

|  |
| --- |
| Denoising MRI  Scanned images using  Deeplearning |
| Medical Image denoising |
| Authored by: Your Name |

# Introduction

|  |  |
| --- | --- |
| This is a detailed work of denoising the noised MRI images.  In this project we approached the solution using Deeplearning. We have used RIDNET-  A kind of convolutional network which is specifically for the denoising the images.  A Detailed approach is being given in the following report. Objectives  |  | | --- | | In this work we present the implementation of denoising the Magnetic Resonance Imaging (MRI) images using Deep-Learning (RID-Network). The algorithms and the Graphical User Interface (GUI) were developed using python. These model was trained ,tested , and analyzed in python. | |

# Algorithms

|  |
| --- |
| * Initially we divided the given image into patches. So that we could obtain a detailed reconstruction of the image. * We Added different levels of noise to the given images. * We used RID-Net (Real Image Denoising with Feature Attention)   a type of convolutional network mainly built for image denoising for our project. To obtain Denoised images.   * We trained the given MRI images and verified the output and metrics. * Verified various models for various levels of noised images.  Adding Noise to the image We added noise to the image by corrupting the random pixels in image with random color value. |

# Final Algorithm to Denoise the Image

# 

# The model is made up of three main modules namely the element domain, the remaining learning module element, and reconstruction, as shown in img

# Let us consider x a sound input image and a yˆ sound output image. Our feature removal module is made with only one layer to extract the original f0 features for image

# The network has a small depth, but provides a wide field of reception using kernel dilation in each EAM for the first two branch convolutions. Outgoing features of the

# final layer are provided in the reconstruction module, which is also composed of a single conversion layer:

# 

# 5.3. Feature attention

# 

# Attention has long been there; however, it has not yet been used to remove noise from the image. Channel features in photo audio removal modes are treated equally, which is inappropriate in most cases. To help and learn important image content, we focus on the relationship between channel features; hence the name: feature attention.

# Since convolutional layers only exploit local knowledge and can use global context information, we first use global integration to produce statistics that mean the whole picture, other merging options can be explored to represent the image definition. Fc should be the output elements of the final conversion layer with h-shaped maps with size h × w; Inland merging will reduce the size from h × w × c to 1 × 1 × c as follows:

# gp = 1 / h x w i = 1- h i = 1- w fc (i, j),

# where fc (i, j) is the element value in place (i, j) in feature maps.

# In addition, the self-assembly method is used to capture the dependence of the channel on the definition obtained by inland ground integration.

# rc = α (HU (δ (HD (gp)))),

# where HD and HU are a reduction of channel and channel rating users, respectively. The output of the gp integration layer is combined with the Conv sample reduction layer followed by the launch of the. To differentiate the channel features, the output is enclosed in a Conv sampling layer followed by sigmoid activation.

# Summary of Network

# This just the last layers and summary of the model-Network.

# 

# Analysis of noise in the image using PSNR (Pead Signal-to-noise-ratio)

# Peak signal-to-noise ratio (PSNR) is the ratio between the maximum possible power of an image and the power of corrupting noise that affects the quality of its representation. To estimate the PSNR of an image, it is necessary to compare that image to an ideal clean image with the maximum possible power.

# Reference: [Python | Peak Signal-to-Noise Ratio (PSNR) - GeeksforGeeks](https://www.geeksforgeeks.org/python-peak-signal-to-noise-ratio-psnr/)

# v

# Outputs of the model.

# 

# ORIGINAL NOISY IMAGE DENOISED IMAGE

# 

# 

# The higher the value the higher the quality.

# Project Setup

First of all before running the project. Install the following python packages

* TensorFlow --------------------------------------------------( pip install tensorflow)
* NumPy --------------------------------------------------------(pip install numpy)
* Matplotlib ----------------------------------------------------(pip install matplotlib)
* cv2 -------------------------------------------------------------(pip install opencv-python)
* tkinter--------------------------------------------------------- (pip install tkinter)
* PIL --------------------------------------------------------------(pip install pillow)

After installing the given python packages.

* For detailed information of the model and output analysis. Refer the provided Python Jupiter Notebook
* And for GUI Application

Run the Shell Command :- python app.py

# GUI Model

# The GUI sample looks like.

# 

# Selecting Image and add noise if wanted.

# 

# Output Images

# Matplotlib Outputs

# 

# 

# Conclusion

|  |
| --- |
| In this work we present the implementation of denoising Magnetic Resonance Imaging (MRI) images using Deep learning approach. |

# References

<https://arxiv.org/pdf/1904.07396.pdf>

<http://ani.stat.fsu.edu/~abarbu/Renoir.html>

# [[1904.07396] Real Image Denoising with Feature Attention (arxiv.org)](https://arxiv.org/abs/1904.07396)