UNIVERSITY OF CAMBRIDGE

DiS MPHIL

A2 Data Science Applications to Medical Imaging Coursework

P. Antonopoulos
March 2025

Word count: 2999

Contents

1 Introduction

In this project, PET-CT image reconstruction, MRI image de-noising, and CT image segmentation and classification were explored. This report is split according to the three aforementioned sections. In each section, the method followed for each exercise is detailed where appropriate and observations are discussed. Additionally, responses to written questions are also presented.

2 Module 1

2.1 Exercise 1.1

To clean up the CT sinogram before reconstruction, the dark field, D `ct_dark.npy` and flat field, F `ct_flat.npy` measurements were used in the following equation [1],

$$S' = \frac{S - D}{F - D} \tag{1}$$

where S and S' are the original and cleaned CT sinograms, respectively and are illustrated below.

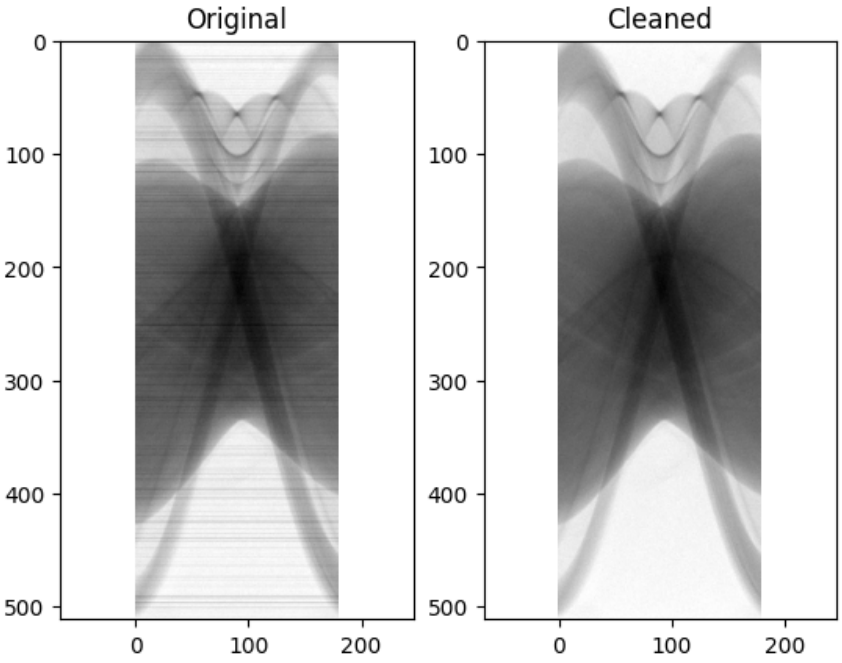

Figure 1: The original and cleaned CT sinograms.

Similarly, to clean up the PET sinogram, the detector gain calibration, C was used in the equation,

$$P' = \frac{P}{C} \tag{2}$$

where P and P' are the original and cleaned PET sinograms, respectively and are also illustrated below.

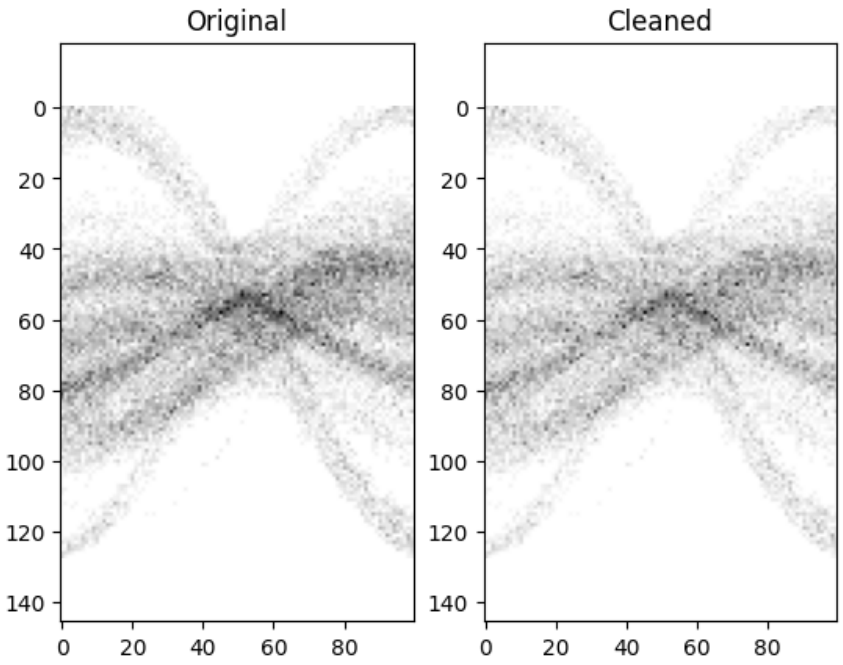

Figure 2: The original and cleaned PET sinograms.

These operations were done using `numpy` for element-wise division of the arrays.

2.2 Exercise 1.2

To reconstruct the CT image with Filtered back Projection (FBP), a Python function was defined which took the sinogram and applied FBP with various filters using the `iradon` function from `skimage`. It should be noted that although the Beer-Lambert equation states the absorption sinogram, p as, $p = -\log(I/I_0)$ (where I is the transmission sinogram and I_0 is a normalisation), the correction of the CT sinogram using the flat field in equation 1 contain I_0 and therefore the sinogram passed into this function (and for OS-SART and SIRT below) was $p = -\log(S')$.

The implementation of the Ordered-Subset Simultaneous Algebraic Reconstruction Technique (OS-SART) made use of the equation given in the question sheet. To divide the sinogram into 10 parts with an equal number of angles, a function was written to generate 10 subsets of 18 random angles between 0-180 *without replacement* to ensure all projection angles were used. Following this, OS-SART was performed by sequentially accessing these random subsets, indexing the sinogram, and using the subset angles in the `radon` and `iradon` functions, while incorporating the indexed sinogram in the residual calculation.

To find appropriate values of the learning factor, γ and number of iterations, $K = max(k)$ where OS-SART produced a better image than FBP, trial and error was used which resulted in values of $\gamma = -0.001$ and $K = 100$. The reconstructed images can be seen below.

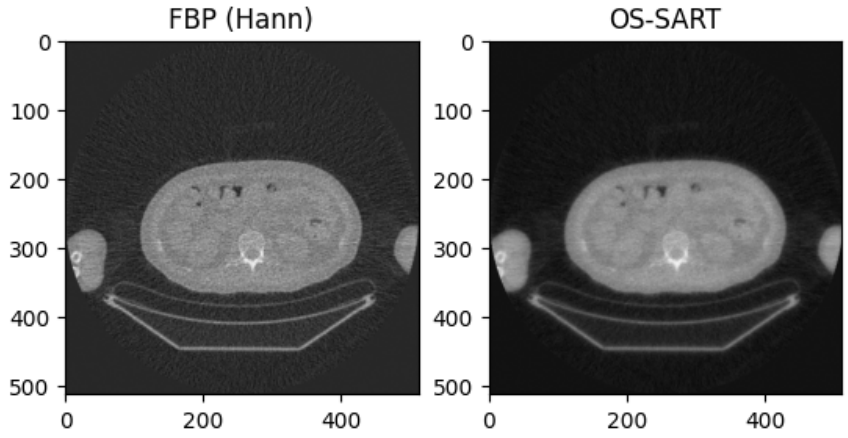

Figure 3: FBP (using Hann filer) and OS-SART CT reconstructed images.

As an extra task, the Simultaneous Iterative Reconstruction Technique (SIRT) was implemented and compared to OS-SART to see which converged to an acceptable image in fewer iterations and which was faster. This comparison was done by displaying the reconstructed images every 10 iterations and recording the total reconstruction time. While OS-SART and SIRT initially returned noticeably different images (with SIRT appearing slightly better), both methods soon converged to an adequate image in ≈ 90 iterations (although they became indistinguishable within ≈ 10 iterations). However, SIRT's average iteration time was an order of magnitude higher, leading to a significantly longer total convergence time. This is illustrated below.

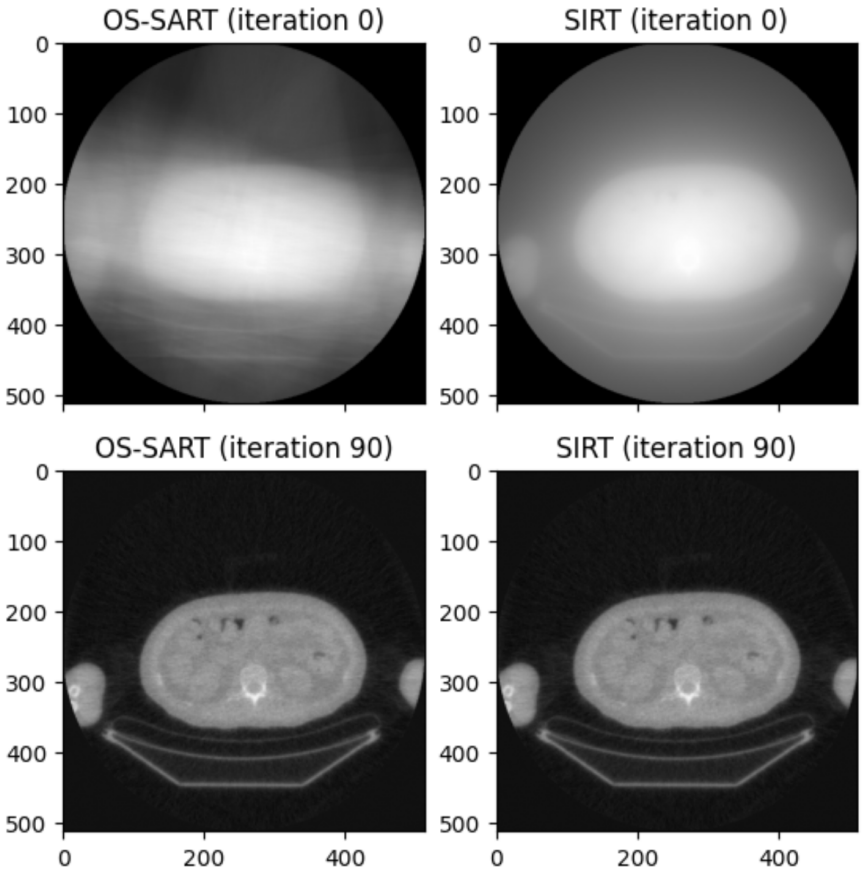

Figure 4: OS-SART and SIRT reconstructed images after the first (0 based indexing) and 90th iteration.

2.3 Exercise 1.3

To correct the PET sinogram for attenuation, the CT sinogram was used to obtain an attenuation sinogram as it gives structural information. To do this, a scale factor of 0.25 was determined between the PET and CT images using the values given in the question paper. This factor was used to resize the CT image to the same size as the PET image (so the sinograms would be the same size). The sinogram produced using `radon` was used to obtain an attenuation sinogram by taking the exponential of it according to the Beer-Lambert law. Finally, the attenuation corrected PET sinogram (shown below) was obtained by multiplying the cleaned PET sinogram from exercise 1.1 with the attenuation sinogram.

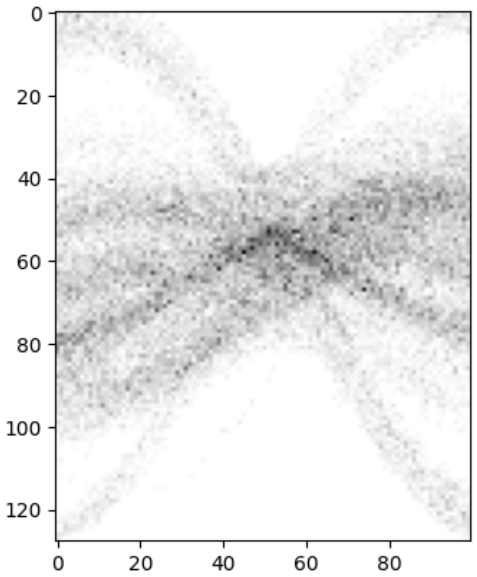

Figure 5: Attenuation corrected PET sinogram.

2.4 Exercise 1.4

To reconstruct the PET image with FBP, the attenuation corrected PET sinogram was used in the same Python function defined to perform FBP in exercise 1.2. Importantly, 100 angles between 0-180 degrees was used for the FBP reconstruction to reflect the sinogram being binned to 180 degrees over 100 different angles as specified in the question paper.

The implementation of the Ordered Subset Expectation Maximisation (OSEM) technique made use of the equation given in the question sheet. Similar to the method followed for OS-SART, a function was defined which generated 10 subsets of 10 random angles between 0-100 without replacement. It should be noted that to account for the binning, all angles were scaled by 1.8. Following this, the OSEM equation was implemented by sequentially accessing these random subsets and using them to index the attenuation corrected PET sinogram. The subset angles were then used in the `radon` and `iradon` functions, while the indexed sinogram was utilised in the ratio calculation, with small offsets added to prevent division by zero errors. The reconstructions from FBP and OSEM are presented below.

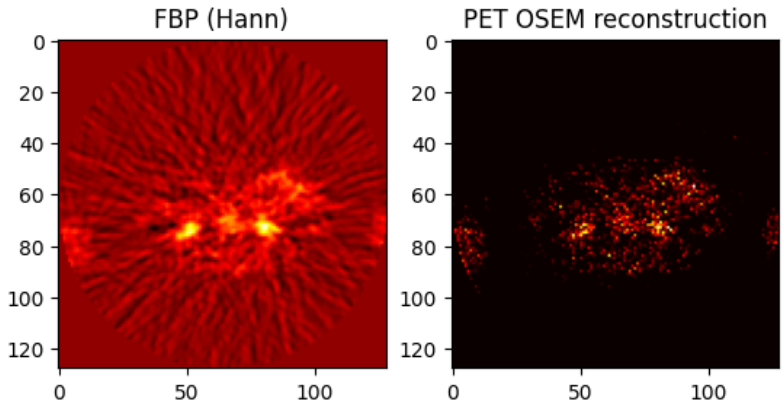

Figure 6: FBP (using Hann filer) and OSEM PET reconstructed images.

As an extra task, the Maximum likelihood Expectation Maximisation (MLEM) algorithm was implemented and compared to OSEM. This comparison was done in the same way as with OS-SART and SIRT. While OSEM and MLEM initially produced different images, they eventually converged to similar reconstructions within $\approx$10 iterations and remained consistent thereafter, with both yielding the best image after $\approx$90 iterations. Additionally, the reconstruction time required for MLEM was an order of magnitude greater than OSEM, meaning it ultimately took longer. A comparison of the reconstructions is presented below.

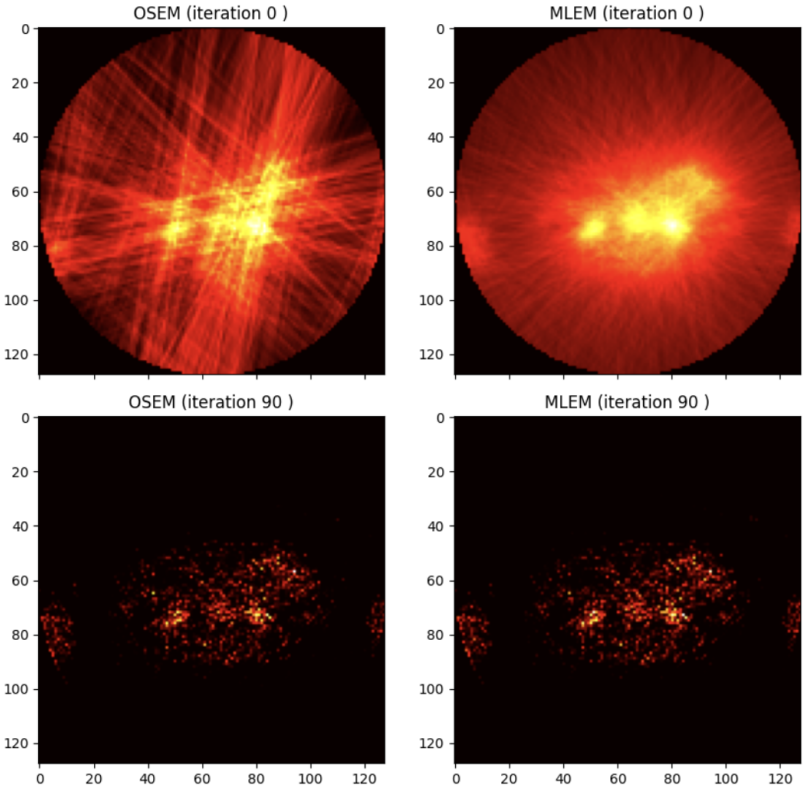

Figure 7: OSEM and MLEM reconstructed images after the first (0 based indexing) and 90th iteration.

2.5 Exercise 1.5

The PET and CT scans were overlayed by displaying them both and using the `alpha` parameter to change the opacity of each image so they could both be seen, this is shown below.

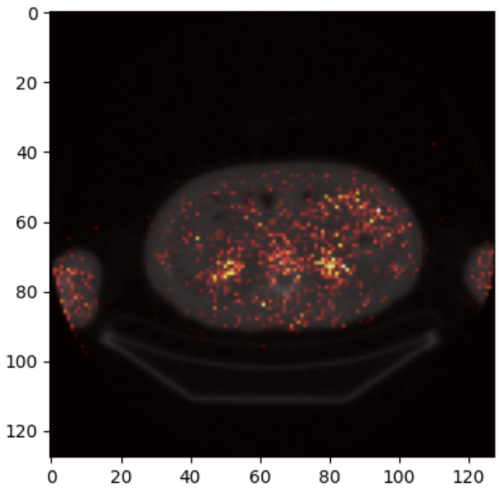

Figure 8: Overlayed OSEM PET reconstruction and OS-SART CT reconstruction.

The location of the annihilation event along the line of response, x in TOF-PET can be obtained using the difference in the arrival times, Δt of the two photos at the opposite detectors through the equation,

$$x = \frac{c\Delta t}{2} \tag{3}$$

where c is the speed of light [2].

In order to eliminate the need for reconstruction, a required spatial resolution of 1mm can be taken which, when substituted into equation 3 and rearranged yields a temporal resolution of Δt =6.7ps.

Because PET involves a Poisson process (counting the number of photons detected), it can be understood that OSEM is well suited to model the noise as it minimises the log-loss assuming a Poisson distribution [3]. However, CT scans generally have Gaussian noise, hence why OSEM is used more in PET rather than CT. OSEM is mathematically different to gradient descent as the former is based on the expectation maximisation algorithm and multiplies the residuals instead of the latter gradient descent algorithm where the residuals are summed.

The extra data processing in a PET-MR scanner could be to correct for artefacts that occur when using MRI instead of CT, such as image wrap/aliasing, susceptibility or RF coils. This would be done computationally.

3 Module 2

3.1 Exercise 2.1

After loading the complex data into an array, the `.shape()` function was used to print the array shape. From this, it could be seen that the coil dimension corresponded to the first array dimension.

Following this, images of the magnitude of k-space were displayed by iterating through the coil dimension, using `np.abs()` to compute the magnitude, and `np.log1p()` to scale the result. This is shown below.

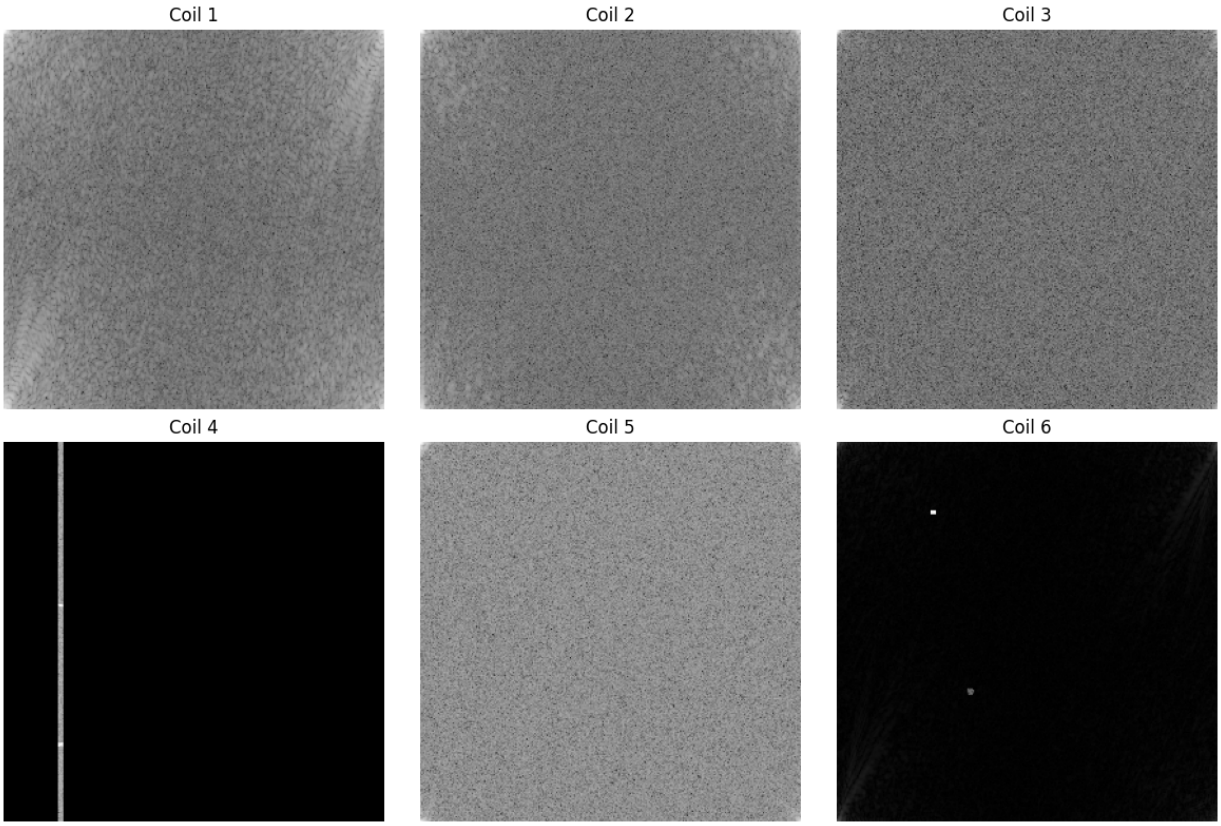

Figure 9: Magnitude of MRI coils in k-space.

At this point, the `np.fft.ifft2()` function was applied to the complex data to transpose it into image space. The magnitude of the image space data was displayed for the first coil using the `np.abs()` function and the associated phase image was obtained using `np.angle()`. These images are shown below.

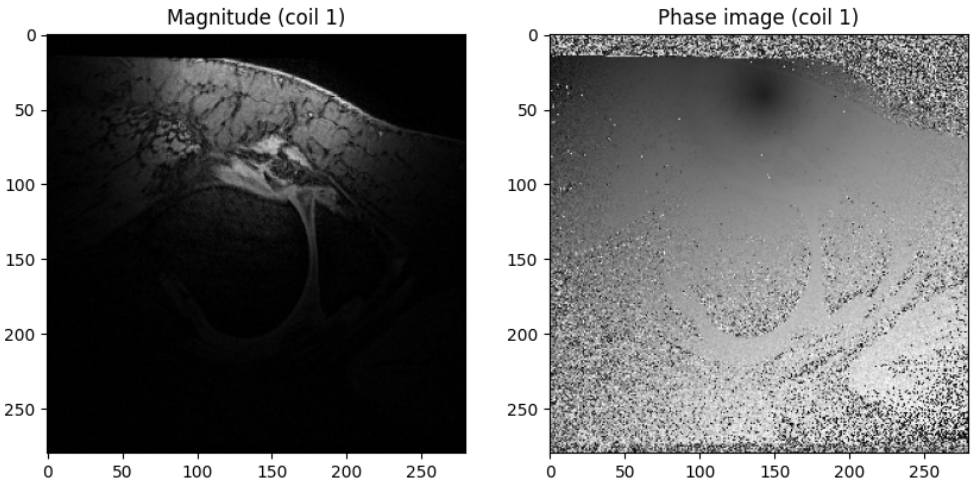

Figure 10: Magnitude and phase of the first MRI coil in image space.

Similarly, the magnitude images from all coils presented below were obtained by iterating through the coils and plotting the `np.abs()` of the transposed array.

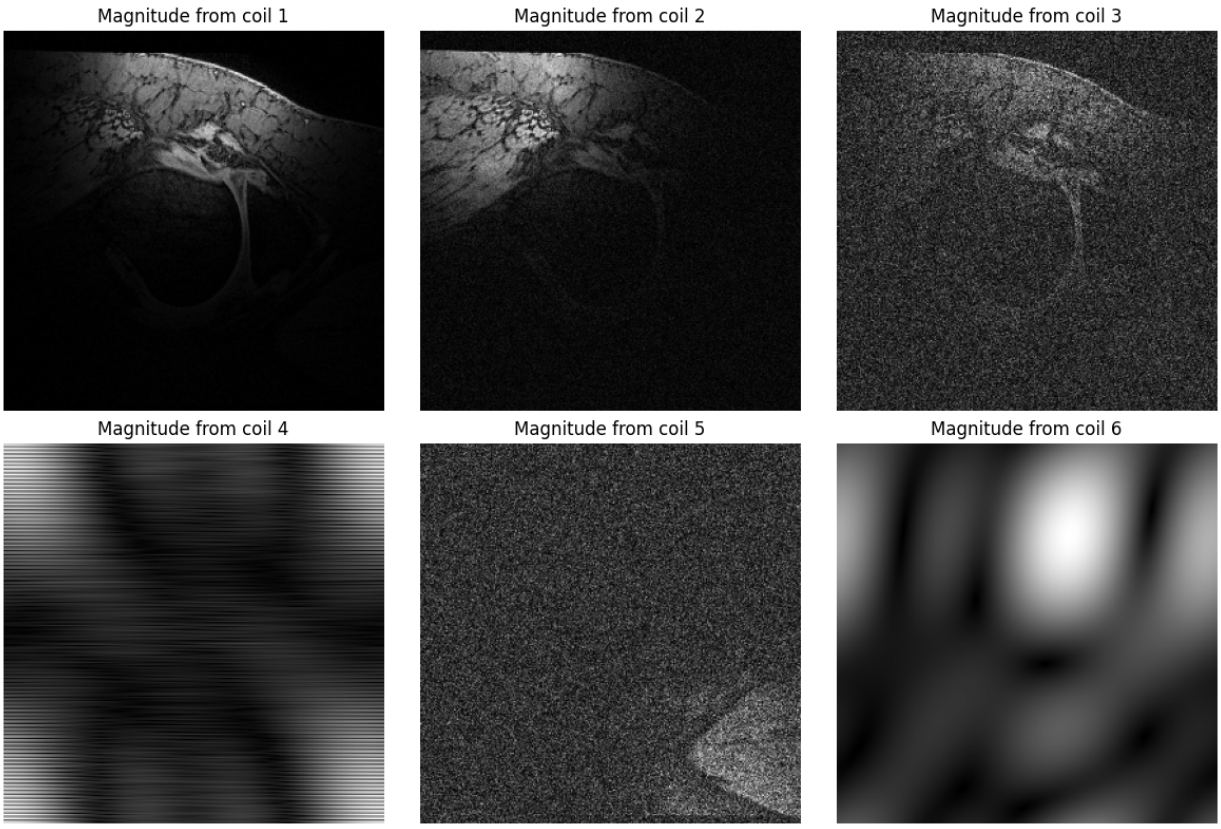

Figure 11: Magnitude of MRI coils in image space.

To combine the image space data from all coils into a single image, the sum of the squared magnitudes across the coils was computed and the final array square rooted. This yielded the image below.

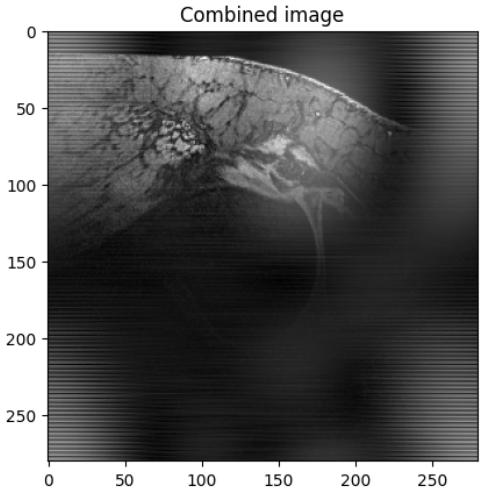

Figure 12: Combined images from all coils.

From figure 11, it was observed that coils one and two produced adequate images and three and five appeared primarily affected by noise. In contrast, coils four and six appeared erroneous (as could also be seen in figure 9), which may explain the combined image in figure 12 exhibiting horizontal stripes and appearing 'patchy'. To investigate this further, another combined image was computed in the same manner as before, but with coils four and six excluded. This is presented below for comparison.

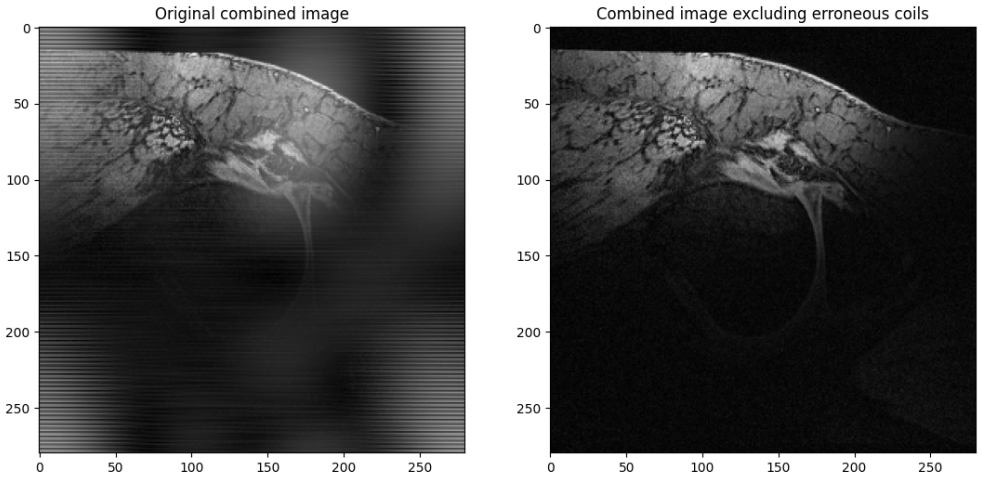

Figure 13: Comparison between the original combined image and the image obtained when excluding the erroneous coils.

It can be seen that this image is significantly improved, suggesting that manual intervention to omit erroneous coils can enhance the obtained image.

3.2 Exercise 2.2

To de-noise the images from all coils, the Gaussian filter, bilateral, and wavelet methods were used. To do this, the Gaussian filter was first loaded in from `scipy.ndimage` and applied to filter the magnitude of the image from each coil using an arbitrary `sigma=1` parameter. These images are shown below.

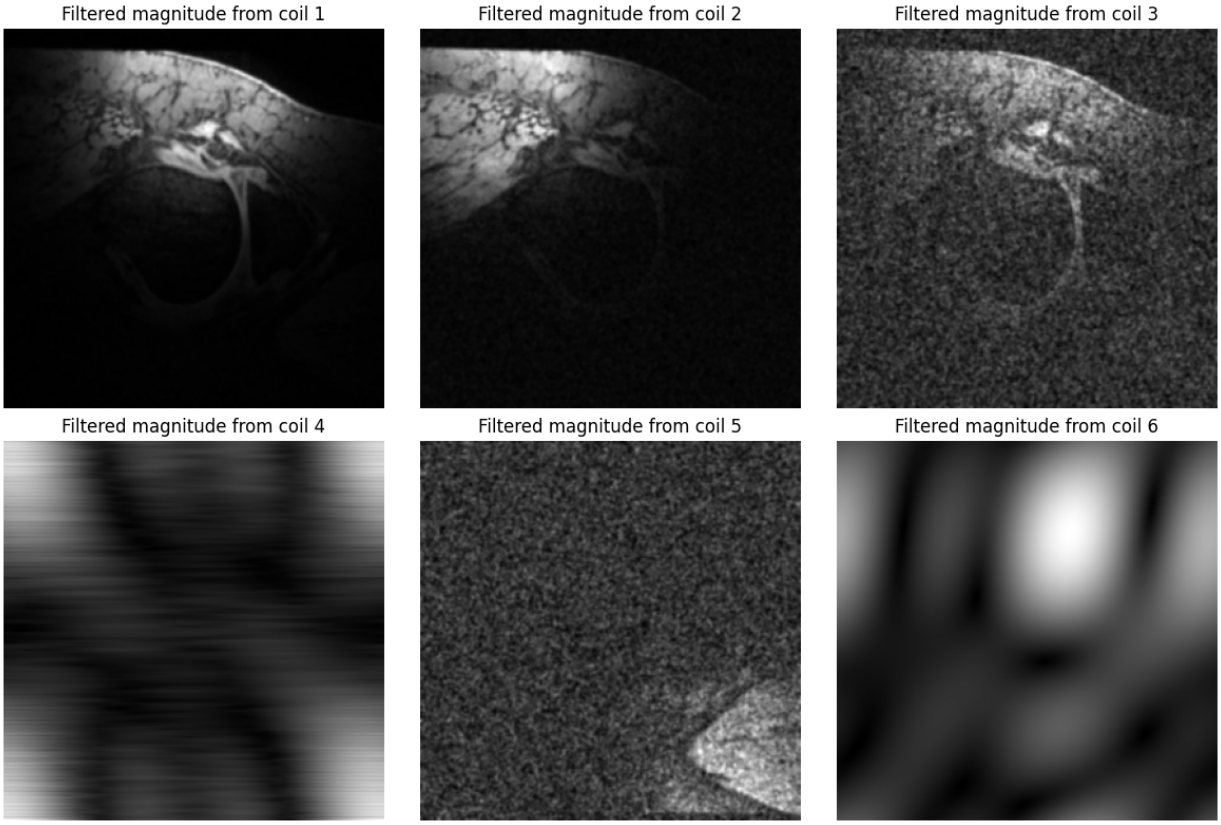

Figure 14: De-noised images from all coils using the Gaussian filter.

Similarly, the bilateral and wavelet de-noising functions from `skimage.restoration` were imported and applied to the magnitude images, yielding the results displayed below.

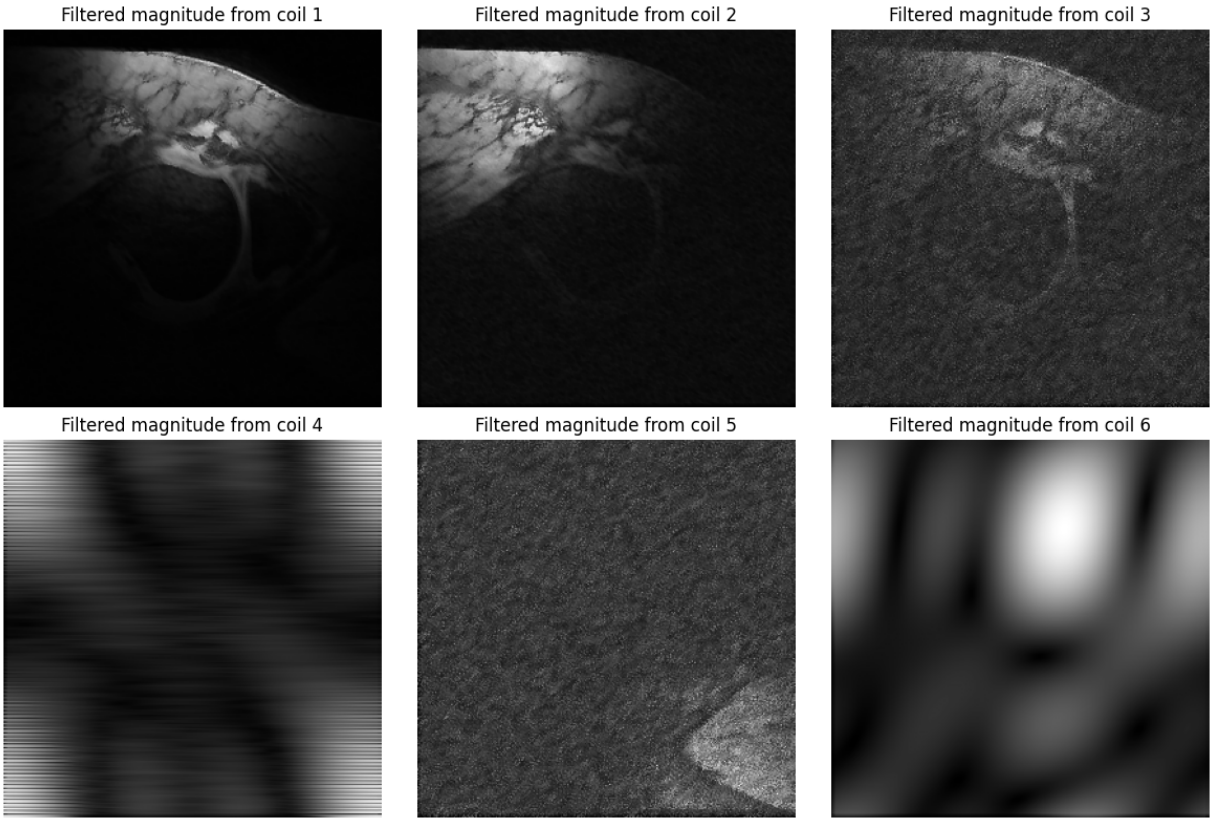

Figure 15: De-noised images from all coils using the bilateral method.

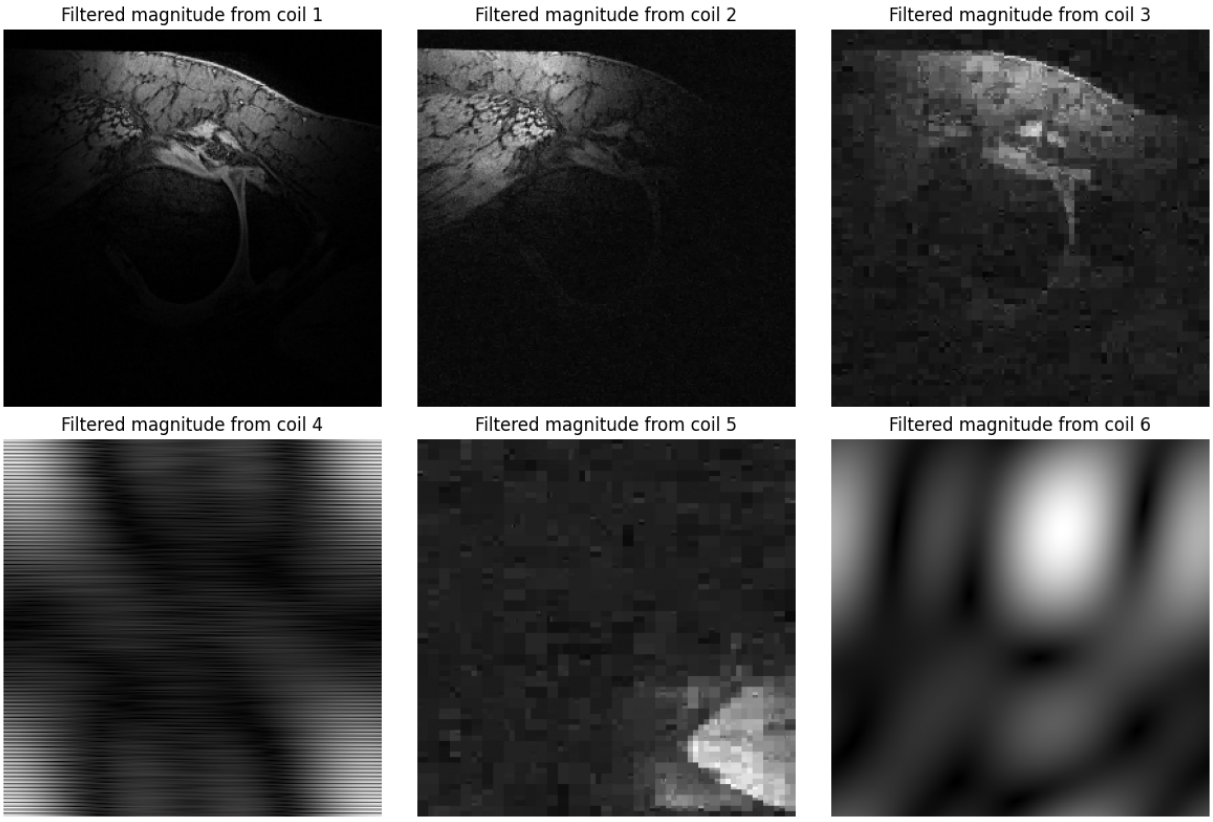

Figure 16: De-noised images from all coils using the wavelet method.

It can be seen that the Gaussian filter improves the images obtained from coils three and five, with the region of interest being highlighted more compared to background noise which is smoothed out. Specifically, in coil three, features in the bone can be made out more clearly compared to the original image where it might have been challenging to distinguish them from noise. Additionally, this smoothing effect can also be seen in coils one and two, where the image appears slightly blurry compared to the original case. Finally, little effect is noted for coils four and six, reflecting their original blurry images.

The bilateral de-noising method does not appear optimal for the images. This is because it fails to highlight the region of interest in coils three and five in the same way that the Gaussian filter did, hence making features harder to identify. This is further noted in coils one and two where features previously observed appear less pronounced. While the wavelet de-noising method preserves features observed in coils one and two (and the images look almost unchanged), the images from coils three and five appear extremely pixelated. Although in these images the region of interest is highlighted more compared to background noise, it is challenging to resolve any meaningful features within the bone. As with the Gaussian filter, both the bilateral and wavelet method yield approximately no change to the erroneous coils four and six.

Following this, a Butterworth filter of the same size as the first coil's image was defined using the provided function. Since applying a filter is a convolution, the convolution theorem was utilised, where the filter and image are multiplied in k-space. To achieve this, the image was first shifted to be zero-centered using `np.fft.fftshift` before being multiplied by the Butterworth filter. To obtain the filtered image, the product was passed through `np.fft.ifftshift` to ensure the input was correctly formatted for `np.fft.ifft2`, which performed the inverse Fourier transform to convert the k-space filtered result back into image space. As previously done, the magnitude and phase images shown below were displayed using the `np.abs()` and `np.angle()` functions, respectively.

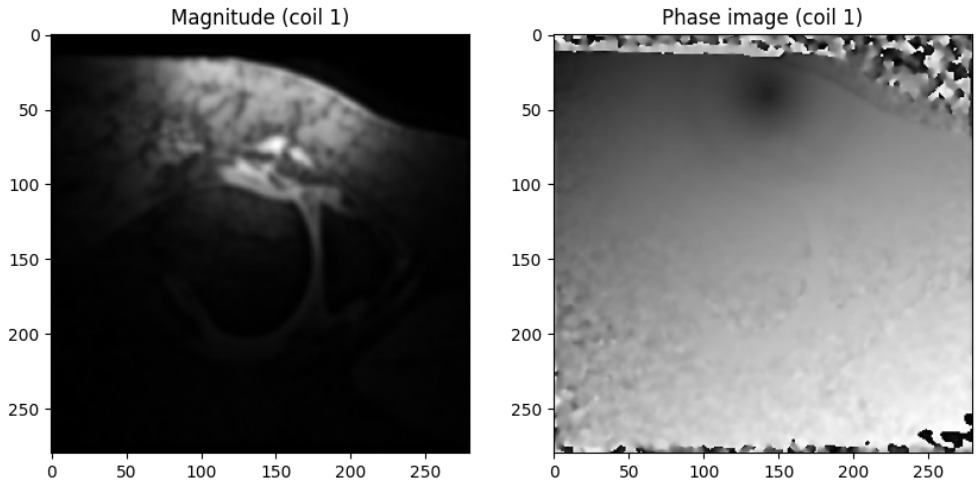

Figure 17: Butterworth filtered magnitude and phase images.

This de-noising method produces a similar result to the Gaussian filter, where features appear smoothed, potentially reflecting the fact that both filters are not edge preserving [4, 5]. However, it is apparent that the Butterworth-filtered image overall appears blurrier. This is because Gaussian filtering applies localized smoothing across the image, while the Butterworth filter attenuates high-frequency information in k-space, leading to more pronounced blurring, especially at edges.

Additionally, the Butterworth filtered image also appeared smoother than images obtained from the wavelet and bilateral de-noising methods, reflecting the edge preserving nature of the bilateral [6] and wavelet methods [7].

Beyond their filtered images, these filters also have differences in the way they are applied to images. While the Butterworth and Gaussian filters can be applied in Fourier space using the convolution theorem, the wavelet method instead makes use of wavelet transforms [7]. Likewise, the bilateral de-noising method is instead applied in the spatial domain as it is a non-linear operation [8].

The Butterworth filter was used to create a new recombined image (shown below) using the same method as done previously.

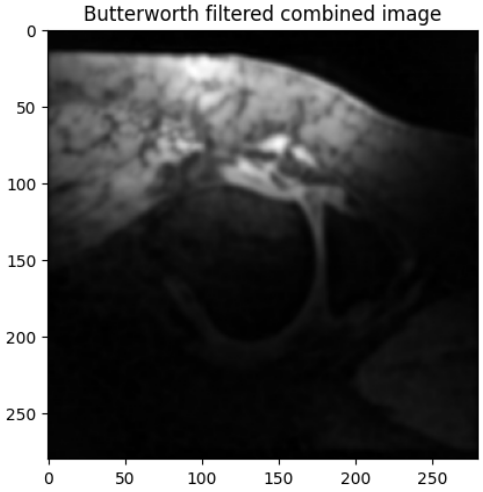

Figure 18: Combined images using the Butterworth filter.

One method to further improve this combined image quality could be to exclude the erroneous coils as done previously. Another method could be to tune the cut-off and order hyperparameters for the Butterworth filter.

4 Module 3

4.1 Exercise 3.1

The NIfTI files were opened and an array was created for each patient scan and segmentation mask by defining a Python function which utilised the SimpleITK library to read an image into a `numpy` array. This function was iteratively called for each patient scan and segmentation mask. To find the range of voxels in which the segmentation exists, the `np.where()` function was used to obtain the indices in all three dimensions of the segmentation. Then, the voxel ranges were determined using list comprehension to find the minimum and maximum indices where the segmentation was defined in each dimension. At this point, the minimum and maximum ranges were expanded by 30 voxels in the x and y dimensions and by 5 voxels in the z dimension using list comprehension. These were then used to create a subvolume of the images by slicing the original image within the widened ranges.

To segment the images using a threshold function, two Python functions were written. The first computed the minimum and maximum thresholds corresponding to the minimum and maximum intensities of voxels inside the segmentation. This was achieved by extracting a subvolume of the segmentation mask, using it to obtain an array of voxel intensities within the subvolume, and then determining the minimum and maximum values of this array. The second function used these values to segment the subvolume by keeping only elements where the intensities fell within the thresholds.

The segmentations obtained from the thresholding algorithm were compared with the ground truth ones by displaying an arbitrary slice for all patients. While the thresholding algorithm produced some segmentations which were comparable to the truth, there were numerous examples where background voxels would be included - as shown below. This was not wholly unexpected due to the algorithm naively finding minimum and maximum intensities of the tumour and assuming background regions would not contain voxels which fall within this range. For example, in CT scans, the intensity is dependent on x-ray attenuation, which in turn depends on tissue type. A tumour, being soft tissue, can therefore have a similar intensity to other soft tissues such as muscle and organs (assuming no contrast agents are used). The segmentation function could be improved by using Gaussian Mixture Models to fit the subvolume intensities and using this to segment the tumour.

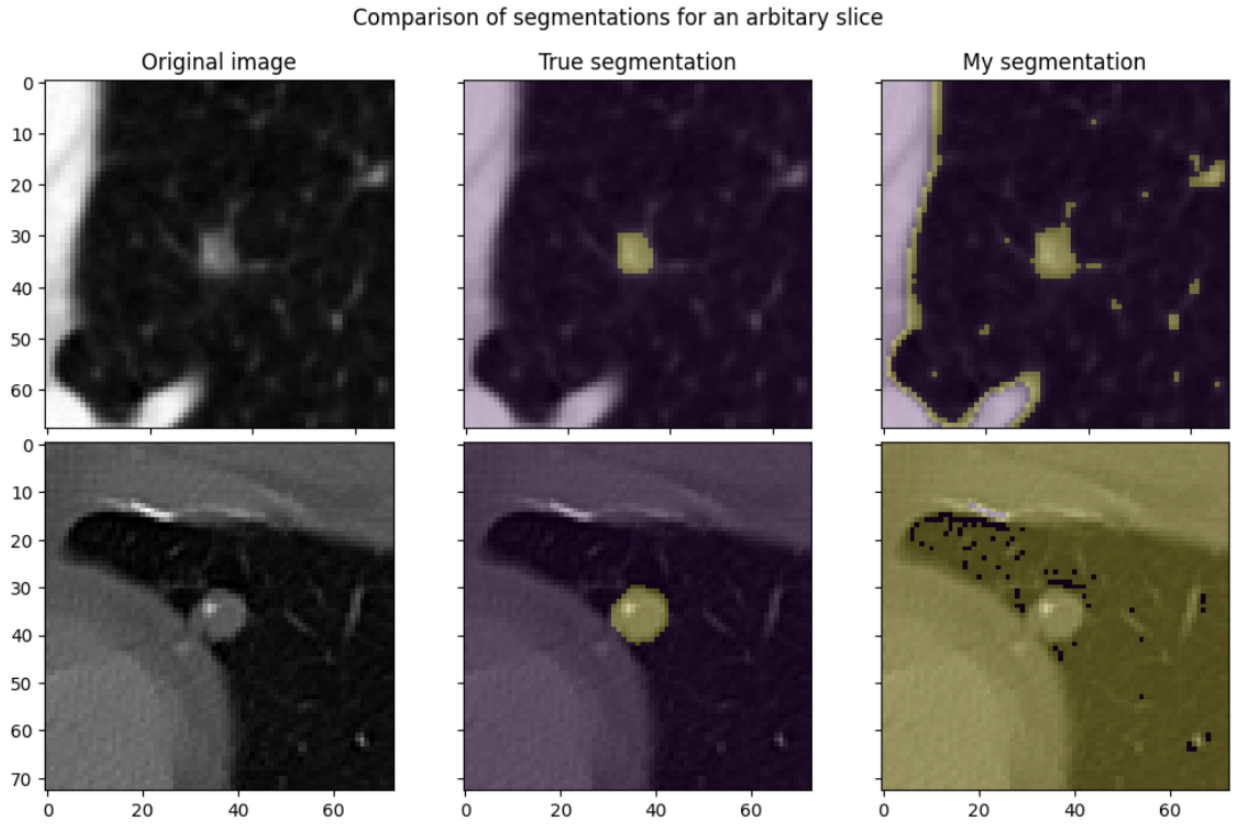

Figure 19: A decent (top) and bad (bottom) example of the thresholding algorithm's performance. The overlayed yellow regions represent the segmentations.

4.2 Exercise 3.2

To write Python functions which calculated the energy, Mean Absolute Deviation (MAD), and Uniformity, one which returned the set of voxel intensities in the ground truth segmentations was defined. The equations to calculate Energy, MAD and Uniformity were then translated into code where the data type of the intensity array was changed to `np.float64` to have enough precision for the Energy values. In generating the histogram to calculate the Uniformity, an arbitrary 100 bins were used to strike a balance between being too wide that features would not be captured and too narrow that noise would prevail over signal. These radiomic features were then computed and displayed in a `pandas` dataframe alongside the clinical labels.

To determine which feature should be used to classify between benign and malignant tumours, two methods were employed. The first method involved plotting histograms of the features, with the data labelled according to whether they corresponded to benign or malignant tumours. The motivation for this method was that if the histograms for benign and malignant tumours were clearly separated for a particular feature, then that feature would be well-suited to classify between the two. These histograms are shown below where it should be noted that the labels for 2 and 3 were combined, as both corresponded to malignant tumours [9].

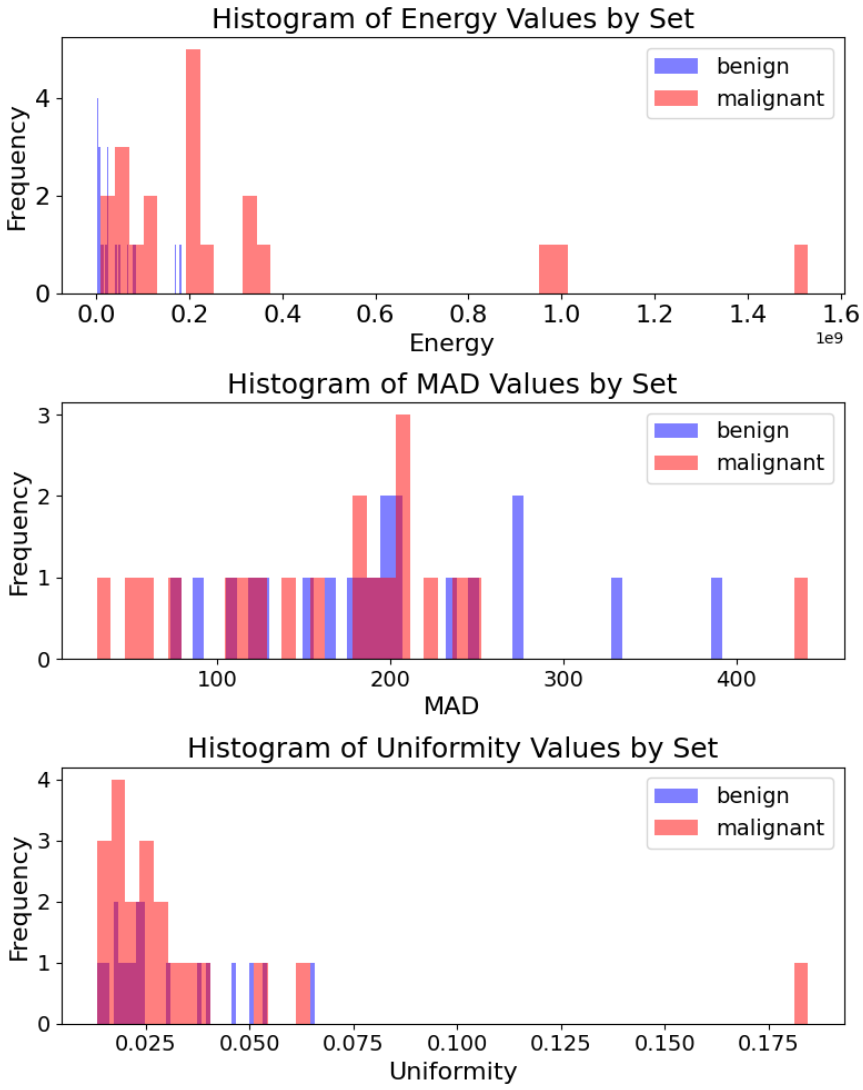

Figure 20: Histograms for the radiomic features.

From figure 21, it can be seen that while the histograms for all features appear overlapped, the benign and malignant Energy histograms appear more separated than the MAD and Uniformity ones. To further investigate this, a second method was proposed which was to create a box-plot for each feature, where again the benign and malignant tumours were separated. This was to visualise the spread of values and the median without any hyperparameters such as the number of bins. These are also shown below.

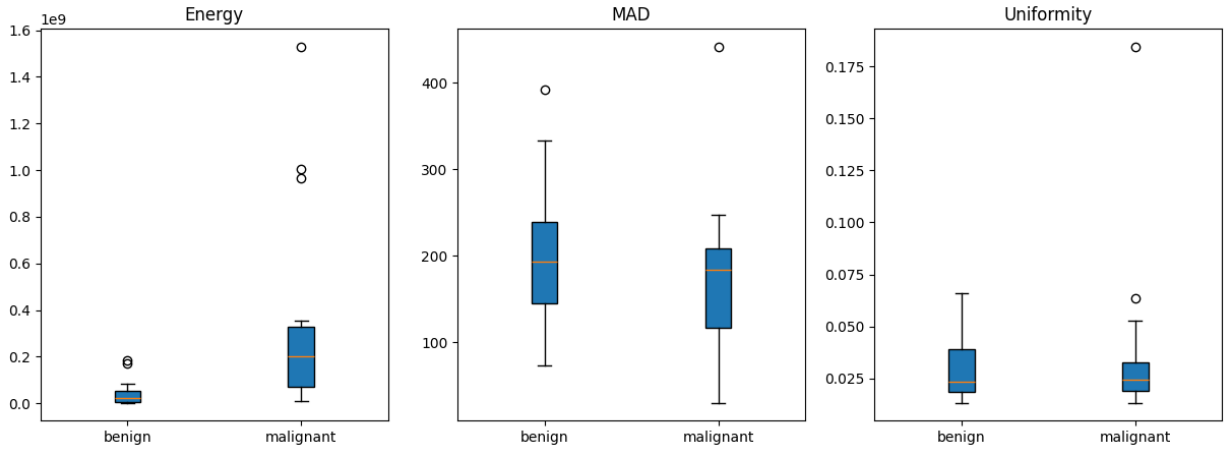

Figure 21: Box-plots for the radiomic features where the orange line represents the median.

From figure 22, it is observed that the Energy feature provides a clearer distinction between benign and malignant tumours compared to the MAD and Uniformity features. The spread of values and the median for Energy appear more distinct, reinforcing the preliminary observations from figure 21. Therefore, the Energy feature would be used to classify between benign and malignant tumours as a new tumour of a given energy could be attributed to either set.

5 Summary

In conclusion, this project explored PET-CT image reconstruction, MRI image de-noising, and CT image segmentation and classification. The accompanying code can be found as Jupyter notebooks in the accompanying GitLab repository.

References

[1] V. Van Nieuwenhove, J. De Beenhouwer, F. De Carlo, L. Mancini, F. Marone, and J. Sijbers, "Dynamic intensity normalization using eigen flat fields in x-ray imaging," *Optics express*, vol. 23, no. 21, pp. 27 975–27 989, 2015.

[2] S. Surti, "Update on time-of-flight pet imaging," *Journal of Nuclear Medicine*, vol. 56, no. 1, pp. 98–105, 2015.

[3] N. Denisova, P. Ruzankin, and Y. Lim, "Statistical approach to inverse problems in emission tomography with poisson data," *METHODS OF AEROPHYSICAL RESEARCH*, p. 37, 2020.

[4] G. Deng and L. Cahill, "An adaptive gaussian filter for noise reduction and edge detection," in *1993 IEEE conference record nuclear science symposium and medical imaging conference*. IEEE, 1993, pp. 1615–1619.

[5] M. Iwanowski, "Edge-aware color image manipulation by combination of low-pass linear filter and morphological processing of its residuals," in *Computer Vision and Graphics: International Conference, ICCVG 2020, Warsaw, Poland, September 14–16, 2020, Proceedings*. Springer, 2020, pp. 59–71.

[6] B. Zhang and J. P. Allebach, "Adaptive bilateral filter for sharpness enhancement and noise removal," *IEEE transactions on Image Processing*, vol. 17, no. 5, pp. 664–678, 2008.

[7] N. You, L. Han, D. Zhu, and W. Song, "Research on image denoising in edge detection based on wavelet transform," *Applied Sciences*, vol. 13, no. 3, p. 1837, 2023.

[8] P. D. Patil and A. D. Kumbhar, "Bilateral filter for image denoising," in *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*. IEEE, 2015, pp. 299–302.

[9] S. G. Armato III, G. McLennan, L. Bidaut, M. F. McNitt-Gray, C. R. Meyer, A. P. Reeves, B. Zhao, D. R. Aberle, C. I. Henschke, E. A. Hoffman *et al.*, "The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans," *Medical physics*, vol. 38, no. 2, pp. 915–931, 2011.