fastMRI - Attentive Reconstruction Project Report

Authors listed in alphabetical order

Arun Kumar Siddanahalli Ninge Gowda
agowda35@gatech.edu
Georgia Tech

Kamil Żmijewski
kzmijewski3@gatech.edu
Georgia Tech

Kamran Munawar
kmunawar3@gatech.edu
Georgia Tech

Radu Diaconescu
rdiaconescu3@gatech.edu
Georgia Tech

Abstract

MRI scans provide high quality images that often serve as an important imaging diagnosis tool for many ailments, although with slower acquisition times compared to other imaging modalities like ultrasound and X-ray CT. This long acquisition time leads to high costs for the end beneficiary, i.e. the patient. Also, such longer acquisition times cause discomfort to the patients and limit the applicability of MRI to situations where faster scans are required. An approach to minimize the time required for scanning (thereby lowering the costs) is to reduce the number of measurements taken and use only those to reconstruct the images. Deep Neural Networks (DNN) represent an avenue of research for said construction of images using fewer measurements and they are able to achieve good reconstruction results. However, existing methods focus on improving the image quality across all the regions rather than the ones that are clinically significant. This project aims to explore and improve DNN's MRI reconstruction capabilities for knees by focusing on areas that are clinically relevant and significant; more specifically the areas that typically contain pathologies. We do this by using masks and heatmaps in order to guide the network to focus on said areas. In this paper, we describe our approach and our experimental setup, report on our results and analyse our findings.

1. Introduction/Background/Motivation

Magnetic resonance imaging (MRI) is an potent tool for diagnosis for a plethora of disorders: neurological, musculoskeletal and oncological diseases. This is due to the excellent soft tissue contrast it offers. The downside of it is a long acquisition time, which depending on the body part examined, can easily take up to an hour, leading to a low patient throughput. In order to alleviate this issue, the topic of decreasing acquisition time has been a researched since the introduction of MRI machines in medical imaging.

A major breakthrough was the introduction of undersampling of the measurements taken by MRI machine, with minimal impact of the quality of the final diagnosis images. This technique is called compressed sensing. Concretely, during MRI data acquisition, magnetic fields with different properties are generated which induce the body to emit resonant electromagnetic response fields which are then measured by receiver coils in the machine. These measurements are represented in a multidimensional Fourier space called the k-space. Undersampling here means sampling up to a lower maximum frequency, which results in images of lower spatial resolution. The notion of acceleration factor is used here which means "the ratio of the amount of k-space data required for a fully sampled image to the amount collected in an accelerated acquisition" [8]. Clinically, acceleration factors are typically 2 or 3.

During the last few years, there has been an explosion of interest in Deep Learning (DL) based techniques for image reconstruction [12]. These include, but are not limited to: Super Resolution models, adversarial networks training, self-supervised losses etc. Loss functions are typically customized in order to improve the image reconstruction quality. For example, perceptual loss, image loss and frequency loss can all be integrated into the final loss function in order to preserve the fine structures in the reconstructed images [15].

Our work also falls in the area of loss modification. We add weighting to the loss function by way of masks and heatmaps, giving more importance to areas where pathologies of the knee are usually located. For obtaining the models to generate heatmaps the MRNet dataset [5] was used and for reconstruction the fastMRI dataset [1]. For both

datasets, only the coronal view of the knee was used, with the additional constraint of using only single coil data for the fastMRI dataset. Additionally, only the middle slices of MRI volume were used because these are the most clinically relevant. The MRNet dataset training set contains 1130 MRI exams of training data and the validation set 120 exams. The fastMRI training set contains 973 MRI exams and the validation set 199. We include detailed results, figures and analysis of our findings which could be useful for further research looking into adding an improvement to existing SOTA architectures for MRI reconstruction. The idea of heatmaps and masks to guide the network is versatile and can be applied in many scenarios.

2. Approach

Our work is split into 3 parts: Scan region/Binary mask generation, Heatmap generation and the usage thereof in MRI reconstruction. The binary masks are generated using traditional computer vision techniques such as dilation, erosion, blurring, GrabCut [11] algorithm. The heatmaps are obtained by first training an image classifier on the MRNet dataset which learns whether the MRI presents an ACL tear or not. Then the model is used for inference on the fastMRI data, with Class Activation Maps (CAM) [19] being extracted and used as heatmaps for guiding the learning during reconstruction. For the classification model we developed an end-to-end training and experimentation pipeline in Pytorch, using baseline model definition and data loader parts from [2, 9] as the starting point. We made extensive changes to have a pipeline that would enable us to easily explore and experiment different model architectures, classification problem setups (Binary:one vs. rest, multiple vs. rest, Multilabel) and loss functions. For the MRI reconstruction training, we initiated the process by implementing a basic FastMRI U-Net model in a Jupyter Notebook. This involved using components such as the U-Net model definition and metric computation from the FastMRI repository [3] and other online sources [4, 6]. Subsequently, we developed a comprehensive end-to-end training and inference pipeline in PyTorch, incorporating similar components like the U-Net model definition and metric computation from the FastMRI repository [3]. This approach was adopted to gain a deeper understanding of what a complete deep learning workflow entails in PyTorch, as well as to facilitate easier modifications for our experiments, considering that the originial repository used the Pytorch Lighting framework. During reconstruction model training, the binary masks and heatmaps are used to enhance the spatial resolution of the clinically relevant areas of the knee MRI by increasing the loss for those areas. Concretely, during training, a subsampled MRI image is given as input, along with the corresponding heatmap and the binary mask. The output of the U-Net is compared with the target image and the weights are adjusted accordingly. This workflow is illustrated in Fig. 1(a). During inference, the subsampled MRI image is passed to the model and the reconstructed image is generated. This can be seen in Fig. 1(b).

2.1. Scan Region/Binary masks generation

In order to focus the MRI reconstruction task on the scan regions (foreground) that contain the anatomy under study, we had to develop algorithms to segment an MRI image into foreground and background regions. Since the fastMRI dataset does not contain any annotations explicitly marking these regions (to train any data driven ML or DL models), we developed an algorithm based on classic computer vision methods to perform the region segmentation. Fig. 2 illustrates the algorithm we developed for this segmentation. This algorithm consists of two components, namely the ImageProcessingBlock, which is based on chaining various primitive image processing operations; and the GrabCutBlock based on GrabCut [11] image segmentation algorithm.

While ImageProcessingBlock accurately detected entire foreground regions, it suffered from having islands of background regions within the foreground regions. On the other hand, GrabCutBlock had issues in correctly identifying the boundaries of foreground regions; while the detected foreground regions were complete without any background islands. Such complementary benefits associated with each of the methods prompted us to use a combination of both these methods to generate good quality segmentation. More specifically we fuse the segmentation masks from both the methods using a pixel wise binary OR operation. We observed that these fused segmentation masks were accurate on most of the MRI image slices, and hence we used these masks for all the MRI reconstruction experiments done in this project. Fig. 3 illustrates the sample segmentation masks generated by each of the algorithms on fastMRI images.

2.2. Heatmaps generation

The heatmaps to highlight clinically relevant anatomical regions in the MRI image were obtained using the CAM of an image classification (pathology detection) model trained on the MRNet dataset. Since the fastMRI dataset does not have such annotations, we had to use the MRNet dataset to develop such a model and then use the model to generate heatmaps for the fastMRI scans/exams.

The MRNet dataset has annotations for three anomalies/classes, namely acl tears (“acl”), meniscus tears (“meniscus”) and other generic abnormalities (“abnormal”). Unlike the per class binary classification models used in the original MRNet paper, we explored various model configurations based on different grouping of these individual classes. One avenue was to develop a single multilabel clas-

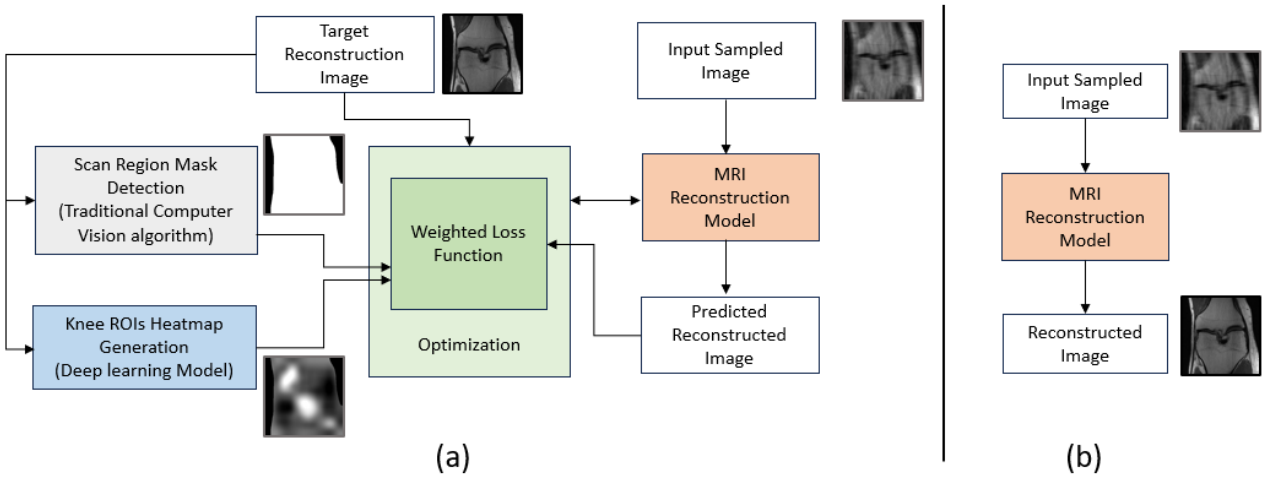

Figure 1: MRI Reconstruction Model **(a)** Training Workflow **(b)** Inference Workflow

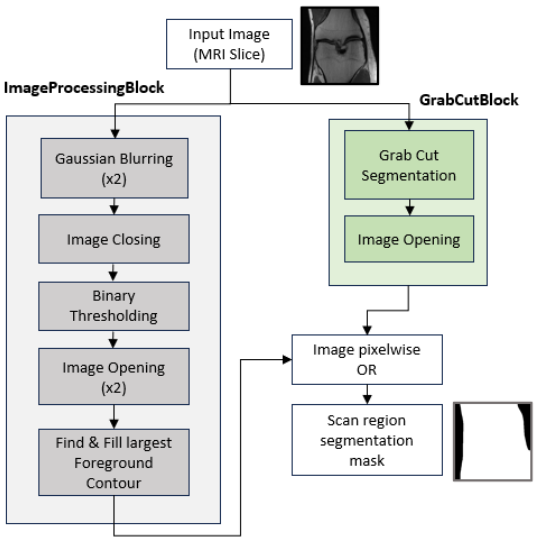

Figure 2: Scan Region Mask Generation Algorithm

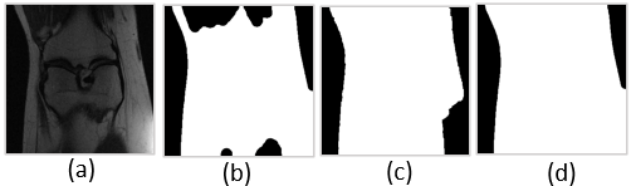

Figure 3: Scan Region Mask Generation, **(a)** Input Image, **(b)** ImageProcessingBlock Segmentation, **(c)** GrabCutBlock Segmentation, **(d)** Fused/Combined Segmentation

sification model with three outputs (one per class), which could predict the presence/absence of each pathology given an input image. Another set of explorations was using different binary classification problems by grouping the labels available in MRNet, e.g. classifying "acl" vs rest, "meniscus" vs. rest, "acl" or "meniscus" vs rest , "acl" or "abnormal" vs rest. The main motive for exploring such different combination of classes was to generate heatmaps that could highlight all the regions which are clinically significant to detect most of the pathologies commonly associated with the knee. However, the best quality of models was obtained for the "acl" vs rest combination. The initial experiments were made using the SqueezeNet model [14], unlike the AlexNet model used in the original MRNet paper. The reason for choosing SqueezeNet over AlexNet is that is achieves similar performance on image classification benchmarks while having 50x fewer parameters. The network was trained using the Adam optimizer with binary cross entropy using a learning rate of 1e-5 and weight decay of 3.5e-5. The use of Adam is motivated by the fact that it can achieve satisfactory results without need for extensive hyperparameter tuning [10]. The SqueezeNet configuration can be seen in Fig. 4. As will be detailed in the experiments section, even though the SqueezeNet model had good classification metric results, the resulting heatmaps were not clear enough to be used for image reconstruction. For this reason, a newer encoder model was tested, namely EfficientNet V2 [16]. The EfficientNet V2 configuration can be seen in Fig. 5

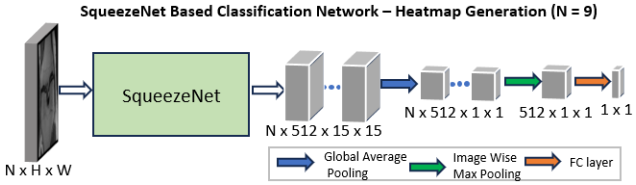

Figure 4: SqueezeNet Image Classifier

One challenge we could foresee is the difference in the data distribution between the MRNet and fastMRI. Unlike the scans in the MRNet dataset which are some sort of post-processed, the scans or images in fastMRI are reconstructed

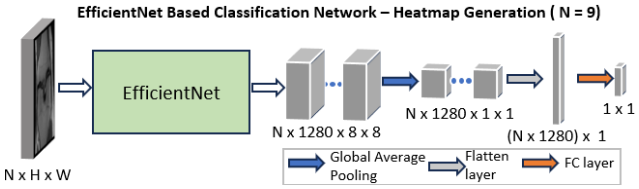

Figure 5: EfficientNet V2 Image Classifier

Figure 6: Illustration of MRI reconstruction Loss functions, a) MRI Image b) Scan Mask Model Loss weights c) Heatmap Mask Model Loss Weights.**W1**–Foreground Weight,**W2**-Background weight, **W3**–Heatmap weight

directly from the raw kspace signals without much of additional processing. We were quite curious to understand how good the quality of heatmaps generated with fastMRI dataset will be when using the model that was trained only on the MRNet dataset.

2.3. MRI reconstruction

To perform MRI reconstruction, we used the U-Net [18] architecture for which we developed an end-to-end training pipeline using PyTorch. We initially trained a baseline model as detailed in [17] to provide a benchmark for attentive reconstruction. Next, we trained models using scan masks and heatmaps, with loss weighting as illustrated in Figure 6. The masks comprised of a set of background and knee region weights to scale the L1 loss during training. Training of the default masked models was followed by hyperparameter tuning as reported in 3.2. Models contained 64 output channels in the first convolutional layer and were trained for 50 epochs, using the RMSprop optimizer with a learning rate of 1e-4 and batch size 1. A step scheduler was used to reduce the learning rate (γ = 0.1) at epoch 40. The best model was selected based on the lowest validation loss. Due to computational constraints, all reconstruction tasks were performed on 9 central slices of each volume in the dataset. The following metrics are computed on the validation data: mean squared error (MSE), normalized mean squared error (NMSE), peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM). Mask-specific versions of these metrics are also calculated, such that image regions are weighted analogously to the loss calculation for mask models. A problem we anticipated was determining the optimal values for the loss weights which was addressed by hyperparameter tuning. We expected to see an improvement in these metrics but the differences were minimal.

3. Experiments and Results

3.1. Heatmaps generation

Firstly, we developed a single multilabel classification model that could generate 3 class labels, namely "acl","meniscus" and "abnormal" for each MRI scan. This model used SqueezeNet as the feature extractor and was initialized with ImageNet [7] pre-trained weights. Since the class label distribution is imbalanced, weighted multilabel cross entropy loss was used to train this model, where each class label was weighted by its frequency. Experiments were carried out to find the best hyperparameter configuration; more specifically the learning rate was tuned. The best model obtained by training the network for 60 epochs with a learning rate of 1e-5 achieved an average validation AUC (across all classes) of 0.8025 as illustrated in Table 1. Though this model achieved good performance on "acl" class, its performance was suboptimal on the other two classes. Also, the quality of heatmaps generated with this model was not consistent and optimal for usage in MRI reconstruction. One reason for this could be that the coronal view may not be optimal to diagnose meniscus tears. On the other hand, for "abnormal" class, since this label covers multiple pathological conditions, it would be difficult for the model to learn to associate this class label to localized coherent regions (in the image).

Next, we developed individual models (Binary classification) to classify each label separately. Similar to Multilabel model, these models were based on SqueezeNet and were trained using the Binary cross entropy loss. For reasons mentioned in the previous section, the losses for each sample were weighted based on their class frequency. Experiments were carried out to find the best parameter configuration for each of the models. This included tuning learning rates, adding learning rate schedulers and using alternate model architectures (like AlexNet in MRNet paper). However, SqueezeNet based models with a learning rate of 1e-5 when trained for 60 epochs yielded the best performing models. The AUC scores for these models on the validation dataset are illustrated in Table 1. Similar to the multilabel case, "acl" classification model achieved the best result, while the performance for other class models was slightly on the lower side. Reasons highlighted in the previous section for this performance discrepancy are valid here too. However, the quality of heatmaps generated from these models was pretty consistent and coherent, especially for the "acl" model (Fig. 8). In the following section we also highlight experiments carried out to further improve the quality of the generated heatmaps.

Besides the previously described multilabel and single label settings, we also experimented with combining label, i.e. "acl" and "meniscus" and "abnormal" versus rest, "acl" and "meniscus" versus rest. SqueezeNet image clas-

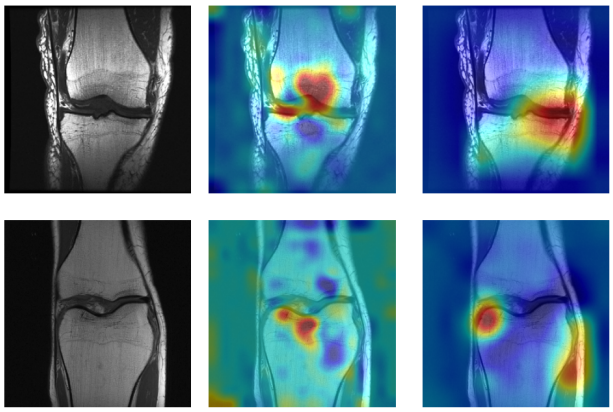

Figure 7: Sample Heatmaps : First row MRNet image, and Second row fastMRI image, with a) Input image b) SqueezeNet heatmap and c) EfficientNet heatmap

sification models were trained on these different combinations and the best results obtained were for "acl" versus rest (which was explored in the previous section). The likeliest reason for this is that ACL injuries are easier to see on a MRI scan in coronal view. After deciding on these 2 classes, the learning rate and weight decay were further tuned for the SqueezeNet network yielding an optimal learning rate of 1e-5 (as previously reported) and a weight decay of 3.5e-5. Without weight decay, overfitting was observed from the learning curves. Additionally, for this combination of labels, the class split is roughly 20% "acl" and 80% rest, so a relatively imbalanced dataset. In order to alleviate this issue, the binary cross entropy loss used previously was replaced with the binary focal loss[13] , with tuning for the gamma parameter which was found to 3.25, which translates into a relatively lower loss for well classified examples. Taken together, these changes led to an improvement in AUC from 0.8474 to 0.8878 as can be seen in Table 1. Even with these metric improvements, the quality of the produced heatmaps was not satisfactory, so SqueezeNet encoder was replaced with EfficientNet v2 encoder which did not work out of the box. Concretely, regardless of the learning rate, the AUC value would hover around 0.5 which roughly reflects a random prediction. On closer inspection, the issue was the Image Wise Max Pooling layer which discarded a lot of information. Replacing this with a Flatten layer whose output goes directly to a now much larger Fully connected layer allowed the EfficientNet encoder to learn, yielding an AUC value of 0.9475. Focal loss was not added to the network because the produced heatmaps were good enough for our use case. An example of heatmaps can be seen in Fig. 8.

Remark that, even though neither SqueezeNet nor EfficientNet were trained on fastMRI data, the generated heatmaps are comparable to those obtained when running on the original MRNet data.

Model Type	ACL	Meniscus	Abn.*
Multilabel	0.8305	0.7918	0.7852
SqueezeNet + CE Loss	0.8474	0.7791	0.7853
SqueezeNet + Focal Loss	0.8878	-	-
EfficientNet	0.9475	-	-

Table 1: AUC values on MRNet validation set,(**Abnormal*)

Model type	Train loss	Validation loss	Mask NMSE*
Default binary	2.496	2.639	0.03469
Tuned binary	2.644	2.896	0.03502
Default heatmap	2.570	2.723	0.03485
Tuned heatmap	3.266	3.447	0.03488

Table 2: Tuning validation metrics. *Binary mask for the binary model and the heatmap mask for the heatmap model.

3.2. MRI reconstruction training

A baseline U-Net model was trained using the parameters from [3] (as defined in 2.3) to use it for comparison with the subsequent masking experiments. These results are reported in Table 3.

3.2.1 Heatmap mask model training

We trained a heatmap-based model over 50 epochs, using parameter values derived from a baseline model. Hyper parameter tuning was done for the 3 loss weights which represent the importance of different pixels in the image: 1) foreground weight (whole knee area), 2) background weight (whole image minus whole knee area) and 3) heatmap weight (regions of the whole knee area identified as being relevant for pathology classification as detailed in 2.2). In Fig. 6 we can see an example of such pixel classification/weighting.

The model employs a loss function that calculates the mean weighted L1 loss (mean absolute error) and leverages both binary masks and heatmap data to modulate the importance of various regions within the input images. After multiple training runs, we found that a combination of a background weight of 3, a foreground weight of 15, and a heatmap region weight of 17 showed some modest improvements over the default model.

During model training, we observed minor overfitting, which we attempted to mitigate by adjusting the dropout value from 0.0 to 0.1. We also experimented with adjustments to the learning rate and gamma value, but these did not significantly affect performance metrics.

Table 2 and 3 show results from the heatmap model. Comparing the performance metrics of both the baseline and the heatmap models reveals only a slight improvement. This suggests that the heatmap model performs better

around areas designated by the heat mask. Similarly, when comparing the overall metrics of both models, it appears that the heatmap model is very slightly superior to the baseline.

3.2.2 Binary mask model training

Initially, the scan mask U-Net model was trained for 50 epochs with parameters like in the baseline U-Net, except for additional parameters specific to the scan mask model which determine how the mask weighs the loss during training. These parameters included the background and foreground weights and were set to 4 and 10 by default. The model achieved the lowest validation loss at epoch 40 and obtained binary mask metrics on the validation dataset reported in Table 2.

To further improve the scan mask model, a hyperparameter optimization study was implemented with Optuna for 30 trails on 200 training and 75 validation volumes. The objective of the study was to minimize the validation scan mask NMSE. Each trial consisted of 10 epochs and used the Tree-structured Parzen Estimator sampler to sample learning rate (1e-5 to 1e-1) and foreground weight (8-12) values while the background weight was kept constant at 4. The best parameters were determined to be a learning rate of 4.37e-05 and foreground weight of 11.

Similarly to the heatmap mask model training mentioned in 3.2.1, Dropout was experimented with in order to reduce the slight overfitting present, but this lead to worse validation metrics so it was not pursued further.

The final tuned scan masked model was then trained on all data for 50 epochs using the optimal parameters outlined above. A comparison of training metrics for the default parameter and tuned models is shown in Table 2.

The tuned model did not improve reconstruction metrics relative to the default parameter model. Consequently, the default model was used as the final binary mask model. Given lower computational constraints, further work to find optimal parameters could involve training the model with more data and for more epochs per trial to obtain more stable results during tuning.

3.3. Summary of results

The baseline, binary and heatmap models achieved similar performance on the reconstruction task as reported in Table 3. The mask models provided a marginal improvement over the image regions without a discernible mask-specific effect. The best (binary mask) model improved knee region NMSE by 1.2% relative to the baseline.

Having observed marginal differences in validation metrics and images (Fig. 8) for the different masking approaches, we investigated whether further increasing the weighting of the loss would lead to a greater impact on the metrics. The best binary mask model was re-trained with a foreground weight of 200 instead of 10. The resulting model did not obtain an improved scan mask NMSE (0.3471) or other scan mask validation metrics relative to the default binary mask model. A similar pattern emerged when testing the heatmap model with extreme values for background and foreground weights. A possible reason for the similar performance of different scan mask models is the low impact of error in the background regions (which are much smaller in area) on the learning process.

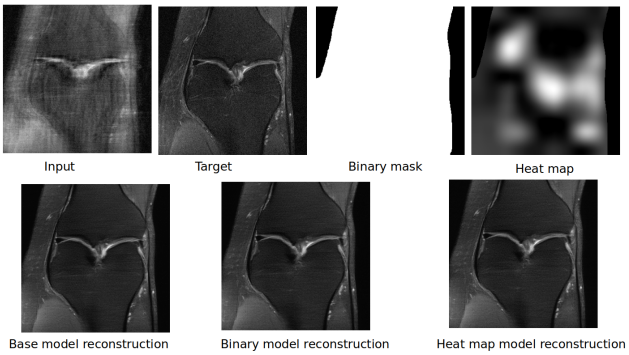

Figure 8: Image reconstruction comparison

	Baseline U-Net	Binary mask U-Net	Heatmap mask U-Net
NMSE	3.854e-02	3.840e-02	3.841e-02
PSNR	29.07	29.09	29.08
SSIM	0.6608	0.6614	0.6611
NMSE[a]	3.510e-02	3.469e-02	3.495e-02
PSNR[a]	29.54	29.64	29.571
SSIM[a]	0.7391	0.7437	0.7398
NMSE[b]	3.500e-02	3.466e-02	3.488e-02
PSNR[b]	36.58	36.65	36.60
SSIM[b]	0.9695	0.9698	0.9695

Table 3: Baseline, Binary mask, and Heatmap model results. [a] BinaryMask region, [b] Heatmap regions

4. Conclusions and future work

In this project we developed an MRI reconstruction methodology and model which can effectively focus on relevant regions within the scans for knee regions. Though we achieved marginal improvements in performance over the regions of interest, the results look promising and open up the potential for further work. Future lines of inquiry include, but are not limited to: 1) comprehensive loss weights tuning 2) other weighted loss functions other than L1 and 3) an extra heatmap prediction task added to the network, forcing the network to learn better features for both tasks. Finally, we could also explore using all the slices instead of middle slices to understand if we can further improve the performance.

Summary of contributions is provided in Table 4.

Student Name	Contributed Aspects	Details
Radu Diaconescu	Heatmaps generation, Scan Mask grabcut step, MRI Reconstruction	Investigated binary classification with label combination. Added and trained SqueezeNet with focal loss, with tuning of gamma parameter. Investigated usage of EfficientNet and debugged cause of it not learning and obtained best model for heatmap generation. Additional contributions include include writing the end to end training code for the MRI reconstruction baseline which includes train and validation loop, training logs and model saving. Investigated grabcut algorithm for improving background segmentation. Providing code review as needed and suggestions for analysis of heatmaps masks and scan masks image reconstruction experiments.
Arun Kumar Siddanahalli Ninge Gowda	Heatmaps generation, Scan Mask Generation, MRI Reconstruction	Defined the overall architecture and methodology for end-to-end attentive/focused MRI reconstruction. Developed end-to-end training flow to train classification models for Heatmap generation. Developed and trained SqueezeNet based Binary Classification and Multilabel classification models using CE loss. Developed an Image processing based algorithm and a fusion algorithm to segment foreground(scan) and background regions in MRI images.Developed code to generate the I/O images, Scan Masks and Heatmap masks(using classification models) needed to train the MRI reconstruction model. Contributed towards the definition and development of Scan Mask and Heatmap mask based loss and evaluation metrics for training reconstruction models. Developed code for training the baseline scan mask and baseline heatmap mask based MRI reconstruction models. Performed code review and provided suggestions for MRI reconstruction analysis
Kamran Munawar	Implementation and Analysis,MRI reconstruction	Implement a basic FastMRI U-Net model in a Jupyter Notebook to gain an understanding of the problem and to estimate the required computational resources, such as processing power. Train baseline model to establish a performance benchmark. Proceed with training and hyperparameter tuning of a heatmap model to enhance the quality of the model. Generate and analyze the performance metrics. Code review as required.
Kamil Żmijewski	MRI reconstruction,	Ensured correct data processing for training and inference to be consistent with prior fastMRI implementation. Contributed to the approach to loss weighting and metric calculation. Provided code review. Trained and tuned the binary mask model. Developed code for inference and calculation of reconstruction metrics. Generated reconstructions and analyzed results.

Table 4: Contributions of team members.

References

[1] https://fastmri.med.nyu.edu/, 2019. 1

[2] https://github.com/ahmedbesbes/mrnet, 2019. 2

[3] https://github.com/facebookresearch/fastmri, 2019. 2, 5

[4] https://github.com/hasibzunair/res-unet-fastmri, 2019. 2

[5] https://stanfordmlgroup.github.io/competitions/mrnet/, 2019. 1

[6] https://github.com/kazishuvo22/fastmri-reconstruction/tree/master, 2020. 2

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database, 2009. 4

[8] Anagha Deshmane et al. Parallel mr imaging, 2012. J Magn Reson Imaging. 1

[9] Bien et al. Deep-learning-assisted diagnosis for knee magnetic resonance imaging: Development and retrospective validation of mrnet, 11 2018. 2

[10] Choi et al. On empirical comparisons of optimizers for deep learning, 2020. 3

[11] Carsten Rother et al. Grab cut -interactive foreground extraction using iterated graph cuts, 2004. ACM Transactions on Graphics (SIGGRAPH). 2

[12] Dilbag Singh et al. Emerging trends in fast mri using deep-learning reconstruction on undersampled k-space data: A systematic review, 2023. Bioengineering MDPI. 1

[13] Girshich et al. Focal loss for dense object detection, 2019. 5

[14] Iandola et al. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡0.5mb model size, 2016. 3

[15] Jiang et al. Accelerating cs-mri reconstruction with fine-tuning wasserstein generative adversarial network, 2019. IEEE Access. 1

[16] Tan et al. Efficientnetv2: Smaller models and faster training, 2021. 3

[17] Florian Knoll et. al Jure Zbontar. fastmri: An open dataset and benchmarks for accelerated mri, 2019. 4

[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. 4

[19] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization, 2015. 2