Project Report : Phase I

Data Collection

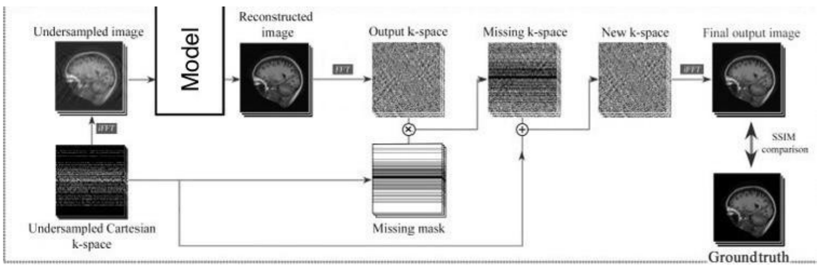
We used dataset avaliable at Kaggle. For our model we required fully sampled MRI kspace data. It was not avaliable. So we used the MRI Images dataset avaliable at Kaggle and processed it according to our need.

Data Preprocessing

- Applied Fourier transform on MRI Images to produce kspaces.
- Choosed an undersampling strategy which will help to reduce the acqusion time significantly and also which is feasible to acquire from the MRI machine.
- Applied this undersampling mask to the kspaces to produce undersampled kspaces.
- Then applied inverse fourier transform to these undersampled kspaces to undersampled images.

Choosing a model architechture

- Summary of the architecture:
  Input Downsampling: The network begins with two downsampling blocks that reduce the input size by half.
  Residual Blocks: It consists of 34 residual blocks. Each block outputs the same size as its input, and the input is added to its output before being forwarded to the next residual block.
  Output Upsampling: Following the residual blocks, the network has two upsampling blocks, which double the input size to restore the original image size.
  Final Convolution Layer: After upsampling, a final convolution layer is added.
  Loss Function: The SSIM Loss is used as the loss function.



This image we have used for the research paper for which we have provided the link before the training and the validation graph.

This model takes input image of size 256*256*1 and the output image of the model is of size 256*256*1.

This model architechture was insipired by paper “ReconResNet: Regularised Residual Learning for MR Image Reconstruction of Undersampled Cartesian and Radial Data”

Link of the paper : https://arxiv.org/pdf/2103.09203.pdf

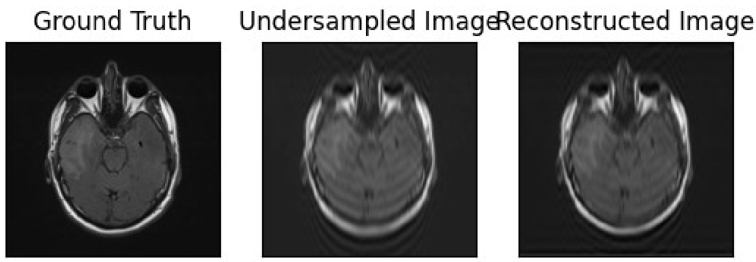
Note: Code is not available for the MRI Reconstruction model in this paper.(only the workflow is given)

Data Postprocessing

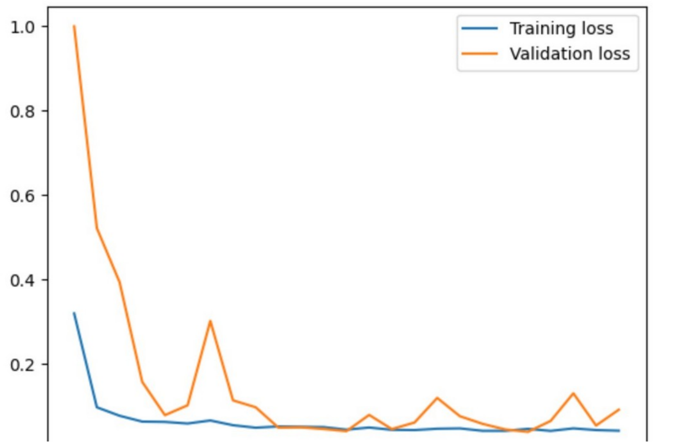
- The obtained image from model is then converted to kspace domain.
- This kspace is multiplied with the inverse of mask used.

- The obtained result is then added with the undersampled kspace to obtain the fully sampled kspace.
- This fullysampled kspace is then converted to image domain which will be the final output.

Results from the model




```
plt.legend()
plt.show()
```



Training loss is 0.0042 and Validation loss is 0.0920.

THIS MUCH PART IS COMPLETED.

TASK FOR FINAL SUBMISSION

- In the original paper , the number of epochs were 200 with batch size 1.
  We have used only 25 epochs with batch size 32 due to less computational resources.
  So in the next submission we will try to enhance this model by training it more.

- We are also going to try a model architecture using GAN to generate the MRI images from the undersampled kspace data.

- We will also provide the quantitative analysis and the comparison of results from both the models with ground truth using some quantitative metrics.

- We will also try to use the Standford Dataset which directly provide us the kspace data.

CHALLENGES FACED

We have drastically reduced the number of epoches and the batch size while training our model due to less computational resources. This might have degraded the quality of results from the expected results.

DIVISION OF TASKS

Satyam performed the data collection , data preprocessing and data postprocessing part.
In the upcoming submission he will be working on the GAN model architechture.

Rohit performed the choosing model architechture , interpreting the research paper and writing the code for the model and model training part.
In the upcoming submission he will be working to enhance the performance of this model and performing the quantitative analysis on the results from both the models and comparing them with the ground truth.