is similar to the original Ronneberger, *et al*. model [11] with the modifications of note being ReLu and Max Pool in the graph changed to LeakyReLu and Average Pool, respectively.

MHSA experiments. We based our implementation of the MHSA block on PyTorch's `nn.MultiheadAttention` class and used it to calculate self-attention as described in Petit, *et al*. [10]. The MHSA block was placed immediately after the convolution block at the bottom of U-Net. We added a positional encoding to the input, similar to positional encoding in the original transformer paper [13], but calculated separately for X and Y to distinguish among individual pixels. Because the module expects input in a shape of `(sequence length, encoding length)`, the input image needed to be reshaped into that form where sequence elements are pixels and the encoding are features from the previous layer. We ran several experiments to determine the optimal number of heads. According to the authors of U-Transformer, increasing the number of heads leads to an increased dice score for the segmentation task in CT scans. Notably, this is not something we observed in our experiments on MRI data. We obtained the best score for 4 heads and slightly lower results for other numbers. We attribute the difference to the unique input data for each study (*i.e*., CT vs MRI scans).

It's worth noting that each U-Transformer model, regardless of the number of heads, performed better than the U-Net baseline for all three metrics. Additionally, adding the new block increased the number of parameters over the

Model	Params	MHSA heads	MHCA heads	SSIM ↑	PSNR ↑	NMSE ↓
U-Net	31M	-	-	0.9026	32.35	0.02944
U-Transformer	35M	1	0	0.9058	32.49	0.02849
U-Transformer	35M	2	0	0.9053	32.48	0.02865
U-Transformer	**35M**	**4**	**0**	**0.9058**	**32.50**	**0.02842**
U-Transformer	35M	8	0	0.9047	32.45	0.02891
U-Net + Conv	49M	-	-	0.9032	32.43	0.02893
U-Transformer	39M	4	1	0.8993	32.17	0.03052
U-Transformer	39M	4	2	0.8905	31.78	0.03338
U-Transformer	39M	4	4	0.8984	32.17	0.03057
U-Transformer	39M	4	8	0.8985	32.13	0.03096
U-Transformer	35M	0	4	0.9008	32.44	0.02884
U-Transformer + MSE loss	35M	4	0	0.9017	32.52	0.02850

Table 1. Performance comparison of the U-Transformer architecture with varying levels of attention using a batch size of 16 and 64 channels compared to the baseline U-Net model using a batch size of 16 and 64 channels. U-Transformer shows an improvement over the baseline U-Net models.

Figure 5. Comparing U-Net performance after using differing training inputs, model inference shown using MRI scan `file1002428_v2.h5` center cropped to size 80x80. From left-to-right: (a) U-Net model trained against full sized k-space data but inference on a smaller image, (b) U-Net model trained against reduced 80x80 input and inference on same size 80x80 input, and (c) the ground truth reconstruction. If a model is trained on higher quality MRI images it will generate a more accurate reconstruction even if provided a lower-quality MRI than if the same model were trained on smaller MRIs and expected to reconstruct the same quality images it was trained upon.

baseline model. We wanted to verify if the increase in performance is because of the increased expressiveness of the model or if it is because of the attention block. Therefore, we ran an additional experiment including replacing the MHSA block with a convolution block, which means at the bottom level there are now 2 convolution blocks. This experiment allowed us to verify if increasing the number of parameters with convolution is enough to improve model performance. Although the number of model parameters increased significantly from 35M to 49M and performance did improve, it was not enough to outperform U-Net with the MHSA block (see *U-Net+Conv* in Table 1). The results of this model were worse for all metrics compared to any model with the MHSA block. This confirmed the claims made by Petit, *et al*. that model performance improvement is attributable to the attention architecture.

MHCA experiments. MHCA blocks takes two inputs: (1) $\mathbf{S}$ from corresponding encoder level and (2) $\mathbf{Y}$ from the upsampled previous decoder layer. Because both inputs go into an attention layer (Figure 4), the dimensions of $\boldsymbol{V}$, $\boldsymbol{Q}$, $\boldsymbol{K}$ need to be the same as the the ouput and in the shape of $w \times h \times d$, where w, h, d are width, height and number of channels, respectively. In order to do that there is a block *Conv 1x1 + BN + ReLU* that takes $\mathbf{Y}$ as an input and outputs $\boldsymbol{Q}$, $\boldsymbol{K}$ in the correct shape.

According to Petit, *et al*. [10] a similar block should

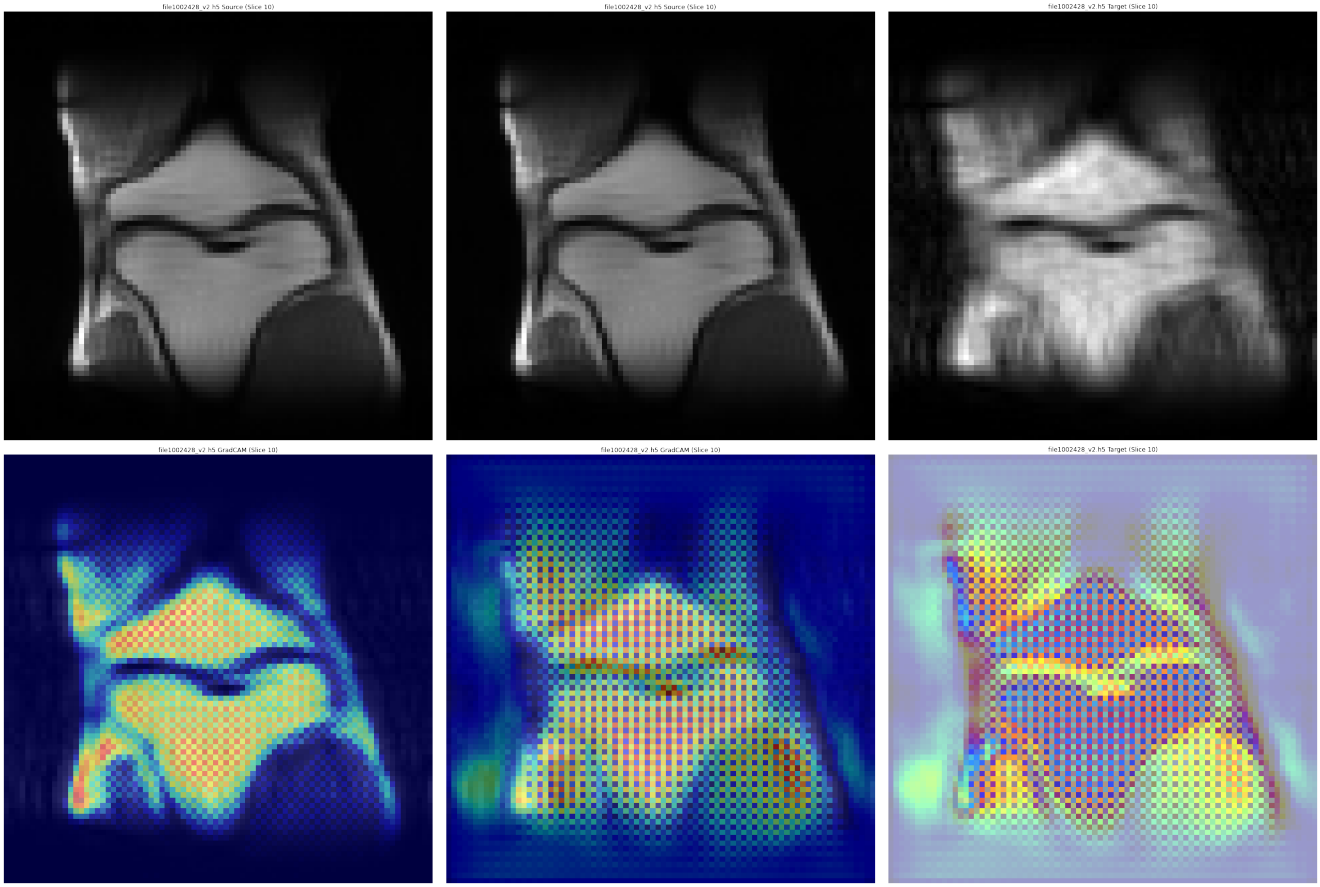

Figure 6. MRI scan `file1002428_v2.h5` from top-to-bottom and left-to-right: (1) U-Net reconstruction from the original k-space data, (2) U-transformer reconstruction from the original k-space data, (3) the provided target reconstruction, (4) GradCAM from the U-Net baseline, (5) GradCAM from the U-Transformer, and (6) GradCAM difference of U-Transformer versus U-Net baseline.

be applied to $\mathbf{S}$ to downsample it to a $w \times h \times d$ shape. The problem is that it suggests using *Conv 1x1(stride=2)* to achieve the desired shape since this convolution results in a loss of information and in our initial experiments performed very poorly. To overcome that issue, we tried using *Conv 2x2(stride=2)*, *AvgPool2d + Conv 1x1*, and *MaxPool2d + Conv 1x1*. Among these blocks the last one performed best and was incorporated into the final U-Transformer model.

Several other modifications to the MHCA block were made (*e.g.*, all *Upsample 2x2* were replaced in our implementation by transposed convolutions). The PyTorch implementation of *Upsample 2x2* when used with bilinear interpolation is not deterministic, therefore it would be difficult to compare results of different experiments if the results change with every run. Petit, *et al*. also suggest using BatchNorm and ReLU for normalization and activation functions, but the provided U-Net implementation uses InstanceNorm and LeakyReLU modules. We ran several experiments to verify which modules work best: LeakyReLU always improved the validation metrics and BatchNorm improved the validation metrics for the initial few epochs but it led to lower scores compared to InstanceNorm. Therefore for the final architecture, we used InstanceNorm and LeakyReLU in all MHCA blocks.

We placed an MHCA block at each upsampling layer, so we had four total MHCA blocks in the network. Additionally, Petit, *et al*. suggests including positional encodings for both $\mathbf{S}$ and $\mathbf{Y}$. However, in our experiments applying positional encoding as suggested in Figure 4 (*i.e.*, adding it to the inputs before the entire MHCA block) resulted in degraded performance. Instead, we opted for adding positional encodings only to the inputs of the attention block.

Table 1 presents the results of running with MHCA blocks with a different number of heads. In those experiments we also included MHSA block with four heads as that performed best in previous experiments. The results are worse than the U-Transformer with just MHSA block and baseline U-Net. We conclude that since the original purpose of MHCA was to allow the network to ignore noisy and irrelevant features from skip connections and attend

to areas that are important for segmentation, it might not necessarily help with reconstruction. Interestingly, we also experimented with only using MHCA blocks and it improved results according to NMSE and PSNR metrics over base U-Net but still performed worse on all metrics than U-Transformer with only MHSA blocks.

Loss function. The base U-Net model was trained using L1 loss but two metrics that our study used (*i.e.*, NMSE, PSNR) are closely related to MSE, which suggests minimizing MSE loss should improve those metrics as well. We tried using MSE loss but it did not improve the network performance. The only metric that was slightly improved was PSNR and both NSME and SSIM reported slightly worse outcomes. Since the fastMRI challenge is using the SSIM metric to compare models, we decide not to use MSE loss for our training.

Optimizer. We also experimented with using two different optimizers: RMSProp and Adam. We discovered that the Adam optimizer performed much better only in the initial epochs (*i.e.*, it required fewer epochs to achieve the same validation loss as RMSProp). During the 20 to 30th epoch, the loss and every metric plateaued while RMSProp was getting ahead with a lower loss. We tried scheduling a drop in the learning rate around the 20 to 30th epoch to overcome the plateau and it did help, but in the end RMSProp converged to a lower loss and better metric scores. Therefore, we used the RMSProp optimizer to train the final model.

Final results. The premise of this study was that using a fraction of the MRI image data (25% of the fastMRI scans) and further reducing each scan to 15 middle slices and center cropping each image to 80x80 would be valid trade off for model performance in exchange for training speed. In Figure 5 we show inference on a sample reduced MRI scan from (a) a U-Net model trained against the full set of scans with all provided images and slices, (b) a U-Net model trained against only the reduced scans, and (c) the ground truth reconstruction. We conclude that while a model trained against the full provided data provides the most accurate reconstruction, training on the reduced data provides sufficient compromise to evaluate model architecture improvements.

To get a valid comparison between U-Net and U-Transformer, one would need the best baseline U-Net using the reduced (*e.g.*, center cropped) training data. In order to do so we decided not to modify any training parameters so they were the same as for our experiments. We trained U-Net with a different number of initial channels: 32, 64, 128, 256. The results of that parameter tuning are presented in Table 3. We can see that U-Net with 256 channels over-fitted and out of the remaining experiments, U-Net with 128 channels performed the best and it was selected for our baseline. For U-Transformer we picked the model that performed the best in our experiments (see Table 1) which is U-Transformer with four MHSA blocks and no MHCA blocks and used the same number of channels as U-Net for a fair comparison.

Model	SSIM ↑	PSNR ↑	NSME ↓
U-Net	0.8305	30.31	0.03163
U-Transformer	0.8201	30.25	0.03193

Table 2. Final results on the test set.

Number of channels	SSIM ↑	PSNR ↑	NSME ↓
32	0.9003	32.14	0.03098
64	0.9026	32.35	0.02944
128	**0.9050**	**32.43**	**0.02913**
256	0.9047	32.36	0.02978

Table 3. Results of an experiment with different number of channels to get the best U-Net baseline.

We used the same reconstruction for target images on the test set as we did on the training set. Therefore, we were able to use the test set to do the final evaluation of our models. The results are presented in Table 2. As we can see, the results of U-Net are better in all metrics than U-Transformer, which means U-Transformer did not generalize as well on unseen data. One reason for that might be that we selected hyperparameters based on how well the model was doing on the same validation set, which introduces a bias into the model selection. A better approach would be to either use k-fold cross-validation or try multiple different seeds when training to gain confidence in how well the model performs. Unfortunately, because of the time limits, we omitted those additional runs.

While we were able to get a novel U-Transformer architecture to have similar performance to the U-Net baseline, we anticipated more substantial performance gains based on the results in other literature [8, 10, 13]. One method of investigating the underlying focus is to utilize layer visualizations such as GradCAM [12] and we showed the gradients of both U-Net baseline and U-Transformer (Figure 6). U-Net followed the contours of the knee very closely while U-Transformer seemed to focus on both the knee and additionally to areas of the MRI scan that were background. From this visualization we can conclude that the U-Transformer model performance change is due to the additional background scan focus since the original U-Net exclusively focused on the biological tissue.

4. Other Sections

Although the attention mechanism in neural networks is targeting the modeling of long-range dependencies, it's still a challenge to come up with methods that can work efficiently for images. It has also been observed that

Student	Contributed Aspects	Details
anetzel3	Implementation, experiments, report	U-Transformer implementation, running experiments, writing Experiments and Results section.
ccandan3	Implementation, experiments, report	Came up with the idea of using attention mechanism in U-Net architecture, found the U-Transformer paper and implementation [10]. Implemented the data reduction scheme so that we can work with our GPUs. Ran the baseline UNet and U-Transformer with toy versions of the data, interpreting the results. Writing report sections.
npho3	Implementation, experiments, report	Prepared fastMRI data, generated figures and visualizations, ran baseline experiments and parameter tuning, interpreting results, wrote sections of the report.

Table 4. Contributions of team members.

transformer-based network architectures proposed for vision applications require large-scale datasets to train properly [3]. Therefore data efficiency and computational complexity is likely going to be the focus for future work.

5. Work Division

Refer to Table 4 for a list of authors and their contributions.

References

[1] Yousef Al-Kofahi, Alla Zaltsman, Robert Graves, Will Marshall, and Mirabela Rusu. A deep learning-based algorithm for 2-d cell segmentation in microscopy images. *BMC bioinformatics*, 19(1):1–11, 2018. 1

[2] Dan Ciresan, Alessandro Giusti, Luca Gambardella, and Jürgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. *Advances in neural information processing systems*, 25:2843–2851, 2012. 1

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 7

[4] William Falcon et al. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, 3, 2019. 2

[5] Yiqi Hou, Sascha Hornauer, and Karl Zipser. Fast recurrent fully convolutional networks for direct perception in autonomous driving. *arXiv preprint arXiv:1711.06459*, 2017. 1

[6] Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1

[7] Matthew J Muckley, Bruno Riemenschneider, Alireza Radmanesh, Sunwoo Kim, Geunu Jeong, Jingyu Ko, Yohan Jun, Hyungseob Shin, Dosik Hwang, Mahmoud Mostapha, et al. State-of-the-art machine learning mri reconstruction in 2020: Results of the second fastmri challenge. *arXiv preprint arXiv:2012.06318*, 2020. 1

[8] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018. 1, 6

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019. 2, 3

[10] Olivier Petit, Nicolas Thome, Clément Rambour, and Luc Soler. U-net transformer: Self and cross attention for medical image segmentation. *arXiv preprint arXiv:2103.06104*, 2021. 1, 2, 3, 4, 6, 7

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1, 2, 3

[12] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 6

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 2, 3, 6

[14] Jure Zbontar, Florian Knoll, Anuroop Sriram, Tullie Murrell, Zhengnan Huang, Matthew J Muckley, Aaron Defazio, Ruben Stern, Patricia Johnson, Mary Bruno, et al. fastmri: An open dataset and benchmarks for accelerated mri. *arXiv preprint arXiv:1811.08839*, 2018. 1, 2, 3